Virt&l-Comm.3.2012.23

Grid Enabled Distributed Computing: from Molecular Dynamics to Multiscale Simulations

Alessandro Costantini

ISSN: 2279-8773

ISSN: 2279-8773

Virt&I-Comm.3.2012.23

UNIVERSITÀ DEGLI STUDI DI PERUGIA

Dottorato di Ricerca in Scienze Chimiche XVI ciclo
Settore scientifico disciplinare CHIM/03

GRID ENABLED DISTRIBUTED COMPUTING: FROM MOLECULAR DYNAMICS TO MULTISCALE SIMULATIONS

Alessandro Costantini

Coordinatore:

Prof. Vincenzo Aquilanti

Relatori:

Prof. Antonio Laganà

Prof.ssa Margarita Albertì

Febbraio 2009

ISSN: 2279-8773

In copertina una rappresentazione simbolica della Griglia di Calcolo come risorsa computazionale in continua espansione e strumento di interconnessone tra i ricercatori appartenenti al mondo scientifico.

ISSN: 2279-8773

"[...] la nobiltà dell'uomo, acquisita in cento secoli di prove e di errori, era consistita nel farsi signore della materia, [...] io mi ero iscritto a Chimica perché a questa nobiltà mi volevo mantenere fedele. [...] vincere la materia è cmprenderla, e comprendere la materia è necessario per comprendere l'universo e noi stessi: [...] quindi il Sistema Periodico di Mendeleev, [...], era una poesia, piú alta e piú solenne di tutte le poesie digerite in liceo [...]."

P. Levi, 1939

"Ma adesso che viene la sera ed il buio
mi toglie il dolore dagli occhi
e scivola il sole al di là delle dune
a violentare altre notti:
io nel vedere quest'uomo che muore,
madre, io provo dolore.
Nella pietà che non cede al rancore,
madre, ho imparato l'amore."

F. De Andrè, 2001

ISSN: 2279-8773

Just a note

I met Professor Antonio Laganà for the first time in 1999 when I did my first exam in computer science for chemists. At that time I didn't know what the destiny was going to reserve me yet but one thing attracted my attention, the huge charisma and the way of teaching proper of the Professor Laganà. Later on, at the beginning of 2004, after some problems that stop my University path, I went straightforward to the "Prof." asking to work with him in order to collect my degree in Chemistry. He accepted and one year later, in 2005, I collected the degree and started my PhD in Chemistry. As a tireless scientist he never ceased to stimulate my mind with new projects, ideas and collaborations everywhere in Europe, sometimes these was amazing, others reputed too difficult by me but I never said "no" tackling them in order to test myself and improve my know-how.

This PhD Thesis is the result of three years of research activities carried out in Italy as well as in Europe with many collaborations, sharing satisfactions as well as disappointments, in science as in life.

During my PhD I met a lot of people and I really want to thank all of them, starting from my colleagues and friends belonging to the Computational Dynamics and Kinetics group, to the Department of Chemistry in general as well as the Department of Mathematics and Computer Science at the University of Perugia and ending with the people that I met during my travels for conferences and collaborations.

A piece of my hart rested in Edinburgh when in 2005, for the first time in my life, I went abroad for a long time. If you want to improve your knowledge in computer science, I suggest you to go to the Edinburgh Parallel Computing Center where you will start to learn computer science in the proper way and immediately you fall in love with the Scottish way of life as I did.

During these years I realized that science is not a matter to put names on research articles but it is an instrument to understand events, in particular events of life, that let you able to open the doors of your mind. For this reason the research is not an easy way and neither the the editing of this Thesis but I definitely enjoyed it.

I would like also to thank all my friends, those I meet every week-end as well as those around the Europe (which say I forget them but it is not). A special thanks to Roberta for her patience and her love, sometimes the couple's life is not easy but all the others is amazing.

Since the end of 2003 I live with my father sharing difficulties as well as happiness, because a disease has deprived us of the presence of my mother but not of her memory, and I hope this Thesis give them one more reason to be proud of themselves because I am what I am thanks also to their teaching.

ISSN: 2279-8773

To conclude, many times in the past Professor Laganà scared me but today I think that sometimes I'm able to scare him or he makes me believe it...

ISSN: 2279-8773

ISSN: 2279-8773

Contents

ISSN: 2279-8773

ISSN: 2279-8773

ISSN: 2279-8773

ISSN: 2279-8773

Abstract

Molecular Dynamics is an approach suited to face several computational grand challenges which involve the determination of the behaviour of chemical system in energy, environment and technology applications. As such molecular dynamics prompts huge requests of computing resources difficult to match even using the supercomputers available at large scale computing facilities. At present, a promising alternative to the not always accessible supercomputers available at big supercomputer centers is distributed computing on Grid platforms.

The possibility of exploiting the Grid technologies and in particular the resources made available by the European Grid project has allowed us to carry out massive computational campaigns after implementing our programs on the EGEE Grid production infrastructure [1]. These efforts have materialized into the constitution of a Virtual Organization (VO) called COMPCHEM [2] whose main goal is to gather together computational researchers willing to calculate the key properties of molecular systems in an *a*b initio fashion. This is indeed the approach adopted in the present thesis with the study of three families of computational chemistry applications.

Accordingly the thesis is articulated as follows.

The first section deals with the theoretical aspects by starting from first principle equations of quantum reactive scattering to land into the classical mechanics approach. The first section continues by discussing the problem of representing molecular interactions and statistically averaging detailed information in order to evaluate observable properties and use them as molecular engines of the microscopic level of multiscale simulations.

The second section deals, instead, with the discussion of the evolution of the ICT technologies from parallel to various forms of concurrency. Then among concurrent technologies the modern distributed computing platforms are discussed in detail. In the same section the innovative features of the production Grid of EGEE and of the Virtual Organizations built on top of it, are discussed.

The third section deals instead with some computationally demanding

ISSN: 2279-8773

molecular science applications.

The first family of applications considered are those dealing with the use of quantum reactive scattering codes. Quantum reactive scattering approaches have particularly high requests of memory and for this reason can seldomly exploited for computing observable properties in realistic simulations. Moreover, most of the related codes are structured in packages complex enough to require specific ad hoc organizations to update and maintain them. There are, in fact, ad hoc organizations to support the maintenance of different Quantum Chemistry programs. However, we shall focus here more specifically on the goal of implementing grid empowered versions of quantum reactive scattering codes dealing with atom-diatom systems that is also one of the most important missions of COMPCHEM together with the design of suitable distributed workflows [3] for *ab initio* simulations of molecular systems. As part of this we illustrate in the first part of section 3 the effort spent by our CDK (Computational Dynamics and Kinetics) group to port a legacy application for atom-diatom quantum reactive scattering onto the grid infrastructure. In particular, we discuss the porting of the quantum mechanical atom-diatom reactive scattering program called ABC [4] devoted to the calculation of the detailed quantum S matrix elements. The ABC code has significant CPU demand and must be executed several times for different sets of input parameters. For this purpose it was chosen to test gridification tools of higher level like the P-GRADE Grid Portal [5, 6] and to provide a template replicable for other computationally intensive applications.

The second family of calculations considered are those concerned with the Molecular Dynamics investigation of the hydrocarbon bulk properties. In particular we investigated the properties of propane and methane which are commonly used as fuels. More recently, due to the Copenhagen amendments to the Montreal protocol [7] that has planned the ban of the use of CFCs and HCFCs before the year 2015, methane and propane are also being considered as long-term alternative refrigerants with no impact on global enviroment and in particular on stratospheric ozone depletion and global warming. For those two hydrocarbons we have carried out massive computational campaigns on the EGEE production grid. The goal is to perform a comparative analysis of calculated macroscopic properties of liquid propane and methane at some temperatures of experimental interest [8,9] by adopting two different formulations of the force field. The calculations were carried out using the DL_POLY [10] software package that is known for being native parallel, for having reasonable requests of memory and offering room for several options in dealing with molecular process. As will be discussed in detail in the second part of section 3 the gridification of DL_POLY has been carried out using the standard (and non-standard) tools of the EGEE environment

ISSN: 2279–8773

and analysing in detail the performances achieved.

The third family of computational applications considered for implementation on the grid are those of is the multiscale suite of codes modelling the production of secondary pollutants in the atmosphere. The atmosphere, in fact, represents an invaluable shared commodity and its quality is increasingly being controlled for preservation from the hazards represented by pollutants released in the atmosphere by human activities. The European Community has issued a directive (96/62/CE) [11] which recommends the Member States to adopt specific tools for controlling air quality. The recommendations include, among other suggestions, the implementation of modeling packages. The software used for this purpose , CHIMERE [12] solves simultaneously the fluid dynamics equations (for the transport of gaseous masses and dusts) and chemical equations (for the interactions between matter and light) by taking in to account the orography and weather conditions. It takes, therefore, as input data from emission inventories, weather conditions and geographical information to produce an estimate of atmospheric concentrations of several kinds of pollutants including some indirect ones like ozone (O_3) and thin dust ($PM_{2.5}$ and PM_{10}). This kind of computational simulations allow to rationalize short term episodes or long-term effects by comparing the calculated values of the pollutants with the ones measured locally. They are also used for designing recovery strategies (reduction plans and pollution control). In particular, in this Thesis we discuss the steps undertaken for implementing on the EGEE grid the chosen computational tool and discuss a study case concerning the air quality in the Umbria Region. More in detail, the case study is concerned with the secondary pollutants production during summer of the year 2004 and its impact on the Umbria territory. The work has been carried out in collaboration with ARPA (Regional Agency Prevention and Environment) Umbria [13].

ISSN: 2279-8773

ISSN: 2279-8773

Chapter 1

Molecular Dynamics

1.1 Introduction

Molecular Dynamics (MD) is an approach to the calculation of dynamics, kinetics, equilibrium and transport properties of matter at molecular level. The present connotation of the word implies a predominant usage of classical and quasiclassical means. In this context, the word *classical* means that the nuclear motion of the constituent particles are treated using the laws of classical mechanics and the prefix quasi means that approaches in which on top of a classical treatment some kind of discretization of initial and final internal energy is enforced.

MD is able to simulate in a realistic way actual ensembles of particles (representing atoms, clusters of atoms, molecules and clusters of molecules) which are treated as mass point systems. In MD dynamics and thermodynamics measurable properties of chemical systems are calculated by integrating the Newton's equations of motion of the relatedmass points starting from their positions and momenta.

Strictly speaking, however, one should describe the dynamics, kinetics, equilibrium and transport properties of matter at molecular level using quantum formulations. In quantum formulations the system is described in terms of a wavefunction that associates a spatial position with an amplitude (whose square modulus represents a probability) and expresses the observable quantities as a function of the features of the system wavefunction.

The combined simultaneous use of quantum formulations for small atomic scale (electrons and few atoms), MD formulations for large atomic and molecular scale and other non molecualar formulations for even larger scales is the approach adopted for realistic simulations of complex systems.

In this thesis we deal with all the three types of approach mentioned

ISSN: 2279–8773

above by emphasizing the related theoretical formalisms in the first chapter and focusing on the computational machinery in the second chapter. In particular we stress out here the importance of exploiting the innovative features of modern computer science especially in terms of parallel and distributed computing and the throughput advances that can be achieved.

The three types of approach are applied in the third chapter to some specific cases. The quantum treatment is applied to the calculation of the reactive properties of some atom-diatom systems to single out some peculiar dynamical features. The MD treatment is applied to the study of some thermodynamic and structural properties of methane and propane to investigate possible innovative industrial applications.

Finally the multiscale treatment is applied to the simulation of the secondary pollutant production (in particular O_3) in central Italy to offer the local public authority a means for developing environmental strategies.

1.2 The Quantum Mechanics approach

The rigorous approach to molecular dynamics is based on the solution of the Schrödinger equation. The most general formulation of the Schrödinger equation is:

$$i\hbar\frac{\partial}{\partial t}\Xi\left(\{\mathbf{W}\},\{\mathbf{w}\},t\right)=\hat{H}\Xi\left(\{\mathbf{W}\},\{\mathbf{w}\},t\right) \tag{1.1}$$

where $\Xi\left(\{\mathbf{W}\},\{\mathbf{w}\},t\right)$ is the system wavefunction and $\hat{H}$ is the related total Hamiltonian operator both for the nuclei and the electrons, t is time and $\{\mathbf{W}\}$ and $\{\mathbf{w}\}$ are the nuclear and electronic coordinate sets, respectively. The problem of working out the $\mathbf{S}$ matrix of a reactive scattering process is not an easy task, because of the impossibility of calculating the exact solution of Eq. (1.1). In this case only a numerical approach can make the problem manageable. Several methods have been developed for that purpose. Usually, the first step of this process is the separation of the motion of the center of mass. In fact, if the molecular system is not subject to external forces, the motion of the center of mass of the system does not change during the process. This implies that the total Hamiltonian can be expressed as a sum of a part describing the motion of the center of mass of the colliding system, and a part describing the relative motion of particles (once the separation of the motion of the center of mass has been done).

ISSN: 2279–8773

1.2.1 The electronic energies and wave functions

Even after separating the center of mass the complexity of the Schrödinger equation remains quite high. To this end the only efficient approach to further simplify Eq. (1.1) is the one exploiting the large difference in mass between electrons and nuclei, by adopting the Born-Oppenheimer [14] approximation. Accordingly, the total wavefunction $\Xi(\{\mathbf{W}\},\{\mathbf{w}\},t)$ is factorized into nuclear and electronic components as follows [1]:

$$\Xi(\{\mathbf{W}\},\{\mathbf{w}\},t)=\sum_n \chi_n(\{\mathbf{W}\},t)\,\Psi_n(\{\mathbf{w}\};\{\mathbf{W}\}) \tag{1.2}$$

where $\chi_n(\{\mathbf{W}\},t)$ are the nuclear wavefunctions which depend only on the nuclear coordinates $\{\mathbf{W}\}$ and time t, while $\Psi_n(\{\mathbf{w}\};\{\mathbf{W}\})$ are the electronic wavefunctions which depend on the electronic coordinates $\{\mathbf{w}\}$ and, parametrically, on the nuclear ones.

The electronic wavefunctions $\Psi_n(\{\mathbf{w}\};\{\mathbf{W}\})$ are calculated by integrating the related eigenvalue equations:

$$\hat{H}_{elec}\Psi_n(\{\mathbf{w}\};\{\mathbf{W}\})=E_n(\{\mathbf{W}\})\,\Psi_n(\{\mathbf{w}\};\{\mathbf{W}\}) \tag{1.3}$$

where $\hat{H}_{elec}$ is the electronic Hamiltonian that contains all the electronelectron and nuclei-electron terms as well as the Coulomb repulsion between the nuclei while the E_n functions (electronic or potential energy functions) are the eigenvalues associated with the Ψ_n eigenfunctions, calculated at the considered nuclear geometry.

The *ab initio* methods

Most of our reasoning about molecular electronic structure is based on the molecular-orbital (MO) method. Although this approach does not always give a correct description of the dissociative events it can be used for the accurate calculation of molecular properties near the equilibrium configuration. In general, in the MO method the total electronic wavefunctions $\Psi_n(\{\mathbf{w}\};\{\mathbf{W}\})$ (when unnecessary we shall drop from the notation the reference to the fixed nuclear geometry $\{\mathbf{W}\}$) are approximated by a sum of single electron wavefunctions $\phi(\mathbf{r})$ (called spin orbitals) functions of the vector $\mathbf{r}$ giving the position of the electron with respect to the nucleus. These

[1] Strictly speaking the $\{\mathbf{W}\}$ and $\{\mathbf{w}\}$ coordinates of Eq. (1.2) differ from those of Eq. (1.1) since in order to simplify the formalism they refer to the center of mass of the nuclei (only). However, for the sake of simplicity, let us assume that we sat from the very beginning the coordinates origin on the center of mass of the nuclei.

ISSN: 2279–8773

mono-electronic spin orbitals depend also on the other electronic spin coordinate (ξ) and can be factorized as:

$$\phi(\mathbf{r}) = \begin{cases} \phi^*(\mathbf{r})\,\alpha(\xi) \\ \text{or} \\ \phi^*(\mathbf{r})\,\beta(\xi) \end{cases} \tag{1.4}$$

By convention the spin can be up ($\alpha(\xi)$) of down ($\beta(\xi)$) meaning the keeping or the changing of sign of $\phi(\mathbf{r})$ under inversion. The $\hat{H}_e$ Hamiltonian for a non interacting electron system can be written as:

$$\hat{H}_e = \sum_{i=1}^{N} \hat{h}(i) \tag{1.5}$$

where $\hat{h}(i)$ is the one-electron operator (describing the kinetic and the potential energy of electron (i)), that in atomic units reads as:

$$\hat{h}(i) = -\frac{1}{2}\nabla_i^2 - \sum_{e=1}^{M} \frac{Z_e}{r_{ie}}. \tag{1.6}$$

The one electron operator $\hat{h}(i)$ may be alternatively considered as an effective one-electron Hamiltonian that includes the effects of electron-electron repulsion in some averaged way. The eigenfunctions of $\hat{h}(i)$ are therefore defined by the following mono-electronic equation:

$$\hat{h}(i)\phi_i(\mathbf{r}_i) = \epsilon_i \phi_i(\mathbf{r}_i). \tag{1.7}$$

In order to determine the eigenfunctions of the total Hamiltonian operator we note that $\hat{H}_e$ is a sum of one-electron Hamiltonians and thus the related eigenfunctions are simple products of spin orbitals for each electron [15]:

$$\Psi^P(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N) = \phi_i(\mathbf{r}_1)\phi_j(\mathbf{r}_2)\cdots\phi_N(\mathbf{r}_N) \tag{1.8}$$

with Ψ^P (the *Hartree product*) being an eingenfunction of $\hat{H}_e$:

$$\hat{H}_e\Psi^P = E_N\Psi^P. \tag{1.9}$$

In Eq. 1.9 E_N is the eigenvalue and can be formulated as the sum of the N spin orbital energies:

$$E_N = \epsilon_i + \epsilon_j + \cdots + \epsilon_N \tag{1.10}$$

ISSN: 2279-8773

To make Ψ^P antisymmetric with respect to the interchange of both the space and spin coordinates of any electron, so as to take into account the indistinguishability of electrons the *Slater determinant* [16] is constructed:

$$\Psi^P(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N) = (N!)^{1\frac{1}{2}} \begin{vmatrix} \phi_i(\mathbf{r}_1) & \phi_j(\mathbf{r}_1) & \cdots & \phi_N(\mathbf{r}_1) \\ \phi_i(\mathbf{r}_2) & \phi_j(\mathbf{r}_2) & \cdots & \phi_N(\mathbf{r}_2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_i(\mathbf{r}_N) & \phi_j(\mathbf{r}_N) & \cdots & \phi_N(\mathbf{r}_N) \end{vmatrix} \tag{1.11}$$

where the term $(N!)^{-\frac{1}{2}}$ is a normalization factor. Eq. 1.10 can be also written in the following form:

$$\Psi^P = \sum_P (-1)^P \hat{P} \left[\phi_i(\mathbf{r}_1)\phi_j(\mathbf{r}_2)\cdots\phi_N(\mathbf{r}_N)\right] \tag{1.12}$$

where $\hat{P}$ is the permutation operator which permutes the electrons among the orbitals ϕ. From a practical point of view, the spin orbitals are usually given as a linear combination of some monoelectronic basis functions (the related expansion is exact only when the basis is complete). These basis functions are often represented by mono-electronic atomic orbitals, each centered on a nucleus. Therefore by expressing the many-electron wavefunction as a linear combination of atomic basis functions and solving the related variational problem, it is possible to solve the electronic Schrödinger equation. Accordingly, for each nuclear geometry one obtains the corresponding electronic energy by activating the related computational procedure.

GAMESS-UK

The procedure considered here for illustrative purpose is **GAMESS** and in particular **GAMESS-UK** [17], whose developement of the code is coordinated by the Daresbury Laboratory as part of the CCP1 project. **GAMESS-UK** is based on the Hartree-Fock (HF) method that can be schematized as follows:

- Formulate Ψ^P as an independent particle model for the N considered electron, neglecting the electron-electron repulsive interaction in the Hamiltonian (see Eq. 1.5 and 1.9) by approximating it by the wavefunction Ψ_0 describing the non interacting N-electron system using a single Slater determinant built up from the $\phi(\mathbf{r})$ (including the spin functions as given in Eq. 1.4):

$$| \Psi_0 > = | \phi_1\phi_2\cdots\phi_N > \tag{1.13}$$

ISSN: 2279–8773

- Include the repulsive terms in the Hamiltonian and calculate a trial energy to approximate E_N with the wavefunction in its determinantal form. This is done by applying the variational method, that is by minimizing the energy expectation value as follows :

$$E' = \frac{\int \Psi^*_{HF} \hat{H} \Psi_{HF} d\tau}{\int \Psi^*_{HF} \Psi_{HF} d\tau} \tag{1.14}$$

where $d\tau$ represents integration over the whole space. It is important to point out that the wavefunction of Eq. 1.13 depends on the choice of the spin orbitals ϕ_i . The minimization of E' at fixed nuclei (here too the dependence of the energy on $\{\mathbf{W}\}$ is omitted to simplify the notation) with respect to the variation of the spin orbitals, leads to the so called (differential) Hartree-Fock equations. Each of these equations is a mono-electronic eigenvalue equation describing the motion of the i-*th* electron in the mean field generated by the other $N-1$ electrons:

$$\hat{f}(i)\phi(i) = \epsilon_i \phi(i) \tag{1.15}$$

where $\hat{f}(i)$ is an effective one-electron operator, called *Fock operator*:

$$\hat{f}(i) = -\frac{1}{2}\nabla_i^2 - \sum_{e=1}^{M} \frac{Z_e}{w_{ie}} + v^{HF}(i) \tag{1.16}$$

In the above equation $v^{HF}(i)$ is the average potential acting on the electron i generated by the other electrons of the system. This means that the many electron problem is replaced by several one-electron problems in which the repulsive interaction among the electrons is treated in an averaged way (Hartree-Fock approximation). Since the dependence of the spin orbital of electron i on the whole set of the other spin orbitals (see equations 1.15) is non-linear and must be solved iteratively the related procedure is known as the *Self Consistent Field* method, in which the solution is obtained iteratively until the field calculated for the $N-1$ electrons differ from that of a previous iteration less than value.

- The values of E' can be improved using post HF treatments to introduce correlation effects.

ISSN: 2279-8773

1.2.2 The nuclei dynamics

When the expansion of Eq. 1.2 is inserted into Eq. 1.1 one obtains a set of coupled differential equations in $\{\mathbf{W}\}$, $\{\mathbf{w}\}$ and t. Then, after averaging over the $\{\mathbf{w}\}$ coordinates and setting equal to zero the terms coupling $\{\mathbf{W}\}$ and $\{\mathbf{w}\}$ (Oppenheimer approximation) one obtains separate Schrödinger equations of the nuclei for each electronic surface n:

$$\hat{H}_n\chi_n(\{\mathbf{W}\},t) = i\hbar\frac{\partial}{\partial t}\chi_n(\{\mathbf{W}\},t). \tag{1.17}$$

In equation 1.17 the Hamiltonian is the sum of the kinetic $T(\{\mathbf{W}\})$ and the potential $V_n(\{\mathbf{W}\})$ terms (each $V_n(\{\mathbf{W}\})$ is an electronically adiabatic function called Potential Energy Surface or PES that is given as a set of $E_n(\{\mathbf{W}\})$ values calculated using Eq. 1.3 for a set of the corresponding $\{\mathbf{W}\}$ geometries.

obtained from the E_n values of Eq. 1.3, whose label n will be dropped when unnecessary). The $\chi_n(\{\mathbf{W}\},t)$ function is the time dependent system wavefunction depending on all the nuclear $\mathbf{W}$ coordinates. Here in after we shall specialize the formalism for atom-diatom systems (A+BC → AB+C or AC+B or A+BC) of which we discuss in chapter e some applications. For this reason in the followings we shall use the Jacobi vectors $\mathbf{R}$ and $\mathbf{r}$ (for the reactants) and $\mathbf{R'}$ and $\mathbf{r'}$ (for the products) which are illustrated in Fig. 1.1. As apparent from the figure the Jacobi vectors are arrangement (τ) dependent ($\tau = 1,2,3$ are respectively the arrangements A-BC, B-CA, C-AB though we use here the popular convention primed for products and unprimed for reactants) and lie on the molecular plane defined by the three atoms A,B,C. Accordingly the two vectors can be defined either in terms of the associated 6 cartesian projections $(x_R, y_R, z_R x_r, y_r, z_r)$, or in terms of the related moduli (R,r) and polar angles $(\theta_R, \phi_R, \theta_r, \phi_r)$, or in terms of R, r and the angle Θ they form and the three Euler angles α, β and γ which define the angles by which the frame of the spacefixed (SF), solid for example with the laboratory frame cartesian axes placed on the centre of mass of the system needs to be rotated to cincide with a given orientation with respect to the molecule on the molecular plane (Body Fixed or BF). On that plane the internuclear distances or the bond order[2] (BO) coordinates (r_{AB}, r_{BC}, r_{AC} and n_{AB}, n_{BC}, n_{AC} respectively, for the $A + BC$ atom-diatom reaction) can be defined. It has to be pointed out here, that while Jacobi coordinates are

[2]The bond order variable of the diatom ij is defined as the exponential functions of the shift of the internuclear distance r_{ij} from its equilibrium value $r_{e_{ij}}$: $n_{ij} = \exp\left[-\beta_{ij}(r_{ij} - r_{e_{ij}})\right], \beta_{ij}$ is an empirical parameter related to spectroscopic properties of the diatom (see Sec. 1.3.3 and in particular Equations 1.73 and 1.74).

ISSN: 2279–8773

orthogonal, internuclear distances and bond order coordinates are not. Yet, if we consider two internuclear distances (or two BO coordinates) and the angle Φ they form (often labeled by the central atom), this set of coordinates is suited to describe the atom-diatom reaction since it includes the coordinate of both the breaking and the forming bond.

As already mentioned, the physical observables of considered for the quantum study carried out in my thesis work are the thermal rate coefficient $k(T)$ and the cross section σ of the atom-diatom system. Rate coefficients $k(T)$ can be worked out from state-to-state rate coefficients $k_{if}(T)$ where i are the initial and f the final states of the system,

$$k(T) = \sum_{if} w_i k_{if}(T) \tag{1.18}$$

with T being the temperature of the system and w_i a weight depending on the simmetry of the considered system and processes. In bimolecular collision processes state-to-state rate coefficients can be evaluated by properly integrating the state-to-state cross section $\sigma_{if}(E_{tr})$ over the translational energy E_{tr}

$$k_{if} = \left(\frac{8}{\pi\mu k_B^3 T^3}\right)^{\frac{1}{2}} \int_0^\infty E_{tr}\sigma_{i,f}(E_{tr}) e^{-\frac{E_{tr}}{k_B T}} dE_{tr}. \tag{1.19}$$

In turn, the state-to-state cross section $\sigma_{if}(E_{tr})$ can be evaluated by summing over all the contributing values of the total angular momentum quantum number J and the properly weighed detailed reactive probabilities $P^J_{i,f}(E_{tr})$:

$$\sigma_{i,f}(E_{tr}) = \frac{\pi}{k_i^2} \sum_J (2J+1) P^J_{i,f}(E_{tr}) \tag{1.20}$$

where $P^J_{i,f}(E_{tr})$ is the fixed total angular momentum quantum number J probability obtained from the square modulus of the related **S** matrix elements. The scattering **S** matrix is therefore the quantity that contains the dynamical information we are looking for and that has to be calculated by integrating eq. 1.17 in its atom-diatom version in the Hamiltonian $\hat{H}_n$ is expressed as:

$$\hat{H}_n = -\frac{\hbar^2}{2\mu_\tau}\nabla^2_{\mathbf{S}_\tau} - \frac{\hbar^2}{2\mu_\tau}\nabla^2_{\mathbf{s}_\tau} + V(S_\tau, s_\tau, \Theta_\tau) \tag{1.21}$$

with $\mathbf{S}_\tau$ and $\mathbf{s}_\tau$ being the mass scaled analogues[3] of the Jacoby vectors $\mathbf{R}_\tau$ and $\mathbf{r}_\tau$ (non bold quantities are, as usual, the moduli of the corresponding

[3]Mass scaled Vectors $\mathbf{S}_\tau$ and $\mathbf{s}_\tau$ (not to be confused with the scattering matrix **S**) are

ISSN: 2279-8773

vector) illustrated in Fig. 1.1 and μ_τ being the atom-diatom reduced mass of the τ arrangement.

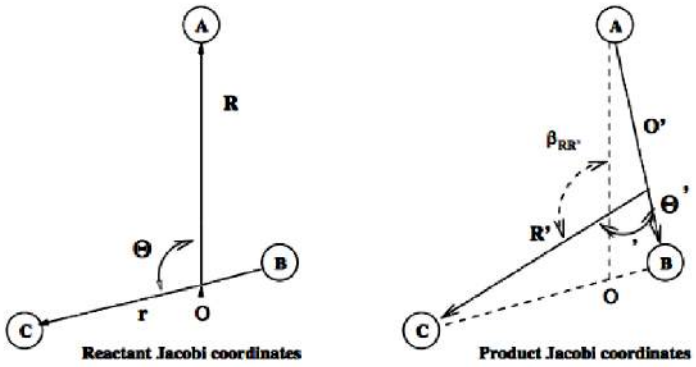

Figure 1.1: Sketch of the reactant (left hand side) and product (right hand side) Jacobi internal coordinates.

Most often, in order to take advantage of formulations in which the three-atom rigid body rotations are separated and related spherical harmonics are used, the Hamiltonian is rewritten as:

$$\hat{H}_n = -\frac{\hbar^2}{2\mu_\tau}\frac{1}{S_\tau}\frac{\partial^2}{\partial S_\tau^2}S_\tau - \frac{\hbar^2}{2\mu_\tau}\frac{1}{s_\tau}\frac{\partial^2}{\partial s_\tau^2}s_\tau + \frac{(\mathbf{J}-\mathbf{j}_\tau)^2}{2\mu_{tau}S_\tau^2} + \frac{\mathbf{j}_\tau^2}{2\mu_\tau s_\tau^2} + V(S_\tau, s_\tau, \theta_\tau) \tag{1.22}$$

where $\mathbf{J}$ is the total angular momentum operator and $\mathbf{j}$ is the rotational angular momentum operator of the diatom (whose quantum numbers are J and j respectively).

Time independent versus time dependent approaches

For the generic atom-diatom reaction:

$$A + BC(v,j) \longrightarrow AB(v',j') + C \tag{1.23}$$

obtained by multiplying and dividing the corresponding $\mathbf{R}_\tau$ and $\mathbf{r}_\tau$ vectors by an arbitrary mass factor (that can be also the atom-diatom mass itself ($\mu_\tau = m_a(m_b+m_c)/M$) or any other factor (see also pag 26)) suited to deform the physical space in a way that makes the formulation of related equations more suitable for the theoretical and computational treatment

the time dependent approach[4] sets the system wavefunction in its initial configuration in the reactant asymptotic region and then let it evolve in time under the effect of the Hamiltonian operator. The overall space-fixed (SF) wavefunction of the atom-diatom system using Jacobi reactant coordinates is usually expanded in terms of partial waves, in the total angular momentum representation:

$$\begin{aligned}\chi_i(\mathbf{R},\mathbf{r},t) &= \frac{2\pi}{k_i^{1/2}}\sum_{JMl_i} i^{l_i+1}C(j_i l_i J; m_i, M-m_i, M) \\ &\times Y^*_{l_i,M-M_i}(\hat{k}_i) \\ &\times \chi^{JM\tau_i v_i j_i l_i}(R,r,\theta_R,\phi_R,\theta_r,\phi_r,t) \end{aligned} \tag{1.24}$$

where C are the Clebsch-Gordan coefficients and $\theta_R, \phi_R, \theta_r, \phi_r$ are, as already mentioned, the angles formed by $\mathbf{R}$ and $\mathbf{r}$ with the SF axes. In Eq. 1.24 the SF partial waves $\chi^{JM}(R,r,\theta_R,\phi_R,\theta_r,\phi_r,t)$ can be expanded in terms of the Body Fixed (BF) partial waves as follows (for simplicity the labels $\tau_i v_i j_i$ and l_i have been dropped from the notation):

$$\chi^{JM}(R,r,\theta_R,\phi_R,\theta_r,\phi_r,t) = \sum_{\Lambda} \psi^{J\Lambda p}(R,r,\Theta,t)\frac{1}{Rr}\hat{D}^{Jp}_{\Lambda M}(\alpha,\beta,\gamma) \tag{1.25}$$

where J is the total angular momentum quantum number, M is the space-fixed z component of total angular momentum, Λ is the body-fixed z component of total angular momentum, p is the parity, $\hat{D}^{Jp}_{\Lambda M}$ is the Wigner rotation function, α, β, γ are the already mentioned three Euler angles characterizing the rotation of the space-fixed axes into the body-fixed axes and R, r and Θ are the three internal coordinates of Fig. 1.1. In general, the time dependent approach to the calculation of the $\mathbf{S}$ matrix elements is articulated in the following steps:

1. the representation of the initial wavepacket
2. the action of the Hamiltonian on the wave packet
3. the analysis of the wavepacket in the product region and reaction attributes

In the followings more details of the three steps are given.

[4]The formalism illustrated here is the one of S.K. Gray and G.G Balint Kurti [28]

ISSN: 2279–8773

1. The body-fixed wavepacket, $\psi^{J\Lambda}(R, r, \Theta, t)$, is initially defined in the reactant asymptotic region as:

$$\begin{aligned} \psi^{J\Lambda}(R, r, \Theta, t) &= \left(\frac{8\alpha}{\pi}\right)^{\frac{1}{4}} e^{-\alpha(R-R_0)^2} \\ &\times e^{-ik(R-R_0)}\ \varphi_{vj}^{BC}(r)\ P_j^{\Lambda}(\Theta). \end{aligned} \tag{1.26}$$

In eq. 1.26, $e^{-ik(R-R_0)}$ is a phase factor which gives the initial wavepacket a relative kinetic energy towards the interaction region, $\varphi_{vj}^{BC}(r)$ is the initial diatomic molecule BC wavefunction (for the vibrational state v and the rotational state j) expressed in the Jacobi coordinates of the reactant arrangement, $P_j^{\Lambda}(\Theta)$ is the normalized associated Legendre polynomials and k is the wavevector determining the relative kinetic energy of the collisional partners. In this way, the wavefunction is defined for a given accessible state of the reactants and a given collisional energy range. When one is interested in computing the properties of the products the wavefunction has to be mapped onto the final diatomic molecule AB wavefunctions $\varphi_{v'j'}^{AB}(r')$ and the related r' coordinate has to be used. Accordingly, the wavepacket, initially set up in reactant Jacobi coordinates R, r and Θ, is transformed into the product ones R' , r' and Θ' . The transformation can be carried out at any step of the propagation though it may require extra attention if it is not performed far in the reactant asymptotic region. Even without carrying out such a transformation (i.e. by using reactant coordinates only) it is still possible to compute total reaction probabilities (yet not the state-to-state one). In fact, by subtracting the non reactive flux to the total one (there is no need for computing detailed state-to-state flux into product channels for that) one obtains the reactive flux.

2. To carry out the propagation, in our implementation of the time dependent technique, the radial part of the wavepacket is collocated on a regularly spaced grid on R and r (or R' and r') Jacobi reactant (or products) coordinates (in Fig. 1.2 a typical grid domain in mass scaled product Jacobi coordinates is shown). The angular component of the wavepacket is instead expanded on a Gauss-Legendre quadrature grid [18]. The R, r grid needs to be large enough to contain the initial wavepacket and to properly describe the wavepacket during the its evolution in time including the asymptotic product region where the analysis line (R'_∞ in the figure) is drawn as well as the strong interaction region. At grid edges, an absorption region (see dashed region of

ISSN: 2279–8773

Fig. 1.2) is also introduced to prevent the wavepacket amplitude from being reflected back (the so called aliasing effect) into the interaction region as the wavepacket approaches the grid edges [19, 20]. Once the wavepacket has been collocated on the grid the propagation is started. The evaluation of $\hat{H}\psi^{J\Lambda}$ is the most time consuming operation of each time propagation step, as well as the most memory consuming. The operation consists of two separate parts: the evaluation of $\hat{T}\psi^{J\Lambda}$ and of $\hat{V}\psi^{J\Lambda}$, where, as already mentioned, $\hat{T}$ is the kinetic energy operator and $\hat{V}$ the potential energy one. The evaluation of the potential term is a relatively simple task: in fact, this is a local operation being the potential energy operator diagonal in the representation that makes use of Jacobi coordinates. Therefore, the $\hat{V}\psi^{J\Lambda}$ term is obtained by multiplying the wavepacket at each grid point i, j, k by $V(R_i, r_j, \Theta_k)$. The evaluation of the action of the kinetic energy operator on the wavepacket represents the key step of the propagation since the $\hat{T}\psi^{J\Lambda}$ term is not local in our representation. The effect of applying the R component ($\hat{T}_R$) of the kinetic operator is expressed as:

$$\hat{T}_R\psi^{J\Lambda}(R, r, \Theta, t) = \frac{1}{2\mu_R}\frac{\partial^2}{\partial R^2}\psi^{J\Lambda}(R, r, \Theta, t). \tag{1.27}$$

This operation can be performed by first Fourier transforming $\psi^{J\Lambda}(R, r, \Theta, t)$

$$\begin{aligned}\psi^{J\Lambda}(R, r, \Theta, t) &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\exp(-ikR)\psi^{J\Lambda}(R, r, \Theta, t)dR \\ &= fft\{\psi^{J\Lambda}(R, r, \Theta, t)\}\end{aligned} \tag{1.28}$$

to obtain the momentum representation of the wavepacket and then by multiplying the Fourier transform by $\frac{k^2}{2\mu_R}$ that is the representation of the kinetic operator in the momentum space. This is a local operation at each grid point of momentum space. Finally, one performs a Fourier transform. Accordingly, one obtains:

$$\begin{aligned}\hat{T}_R\psi^{J\Lambda}(R, r, \Theta, t) &= \frac{1}{2\mu_R}\frac{\partial^2}{\partial R^2}\psi^{J\Lambda}(R, r, \Theta, t) \\ &= fft^{-1}\left\{\frac{k^2}{2\mu_R}\ fft\{\psi^{J\Lambda}(R, r, \Theta, t)\}\right\}.\end{aligned} \tag{1.29}$$

A similar procedure can be adopted for the r component ($\hat{T}_r$). As to the angular component ($\hat{T}_\Theta$) one has the operator:

$$(\hat{T}_\Theta) = \left\{ -\frac{1}{\sin\Theta}\frac{\partial}{\partial\Theta}\ \sin\Theta\ \frac{\partial}{\partial\Theta} \right\} \tag{1.30}$$

whose eigenfunctions are the Legendre polynomials in $\cos\Theta$:

$$\left\{ -\frac{1}{\sin\Theta}\frac{\partial}{\partial\Theta}\ \sin\Theta\ \frac{\partial}{\partial\Theta} \right\}\ P_j(\cos\Theta) = j(j+1)\ P_j(\cos\Theta) \tag{1.31}$$

The grid representation of $\hat{T}_\Theta$ is:

$$T_{l,i} = \sum_{j=0}^{N_\Theta} U_{l,j}(j(j+1))U^t_{j,i} \tag{1.32}$$

where **U** is the matrix that transforms the Legendre polynomial [21] basis set to the grid representation,

$$U_{l,j} = P_j(\cos\Theta_i)\sqrt{w_i} \tag{1.33}$$

where w_i are the Gauss-Legendre weights

$$w_i = \sin(\Theta_i). \tag{1.34}$$

The action of the angular part of the kinetic energy operator at a given point l is therefore formulated as:

$$\{\hat{T}_\Theta\Psi\}_l = \left(\frac{1}{2\mu_R R^2} + \frac{1}{2\mu_r r^2} \right) \sum_i^{N_\Theta} T_{l,i}\Psi_i \tag{1.35}$$

An alternative way of dealing with the R and r components of the kinetic operator, is the discrete variable representation (DVR) that was originally introduced in the area of molecular reaction dynamics by Light and coworkers [21,22] for solving the time dependent Schrödinger equation. This is at present increasingly being used [23–27] in the evaluation of the Hamiltonian operation on the wavefunction in time dependent calculations. The basic idea behind the DVR method is to expand the wavefunction in an orthonormal basis set $\{\Phi_i(x); i = 1, N\}$, and use a quadrature rule, usually Gaussian, consisting of a set of quadrature points $\{x_i; i = 1, N\}$ and weights $\{w_i; i = 1, N\}$ to define

ISSN: 2279–8773

the inner product of these basis vectors. In one dimension, for example, is

$$\Psi(x,t) = \sum_{i}^{N} a_i(t)\Phi_i(x) \tag{1.36}$$

where

$$a_i(t) = \int \Phi_i^*(x)\Psi(x,t)dx. \tag{1.37}$$

Since we are dealing with a discrete Hilbert space rather than a continuous one, the above integral can be approximated by an N-point quadrature rule, provided the basis functions are still orthonormal in the discrete representation. Thus we have

$$a_i(t) = \sum_{j}^{N} w_i\Phi_i^*(x_j)\Psi(x_j,t) \tag{1.38}$$

where w_j is the quadrature weight associated with the grid point x_j. The DVR can be compared to the Fourier method since a discrete Hilbert space is employed in both these representations. Since a finite Fourier series is used in the Fourier method to expand the wavefunction, the quadrature points become uniform grid points, $x_i = i\Delta x, i = 0,\ldots,N-1$. The quadrature weight w_i is just the grid spacing Δx and the numerical quadrature is equivalent to the trapezoidal rule, which is of Gaussian quadrature accuracy for periodic functions. Computationally, the DVR method scales as N^2 compared to $N\log N$ of the FFT method. The advantage of the DVR approach is that, unlike the FFT one, the wavefunction needs not to be transformed to the momentum space forth and back every time step, avoiding, so far, a considerable overhead. Furthermore, it is possible to keep at a minimum the number of basis functions required in the expansion if an appropriate basis is chosen according to the nature of the potential. A feature of this method is that only the real component of the wavepacket may be propagated [28] (it is possible to demonstrate that a two step propagation of the real part of the wavepacket is equivalent to a single step propagation of the complex wave). In this case we can exploit the relationships:

$$q(x,t) = Re[\psi(x,t)] \tag{1.39}$$

ISSN: 2279–8773

$$p(x,t) = Im[\psi(x,t)] \tag{1.40}$$

to introduce q and p which are always real-valued functions with x being the reaction coordinate. Accordingly, the initial wavefunction $\psi(x,t_0)$ can be expressed as:

$$\psi(x,t_0) = q(x,t_0) + ip(x,t_0) \tag{1.41}$$

where, as stated above, q is the real component and p is the imaginary one. Therefore, the propagation in time of the initial value of the real part q of the wavepacket can be formulated as:

$$q(x,t_0+\Delta t) = \hat{H}_s q(x,t_0) - \sqrt{1-\hat{H}_s^2}\; p(x,t_0) \tag{1.42}$$

with $\hat{H}_s = a_s\hat{H} + b_s$ being a scaled and shifted Hamiltonian operator such that its minimum and maximum eigenvalues lie between -1 and 1. We can obtain subsequent values of q by repeatedly applying the recurrence formula

$$q(x,t+\Delta t) = -q(x,t-\Delta t) + 2\hat{H}_s q(x,t) \tag{1.43}$$

for discrete steps of Δt. More in general, the application of the Hamiltonian to the wavepacket is performed by implementing some propagation schemes. In our case use is made of the Tchebyshev polynomial expansion method. Tal-Ezer and Kosloff [29] introduced a global propagation scheme, based on the Tchebyshev polynomial expansion of the evolution operator. Since the Tchebyshev polynomial is bounded in the interval (-1,1), the Hamiltonian has to be renormalized by shifting its eigenvalues to this range. This can be achieved if we have a rough estimate of the eigenvalues of the Hamiltonian. Assuming the Fourier method for the spatial discretisation, the maximum energy represented on the grid is given by

$$E_{max} = \frac{\hbar^2\pi^2}{2m(\Delta x)^2} + V_{max} \tag{1.44}$$

and the minimum energy

$$E_{min} = V_{min} \tag{1.45}$$

Using the mean value

ISSN: 2279-8773

$$\bar{E} = \frac{1}{2}(E_{max} + E_{min}) \tag{1.46}$$

the Hamiltonian can be renormalized to shift its eigenvalues to (-1,1)

$$\hat{H}_{norm} = (\hat{H} - \hat{1}\bar{E})/\Delta E \tag{1.47}$$

with

$$\Delta E = \frac{1}{2}(E_{max} - E_{min}) \tag{1.48}$$

We then rewrite the evolution operator as

$$\exp\left(-i\frac{\hat{H}t}{\hbar}\right) = \exp\left(-i\frac{\bar{E}t}{\hbar}\right)\exp\left(-i\frac{\Delta E t\hat{H}_{norm}}{\hbar}\right) \tag{1.49}$$

and expand it in a complex Tchebyshev series, as follows:

$$\exp\left(-i\frac{\hat{H}t}{\hbar}\right) = \exp\left(-i\frac{\bar{E}t}{\hbar}\right)\sum_{n=0}^{\infty} C_n J_n\left(\frac{\Delta E t}{\hbar}\right) T_n\left(-i\hat{H}_{norm}\right) \tag{1.50}$$

where $C_n = 1$ for $n = 0$ and $C_n = 2$ for $n \geq 1$. The J_n's are Bessel functions of the first kind of order n. $T_n(-i\hat{H}_{norm})$ are complex Tchebyshev polynomials satisfying the recurrence relation

$$\psi_{n+1} = -2i\hat{H}_{norm}\psi_n + \psi_{n-1} \tag{1.51}$$

The attractive feature of the method is that the Bessel function $J_n(\alpha)$, with $\alpha = \frac{\Delta E t}{\hbar}$, falls off to zero exponentially for $n > \alpha$. This happens rapidly for larger α and one needs to include only a few extra terms above $n \geq \alpha$ to get convergence. Thus one achieves the desired accuracy in the calculation by including a sufficient number of terms above the theoretical limit $n = \alpha$. The method is well suited for long time propagation as there is no restriction on the step size for t. One can even complete the entire time evolution in a single time step (the only loss in information being about intermediate results, which may be needed, if one is interested in the dynamics during the course of the collision). A major inadequacy of the method is that it cannot be used with explicitly time dependent Hamiltonians, as one has to include a proper time ordering. Thus the Fourier discretisation for the spatial

ISSN: 2279–8773

part together with the Tchebyshev expansion of the evolution operator forms a highly efficient algorithm for solving the time dependent scattering equations.

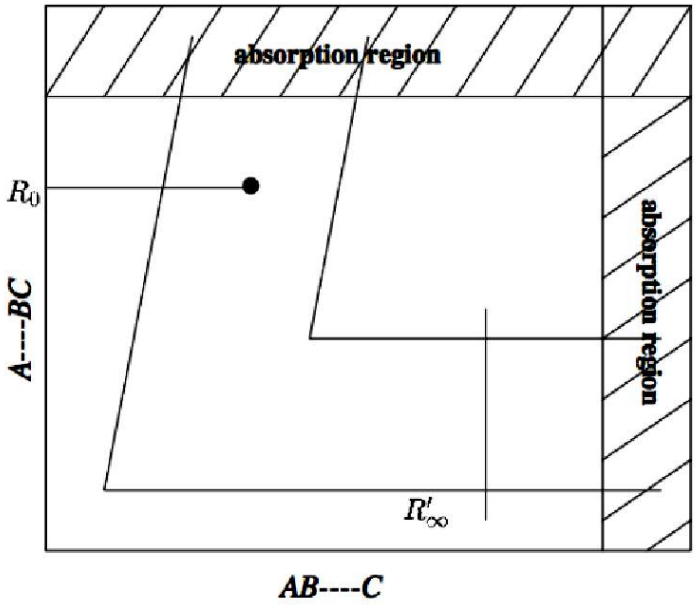

Figure 1.2: A schematic representation of the grid domain in the mass scaled product Jacobi coordinates.

3. As already mentioned, at each time step the wavepacket is expanded at the analisys line in terms of the final diatomic molecule AB wavefunction $\phi^{AB}_{v'j'}(r')$. The time dependent coefficients of the expansion are obtained by the following integral:

$$C^J_{vj\Lambda,v'j'\Lambda'}(t) = \int_{r'} dr' \int_{\Theta'} d\Theta' \sin\Theta' \times P_{j'\Lambda'}(\Theta')\phi_{v'j')}(r')\psi^{J\Lambda'}(R' = R'_\infty, r', \Theta', t) \tag{1.52}$$

The C coefficients are then Fourier transformed in order to obtain the time independent (and energy-dependent) elements of the **A** matrix (whose square modulus represent the reaction probability):

$$A_{vj\Lambda,v'j'\Lambda'}(E) = \frac{1}{2\pi}\int_{t=0}^{\infty} dt \exp(iEt/\hbar)\ C_{vj\Lambda,v'j'\Lambda'}(t) \tag{1.53}$$

the **S** matrix elements are then calculated as follows, in terms of the **A** matrix:

$$S^J_{vj\Lambda,v'j'\Lambda'}(E) = \left(\frac{k_{vj}k_{v'j'}}{\mu\mu'}\right)^{\frac{1}{2}} \frac{\hbar A^J_{vj\Lambda,v'j'\Lambda'}}{g(-k_{vj})} \exp(-ik_{v'j'}R'_\infty) \tag{1.54}$$

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

The time independent approach

Time independent techniques are conceptually more complex than time dependent ones since they require two computational steps: the calculation of a lower dimensionality basis set and the propagation along the reaction coordinate of the scattering coupled differential equations. Our implementation is based on the adiabatically adjusting[5] principal axis of inertia hyperspherical (APH) coordinates [30] formalism. The hyperspherical coordinates are the hyperradius ρ, the two internal hyperangles θ and χ_τ. As already mentioned, the three internal coordinates can be expressed directly in terms of the mass-scaled Jacobi coordinates:

$$\begin{aligned} \mathbf{s}_\tau &= d_\tau^{-1}\mathbf{r}_\tau \\ \mathbf{S}_\tau &= d_\tau \mathbf{R}_\tau \end{aligned} \tag{1.55}$$

with the scaling factor being defined as:

$$d_\tau = \left[\frac{m_\tau}{\mu} - \left(1 - \frac{m_\tau}{M}\right)\right] \tag{1.56}$$

and the reduced triatomic mass as

$$\mu = \left(\frac{m_i m_j m_k}{M}\right)^{1/2}. \tag{1.57}$$

Accordingly, the hyperradius is defined as:

$$\rho = (S_\tau^2 + s_\tau^2)^{1/2} \tag{1.58}$$

and the hyperangles are defined as:

$$\tan\theta = \frac{[(S_\tau^2 - s_\tau^2)^2 + (2\mathbf{S}_\tau \cdot \mathbf{s}_\tau)^2]^{1/2}}{2S_\tau^2 s_\tau^2 \sin\Theta_\tau} \tag{1.59}$$

[5] for the adiabatic nature of the coordinates see pag 48

ISSN: 2279–8773

$$\sin(2\chi_\tau) = \frac{2\mathbf{S}_\tau \cdot \mathbf{s}_\tau}{[(S_\tau^2 - s_\tau^2)^2 + (2\mathbf{S}_\tau \cdot \mathbf{s}_\tau)^2]^{1/2}} \tag{1.60}$$

and

$$\cos(2\chi_\tau) = \frac{(2S_\tau^2 - s_\tau^2)}{[(S_\tau^2 - s_\tau^2)^2 + (2\mathbf{S}_\tau \cdot \mathbf{s}_\tau)^2]^{1/2}} \tag{1.61}$$

These coordinates exhibit several interesting features. The hyperradius ρ is an ideal continuity variable since it smoothly connects the collapsed particle geometry to all possible fragmented geometries and represents the size of the three particle system (ρ is independent of the arrangement channel). The angle θ is a bending angle (collinear configurations are at $\theta = \pi/2$ and $\theta = 0$ represents equilateral triangular configurations for systems with three equal masses. θ is also independent of the arrangement channel). The angle χ_τ is a kinematic rotation angle. The integration of the time independent Schrödinger equation over the reaction coordinate is performed by partitioning ρ into several sectors, labeled by the index i, and by expanding the system wavefunction, within each sector, in an appropriate basis set. At a given parity p the basis functions of the expansion consist of a product of the surface functions $\Phi_{q\Lambda}^{Jp}(\theta, \chi_\tau; \rho_i)$ of the two internal hyperangles times the Wigner rotation functions $\hat{D}_{\Delta M}^{Jp}(\alpha, \beta, \gamma)$ of the three Euler angles and the unknown functions Jpn $\psi_{q\Lambda}^{Jpn}(\rho)$ of the hyperradius ρ. The surface functions are obtained by using an analytic basis method (ABM) to solve the following two dimensional bound state problem (it is also possible to use other methods like the DVR or the FEM methods):

$$\begin{aligned} \Big[& \hat{T}_h + \frac{15\hbar^2}{8\bar{\mu}\rho_i^2} + \hbar^2 J(J+1)(A+B)/2 + C \\ - & \frac{(A+B)}{2\hbar^2\Delta^2} + V(\rho_i, \theta, \chi_\tau) - \varepsilon_{q\Delta}^{Jp}(\rho_i) \Big] \\ \cdot & \Phi_{q\Lambda}^{Jp}(\theta, \chi_\tau; \rho_i) = 0 \end{aligned} \tag{1.62}$$

where q is an ordering index,

$$\begin{aligned} A^{-1} &= \mu\rho_i^2(1 + \sin\theta) \\ B^{-1} &= 2\mu\rho_i^2 \sin^2\theta \\ C^{-1} &= \mu\rho_i^2(1 - \sin\theta) \end{aligned} \tag{1.63}$$

ISSN: 2279-8773

μ is the already seen hyperspherical reduced mass, $V(\rho_i, \theta, \chi_\tau)$ is the potential energy and $\varepsilon^{Jp}_{q\Lambda}(\rho_i)$ is the q-th eigenvalue. In the same equation the hyperangular term $\hat{T}_h$ reads

$$\hat{T}_h = -\frac{\hbar^2}{2\mu\rho_i^2}\left(\frac{4}{\sin 2\theta}\frac{\partial}{\partial\theta}\sin\theta\frac{\partial}{\partial\theta} + \frac{1}{\sin^2\theta}\frac{\partial^2}{\partial\chi_\tau^2}\right) \tag{1.64}$$

When substituting the surface functions into the above mentioned stationary Schrödinger equation for the nuclei one obtains a set of coupled differential equations having the form:

$$\left(\frac{\partial}{\partial\rho^2} + \frac{2\mu E}{\hbar^2}\right)\phi^{Jpn}_{q\Lambda}(\rho) = \frac{2\bar{\mu}}{\hbar^2}\sum_{q'\Lambda'}\langle\Phi^{Jp}_{q\Lambda}\hat{D}^{Jp}_{\Lambda M} \mid \hat{H}_i \mid \Phi^{Jp}_{q'\Lambda'}\hat{D}^{Jp}_{\Lambda' M}\rangle\ \psi^{Jpn}_{q'\Lambda'}(\rho) \tag{1.65}$$

with

$$\hat{H}_i = \hat{T}_h + \hat{T}_r + \hat{T}_c + \frac{15\hbar^2}{8\bar{\mu}\rho_i^2} + V(\rho, \theta, \chi_\tau) \tag{1.66}$$

and the rotational $\hat{T}_r$ and the Coriolis $\hat{T}_c$ terms being respectively defined as:

$$\hat{T}_r = A(\rho_i, \theta)J_x^2 + B(\rho_i, \theta)J_y^2 + C(\rho_i, \theta)J_z^2 \tag{1.67}$$

and

$$\hat{T}_c = -\frac{i\hbar\cos\theta}{\bar{\mu}\rho_i^2\sin^2\theta}J_y\frac{\partial}{\partial\chi_\tau} \tag{1.68}$$

The propagation of the partial waves takes place under the form of the Wigner **R** matrices starting from very small values of ρ and proceeding through the strong interaction region to the asymptotes (large ρ values). There, the final value of **R** is conveniently mapped into the final states using the Jacobi coordinates of the proper arrangement to evaluate the **S** matrix elements by imposing the proper asymptotic boundary conditions.

1.3 The Classical Mechanics approach

As already mentioned usual MD approaches to the evaluation of the properties of large molecular systems are based on the integration of classical mechanics equations.

ISSN: 2279–8773

This makes Molecular Dynamics a deterministic technique: given an initial set of positions and velocities, the subsequent time evolution is *in principle*[6] completely determined. The computer calculates a trajectory of the system in a $6N$-dimensional phase space ($3N$ positions and $3N$ momenta).

However, the classical mechanics approach of MD has intrinsic limits which can be singled out in a simple way by defining the de Broglie thermal wavelength [31]:

$$\Lambda = \sqrt{\frac{2\pi\hbar}{M\kappa_B T}} \tag{1.69}$$

where M is the mass of the system. The classical approximation is valid when $\Lambda \ll a$, where a is the average interparticle distance. For a gas this quantity is approximately $(V/N)^{1/2}$ where V is the volume and N is the number of particles. When the thermal de Broglie wavelength is much smaller than the interparticle distance, the gas can be considered to be a classical or Maxwell-Boltzmann gas. On the other hand, when the thermal de Broglie wavelength is of the order of (or larger than) the interparticle distance, quantum effects will dominate and the gas must be treated as a Fermi gas or a Bose gas, depending on the nature of the gas particles. The critical temperature is the transition point between these two regimes. At the critical temperature, the thermal wavelength is approximately equal to the interparticle distance. That is, the quantum nature of the gas will be evident for:

$$\frac{V}{N\Lambda^3} \leq 1 \tag{1.70}$$

and in this case the gas will obey Bose-Einstein statistics or Fermi- Dirac statistics, whichever is appropriate. On the other hand, for

$$\frac{V}{N\Lambda^3} \gg 1 \tag{1.71}$$

the gas will obey Maxwell-Boltzmann statistics. The classical approximation is poor for light systems and when T is sufficiently low. Molecular dynamics results should be interpreted with caution in these regions. There are however other intrinsic limits in molecular dynamics simulations. In fact in MD atoms interact with each other and these interactions originate forces which act upon atoms, and atoms move under the action of these instantaneous forces. As the atoms move, their relative positions change and forces change as well. Forces are usually obtained as gradient of the potential energy function and especially depend on the particle positions. The accuracy

[6] in practice, the finiteness of the integration time step and of the electronic representation of numbers (in other words the arithmetic rounding) might cause the computed trajectory deviate from the true one.

ISSN: 2279–8773

of the simulation therefore is very sensitive to the suitability of the potential chosen to describe the interaction of the components of the system. MD simulations are performed on systems typically containing thousands, millions or even billions of atom. Simulation times on their side can range, under appropriate conditions, from picoseconds to microseconds. A simulation is "safe" with respect to its minimum duration when duration in time is much longer than the relaxation time of the quantities of interest. However, different properties may have different relaxation times. In particular, systems tend to become slow and sluggish in the proximity of phase transitions, and it is not uncommon to find cases where the relaxation time of a physical property is orders of magnitude larger than times achievable by computer simulations. On the contrary, there is a problem of "safety" also with respect to its maximum duration. In fact, the error accumulation at each time step may sum up to an amount comparable with the integration variables when the number of steps is exceedingly large. The size of the system can also constitute a problem. For example in the case in which one has to compare the size of the MD cell with the correlation lengths of the spatial correlation functions of interest there may be problems when the size of the cell is too small. Correlation lengths, in fact, may increase or even diverge in proximity of phase transitions. Therefore, the result may become no longer reliable when the correlation length becomes comparable with the box length. This problem can be partially alleviated by a method known as finite size scaling. This consist of computing a physical property A using several boxes having different sizes L, and then fitting the results using the relationship:

$$A(L) = A_\circ + \frac{c}{L^n} \tag{1.72}$$

with $A_\circ$, c and n being the best fit parameters. $A_\circ$ then corresponds to $\lim_{L\to\infty} A(L)$, and should therefore be taken as the best estimate for the physical quantity. Despite the above discussed limitations most of the observable can be satisfactorily evaluated [32, 33].

1.3.1 Equations of motion

There are indeed several ways of describing classical systems with the most general of them being the Lagrangian method.

Lagrangian dynamics

In the Lagrangian approach the equations of motion are expressed using an ensemble of generalized coordinates $\{\mathbf{q}(t)\}$. These coordinates do not have

ISSN: 2279–8773

to be coordinates in the usual sense of spanning a high dimensional space (like Cartesian coordinates). They should be seen just as general variables. Given the kinetic energy K and the potential energy V, the Lagrangian L of the system is defined as:

$$L(\dot{\mathbf{q}}(t), \dot{\mathbf{q}}(t), t) = K - V \tag{1.73}$$

where $\dot{\mathbf{q}}(t)$ is the derivative of $\mathbf{q}$ with respect to time:

$$\dot{\mathbf{q}}(t) = \frac{d\mathbf{q}(t)}{dt} \tag{1.74}$$

The kinetic and the potential term can both be a function of $\dot{\mathbf{q}}$, $\mathbf{q}$ and t explicitly. Lagrange's equations of motion are:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial \dot{q}} = 0, i = 1, \ldots, N \tag{1.75}$$

The elegance of this formulation is that it is completely independent of the choice of coordinates $\mathbf{q}$. This is convenient when the potential energy is known in some appropriate coordinate system. For Cartesian coordinates $\mathbf{q} = \mathbf{x}$, with the kinetic energy being given by:

$$K(\dot{\mathbf{x}}) = \sum_i \frac{m_i}{2}(\dot{x}_i^2) \tag{1.76}$$

and the potential energy depending only on the coordinates, Lagrange equations of motion reduce to Newton's equations of motion:

$$m_i \frac{d^2 x_i}{dt^2} = f_i \tag{1.77}$$

where the forces are defined as minus the gradient of the potential:

$$f_i = -\frac{\partial}{\partial x_i} V(\mathbf{q}) \tag{1.78}$$

Hamiltonian dynamics

The Lagrange formulation is most useful for solving the equations of motion, but for statistical mechanics the Hamiltonian formulation is more convenient. The main conceptual difference between the Lagrangian and the Hamiltonian formulation of the classical dynamics is that in the latter the generalized velocities $\dot{q}_i$ are replaced by the generalized momenta momenta p_i:

$$p_i = \frac{\partial L}{\partial \dot{x}} \tag{1.79}$$

ISSN: 2279–8773

The generalized momenta are said to be conjugate coordinates of the generalized positions. The Hamiltonian H is a function of the generalized momenta and coordinates and it can depend explicitly on time:

$$H(\mathbf{p}(t), \mathbf{q}(t), t) = \sum_i p_i \dot{q}_i - L(\dot{\mathbf{q}}(t), \mathbf{q}(t), t) \tag{1.80}$$

The equations of motion can than be derived from this definition:

$$\dot{q}_i = \frac{\partial H}{\partial \dot{p}_i} \tag{1.81}$$

$$-\dot{p}_i = \frac{\partial H}{\partial \dot{q}_i} \tag{1.82}$$

$$\frac{\partial L}{\partial t} = \frac{\partial H}{\partial t} \tag{1.83}$$

Equations (1.81) and (1.82) are called the canonical equations of Hamilton. If the generalized coordinates can be expressed as a function of Cartesian coordinates, the kinetic energy will be a quadratic function of the generalized velocities, with coefficients which depend on the generalized coordinates, but not explicitly on time:

$$K = \frac{1}{2} \sum_{ij} a_{ij}(\mathbf{q}) \dot{q}_i \dot{q}_j \tag{1.84}$$

$$a_{ij} = \sum_k m_k \frac{\partial x_k}{\partial q_i} \frac{\partial x_k}{\partial q_j} \tag{1.85}$$

When the potential is not a function of the generalized velocities, the momenta are given by:

$$p_i = \frac{\partial L}{\partial \dot{q}_i} = \frac{\partial K}{\partial \dot{q}_i} = \sum_j a_{ij}(\mathbf{q}) \dot{q}_i \tag{1.86}$$

Using (1.84) it can be shown that under these conditions the Hamiltonian is equal to the total energy:

$$H = K + V \tag{1.87}$$

From (1.87) we can see that when the potential does not depend explicitly on time the total energy is conserved. Such systems are called conservative.

ISSN: 2279–8773

1.3.2 Molecular Dynamics Simulations

The first paper reporting a molecular dynamics simulation [34] was devoted to the investigation of the phase diagram of a hard sphere system, and in particular the separation between solid and liquid regions. The flowchart of the related programs is quite simple and can be schematized as follows:

1. Read in the parameters specifying the conditions of the run (e.g. initial temperature, number of particles, density, time step).
2. Initialize the system (i.e., select initial positions and velocities).
3. Compute the forces on all particles.
4. Integrate Newton's equations of motion (i.e. move the particles one step forward). Go back to the previous step if the desired length of time has not yet been covered. Otherwise go to next step.
5. Compute and print the averages of measured quantities, and stop.

Initialization

To start the simulation, one has to assign initial positions and velocities to all particles in the system. These initial positions should be compatible with the structure to be simulated. In any event, the particles should be positioned away from locations in which there is an appreciable overlap among atomic or molecular cores. Often this is achieved by initially placing the particles at the equilibrium positions of a cubic lattice.

The Force Calculation

As already mentioned the calculation of the force acting on every particle is the most time-consuming part of almost all Molecular Dynamics simulations. If the case of a model system described by pairwise additive interactions is considered the contributions to the force on particle i due to all its neighbours have to be taken into account. If we consider only the interaction between a particle and the nearest image of another particle, for a system of N particles, we must evaluate $N \times (N-1)/2$ pair distances; if we use no tricks[7], the time needed for the evaluation of the forces scales as N^2. Therefore, we must compute the force between these particles, and the contribution to the

[7]There exist efficient techniques (as we shall see later) to speed up the evaluation of both short-range and long-range forces in such a way that the computing time scales as N , rather than as N^2

ISSN: 2279–8773

potential energy. For instance the x-component of the force calculated with the $u(r)$ pair potential is

$$f_x(r) = -\frac{\partial u(r)}{\partial x} = -\left(\frac{x}{r}\right)\left(\frac{\partial u(r)}{\partial r}\right) \tag{1.88}$$

The main ingredient of a simulation is a model for the physical system. Forces are derived as gradients of the potential with respect to atomic displacements:

$$\mathbf{F}_i = -\nabla_{\mathbf{r}_i} V(\mathbf{r}_1, ..., \mathbf{r}_N) \tag{1.89}$$

This form implies the presence of a conservation law of the total energy $E = K + V$ (where K is the instantaneous kinetic energy). For a molecular dynamics simulation there is, therefore, the need for choosing a function $V(\mathbf{r}_1, ..., \mathbf{r}_N)$ of the positions of the nuclei, representing the potential energy of the system when the atoms are arranged in that specific configuration. This function is translationally and rotationally invariant, and is usually constructed from the relative positions of the atoms with respect to each other, rather than from the absolute positions. As in the example given in the followings, the simplest choice is to write V as a sum of pairwise interactions:

$$V(\mathbf{r}_1, ..., \mathbf{r}_N) = \sum_i \sum_{j>i} \phi(|\mathbf{r}_i - \mathbf{r}_j|) \tag{1.90}$$

with the clause $j > i$ in the second summation imposed by the need of considering each atom pair only once. This is also the most frequently adopted representation of the potential in spite of the fact that it is a very poor approximation for processes leading to bond forming and breaking (like reactions) and systems having highly delocalized bonds (like metals and semiconductors. For this reason wherever is necessary various kinds of many-body corrections are currently used.

Potential truncation and long-range corrections

In practical applications, it is customary to establish a cutoff radius r_c and disregard the interactions between atoms separated by more than r_c . This results in simpler programs and enormous savings of computer resources, because the number of atomic pairs separated by a distance r grows as r^2 and rapidly becomes very large. Since at r_c the interaction is not zero, it is clear that particles entering from outside r_c into inside region exhibit a jump in potential and forces. Formally, the force at the cutoff distance

ISSN: 2279–8773

($\mathbf{F}_{ij} = -\nabla u(r_{ij})$) is infinitely large, since the potential exhibits a step due to truncation. This sudden acceleration of particles usually leads to a heating of the system, since the motion is not reversible. Consider, e.g. a two-particle system, where one particle comes from outside and moving with velocity v_0 to an interparticle distance $r = r_c - \epsilon$, with $\epsilon \ll 1$. Then it gets an abrupt force contribution, accelerating it, that leads to a velocity $v_1 > v_0$. On the other hand, if a particle starts from $r_c - \epsilon$ with velocity v_1 to leave the sphere with radius r_c, then it still has velocity $v_1 > v_0$. For a many-particle system this means that $\langle v^2 \rangle$ increases due to many crossings and recrossings of the interaction sphere, i.e. the temperature increases. In order to avoid this statistical effect, one may introduce smoothing functions, which continuously drop the potential and the forces to zero. The disadvantage with this approach is that there will be a zone of large forces at the cutoff distance if the forces are properly evaluated as derivatives of the potential. For this reason one has to smooth forces in this region, leading however to a non-conservative system. Therefore a different method is most often used, which consists in shifting the whole potential and force by a certain amount, which guarantees that both the potential and the force are exactly zero at the cutoff distance, i.e.

$$u^{(sfp)}(r_{ij}) = \begin{cases} u(r_{ij}) - u(r_c) + (r_{ij} - r_c)F(r_c) & \text{if } r_{ij} \leq r_c \\ 0 & \text{if } r_{ij} > r_c \end{cases}$$

$$\mathbf{F}^{(sfp)}(\mathbf{r}_{ij}) = \begin{cases} \mathbf{F}(\mathbf{r}_{ij}) - F(r_c)\hat{\mathbf{r}}_{ij} & \text{if } r_{ij} \leq r_c \\ 0 & \text{if } r_{ij} > r_c \end{cases}$$

This ensures a smooth transition from outside to inside the cutoff region and vice versa. Commonly used truncation radii for the Lennard-Jones potential are 2.5 σ and 3.2 σ. It should be mentioned here that these truncated Lennard-Jones models[8] are so popular that they acquired a value on their own as reference models for generic two-body systems. In many cases, there is no interest in evaluating truncation corrections because the truncated model itself is the subject of the investigation. Potentials for metals and semiconductors are usually designed from the start with a cutoff radius in mind, and go usually to zero at r_c together with their first two derivatives. Such potentials do not therefore contain true van der Waals forces. Since such forces are much weaker than those associated with metallic or covalent bonding, this is usually not a serious problem except for geometries with two or more separate bodies.

[8]Lennard-Jones functional forms are discussed in pag. 43

ISSN: 2279–8773

The numerical integration of the Equations of Motion

Several algorithms have been designed to integrate Newton's equations of motion. The most popular algorithm used in MD is the Verlet one [35], which is both simplectic (time reversible) and simple in nature [36]. To derive it, one can start with a Taylor expansion of the coordinate of a particle, around time t,

$$r(t+\Delta t) = r(t) + v(t)\Delta t + \frac{f(t)}{2m}\Delta t^2 + \frac{\Delta t^3}{3!}\frac{d^3r}{dt^3} + \mathcal{O}(\Delta t^4)$$

similarly,

$$r(t-\Delta t) = r(t) - v(t)\Delta t + \frac{f(t)}{2m}\Delta t^2 - \frac{\Delta t^3}{3!}\frac{d^3r}{dt^3} + \mathcal{O}(\Delta t^4).$$

Summing these two equations, one obtains

$$r(t+\Delta t) + r(t-\Delta t) = 2r(t) + \frac{f(t)}{2m}\Delta t^2 + \mathcal{O}(\Delta t^4) \tag{1.91}$$

or

$$r(t+\Delta t) \approx 2r(t) - r(t-\Delta t + \frac{f(t)}{2m}\Delta t^2 \tag{1.92}$$

The estimate of the new position contains an error that is of order Δt^4 , where Δt is the time step integration. Note that the Verlet algorithm does not use the velocity to compute the new position. Velocity can be however derived from the knowledge of the trajectory, using

$$r(t+\Delta t) - r(t-\Delta t) = 2v(t)\Delta t + \mathcal{O}(\Delta t^3) \tag{1.93}$$

or

$$v(t) = \frac{r(t+\Delta t) - r(t-\Delta t)}{2\Delta t} + \mathcal{O}(\Delta t^2) \tag{1.94}$$

This expression for the velocity is only accurate to order Δt^2 . However, it is possible to obtain more accurate estimates of the velocity, and thereby of the kinetic energy, using a Verlet-like algorithm (i.e., an algorithm that yields trajectories identical to that given by equation 1.92). After each time step, we can compute the current temperature, the current potential energy calculated in the force loop, and the total energy.

ISSN: 2279–8773

Periodic Boundary Conditions and Neighbour Lists

In order to keep the samples of particles as small as possible, if effects at the surface of the sample are not extremely important we need to impose boundary conditions. Consider 1000 atoms arranged in a $10 \times 10 \times 10$ cube. More than half the atoms are on the outer faces, and these will have a large effect on the measured properties. Even for $10^6 = 100^3$ atoms, the surface atoms amount to 6% of the total, which is still nontrivial. Surrounding the cube with replicas takes care of this problem. Provided the potential range is not too long, we can adopt the minimum image convention that each atom interacts with the nearest atom or image in the periodic array. In the course of the simulation, if an atom leaves the basic simulation box, it is replaced by the incoming image. This is illustrated in the scheme of Fig. 1.3.

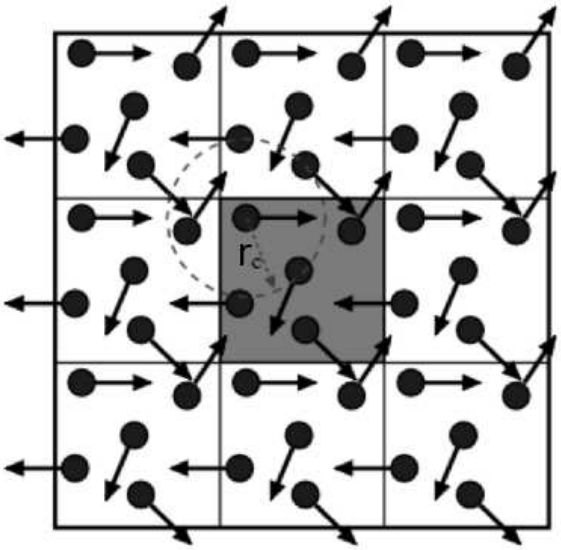

Figure 1.3: Periodic boundary conditions. As a particle moves out of the simulation box, an image particle moves in to replace it. In calculating particle interactions within the cutoff range (r_c), both real and image neighbours are included.

Of course, it is important to bear in mind the imposed artificial periodicity when considering properties which are influenced by long-range correlations. Computing the non-bonded contribution to the interatomic forces in an MD simulation involves, in principle, a large number of pairwise calculations: we consider each atom i and loop over all other atoms j to calculate the minimum image separations r_{ij} . Let us assume that the interaction potentials are of

ISSN: 2279-8773

short range, $v(r_{ij}) = 0$ if $r_{ij} > r_c$. In this case, the program skips the force calculation, saving on computing time and considers the next candidate j. Nonetheless, the time spent to examine all pair separations and to calculate for each of them r_{ij}^2 is still considerable being proportional to the number of distinct pairs, $\frac{1}{2}N(N-1)$ in an N-atom system. Some economies result from the use of lists of nearby pairs of atoms. Verlet [36] suggested such a technique for improving the speed of a program. The potential cutoff sphere, of radius r_c , around a particular atom is surrounded by a "skin", to give a larger sphere of radius r_{list} as shown in Fig. 1.4.

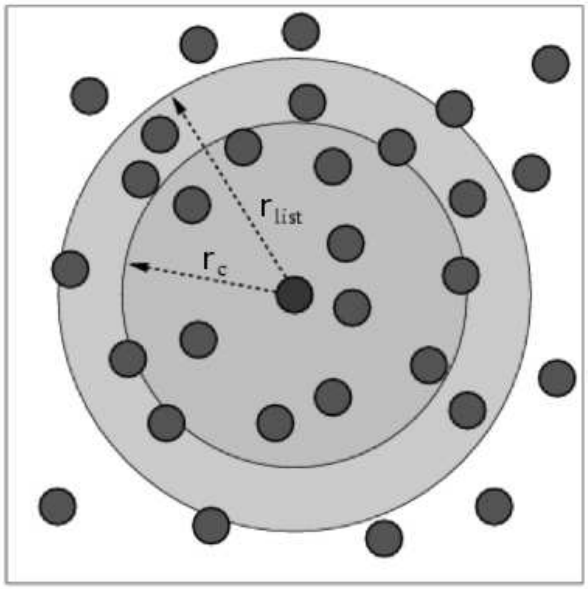

Figure 1.4: A graphical depiction of the cutoff distances in a Verlet neighbour list.

At the first step in a simulation, a list is constructed of all the neighbours of each atom, for which the pair separation is within r_{list} . Over the next few MD time steps, only pairs appearing in the list are checked in the force routine. From time to time the list is reconstructed: it is important to do this before any unlisted pairs have crossed the safety zone and come within interaction range. It is possible to trigger the list reconstruction automatically, if a record is kept of the distance travelled by each atom since the last update. The choice of list cutoff distance r_{list} is a compromise: larger lists will need to be reconstructed less frequently, but will not give as much of a saving on cpu time as smaller lists. This choice can easily be made by experimentation. For larger systems ($N \geq 1000$ or so, depending on the potential range) other

techniques may become preferable. A popular technique considers a cubic simulation box (extension to non-cubic cases is possible) that is divided into a regular lattice of $n_{cell} \times n_{cell} \times n_{cell}$ cells (see Fig. 1.5).

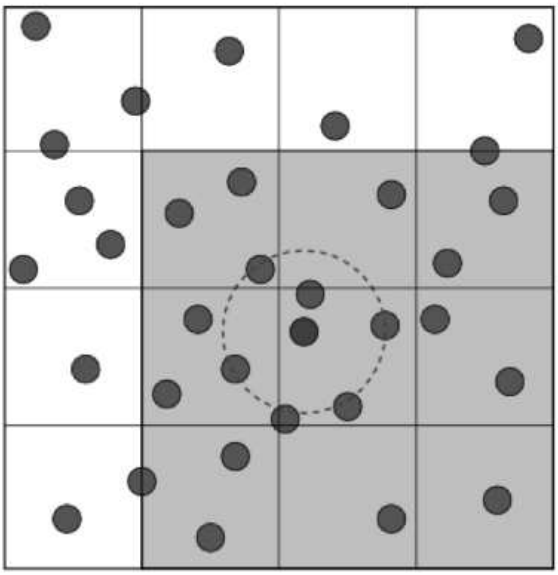

Figure 1.5: The cell structure. The potential cutoff range is indicated. In searching for neighbours of an atom, it is only necessary to examine the atom's own cell, and its nearest-neighbour cells (shaded).

These cells are chosen so that the side of the cell $l_{cell} = L/n_{cell}$ is greater than the potential cutoff distance r_c . If there is a separate list of atoms in each of these cells, then searching through the neighbours is a rapid process: it is only necessary to look at atoms in the same cell as the atom of interest, and in nearest neighbour cells. The cell structure may be set up and used by the method of linked lists. The first part of the method involves sorting all the atoms into their appropriate cells. This sorting is rapid, and may be performed every step. Then, within the force routine, pointers are used to scan through the contents of cells, and calculate pair forces. This approach is very efficient for large systems with short-range forces. A certain amount of unnecessary work is done because the search region is cubic (and not, as for the Verlet list, spherical).

ISSN: 2279–8773

The validity of a MD simulation

It is obvious that a good Molecular Dynamics program requires a good algorithm to integrate Newton's equations of motion. In this sense, the choice of algorithm is crucial. Let us look at the different points to consider. Accuracy for large time steps is more important, because the longer the time step that we can use, the fewer evaluations of the forces are needed per unit of simulation time. Hence, this would suggest that it is advantageous to use a sophisticated algorithm that allows the use of a long time step. Energy conservation is an important criterion, but actually we should distinguish two kinds of energy conservation, namely, short time and long time. The sophisticated higher-order algorithms tend to have very good energy conservation for short times (i.e., during a few time steps). However, they often have the undesirable feature that the overall energy drifts for long times. In contrast, Verlet-style algorithms tend to have only moderate short term energy conservation but little long-term drift. It would seem to be most important to have an algorithm that accurately predicts the trajectory of all particles for both short and long times. As a matter of fact, no such algorithm exists. Essentially all MD simulations fall in the regime where the trajectory of the system through phase space (i.e., the 6N -dimensional space spanned by all particle coordinates and momenta) critically depend on the initial conditions (this means that two trajectories which are initially very close will diverge exponentially as time progresses). We can consider the integration error caused by the algorithm as the source for the initial small difference between the "true" trajectory of the system and the trajectory generated in our simulation. We should expect that any integration error, no matter how small, will always cause our simulated trajectory to diverge exponentially from the true trajectory having the same initial conditions. This so-called Lyapunov instability would seem to be a devastating blow to the whole idea of MD simulations. Yet, it is not so serious, because MD simulations do not aim to predict precisely what will happen to a system that has been prepared in a precisely known initial condition. MD is always interested in providing statistical estimates and therefore to predict the average behavior of a system prepared in an initial state about which we know something (e.g., the total energy) but by no means everything. Still, this would not justify the use of inaccurate trajectories unless the trajectories obtained numerically, in some sense, are close to true trajectories. For example, since Newton's equations of motion are time reversible, so should be our algorithms. In algorithms which are not time reversible instead future and past phase space coordinates do not play a symmetric role. As a consequence, if one were to reverse the momenta of all particles at a given instant, the system would not trace back its trajectory in

ISSN: 2279–8773

phase space, even if the simulation would have been carried out with infinite numerical precision. Another crucial aspect is that Hamilton's equation of motion leave the magnitude of any volume element in phase space unchanged. Several numerical schemes, in particular those which are not time reversible, do not reproduce this area-preserving property. After sufficiently long times, we expect that the non-area-preserving algorithm will have greatly expanded the volume of our system in phase space. This is not compatible with energy conservation. Hence, it is plausible that nonreversible algorithms will have serious long-term energy drift problems. Finally, it should be noted that even when integrating a time-reversible algorithm, we shall find that the numerical implementation is hardly ever truly time reversible. This is due to the finite machine precision of a computer with using a given floating-point arithmetic that results in rounding errors (on the order of the machine precision). In this respect, the Verlet algorithm scores on these points. First of all, the Verlet algorithm is fast (though we had argued that this is relatively unimportant). Second, it is not particularly accurate for long time steps and therefore needs to compute the forces on all particles rather frequently. Third, it requires about as little memory as is at all possible (though this is useful when simulating very large systems, in general it is not a crucial advantage). Fourth, its short-term energy conservation is fair (in particular in the versions that use a more accurate expression for the velocities) but, more important, it exhibits little long-term energy drift. This is related to the fact that the Verlet algorithm is time reversible and area preserving.

1.4 The Potential Energy Surface formulation

By repeating the calculation at several values of $\{\mathbf{W}\}$ (different nuclear geometries covering all the relevant configuration space) one generates a multidimensional matrix giving a pointwise representation of the potential energy function. As already mentioned, this ensemble of values (or their functional representation) is usually called Potential Energy Surface (PES) of the system and is indicated as $V(\{r_{ij}\})$ (where i and j are two generic atoms of the system and r_{ij} is the related internuclear distance defined as $r_{ij} =\mid W_j - W_i \mid$). This is the surface (or better the hypersurface) on which the motion of the nuclei takes place. In priciple, potential energy values should be evaluated using *ab initio* methods at all geometries relevant to the calculation of the dynamics of the system. This computational task is not affordable even for small chemical systems. More often *ab initio* calculations are performed for

ISSN: 2279–8773

a reasonably large grid of nuclear positions (say at 10 points per variable) since extended regions of the molecular geometries may become accessible during the collision. Regions of interest include intermediate wells, barriers, ridges as well as long range and asymptotic reactant and product regions. Sometimes, further specific sets of points need to be added, in order to better define some critical features of the interaction, like minimum energy paths and saddles to reaction making the procedure deal with a fairly large amount of *ab initio* values and their fitting using a proper means.

Recently, the so called Shepard [37] interpolation, that belongs to the family of local[9] interpolations, has become increasingly popular. The Shepard method formulates the PES as a Taylor series expansion $T_p(\{\mathbf{R}\})$ (truncated to the second order) around each ab initio point p. Accordingly, the value of the potential at the $\{r_{i,j}\}$ geometry of interest is obtained by summing of the values obtained on allpoints p from the corresponing truncated Taylor series multiplied by the each weight of $w_p(\{r_{i,j}\}))$ associated with the proximity of the point p to that of the geometry of interest

$$V(\mathbf{R}) = \sum_{p=1}^{P} w_g(\mathbf{R}) T_p(\mathbf{R}) \tag{1.95}$$

where P is the number of points p at which *ab initio* calculations have been performed. Collins and coworkers [38] use for the Taylor expansion the inverse of the internuclear distance. This has the advantage of better enforcing the asymptotic behavior of the isolated diatomic potential since a Taylor expansion in the inverse coordinates has a larger domain of convergence for diatomic potentials than the corresponding expansions in the coordinates themselves. Another local method is the Moving Least Square (MLS) one [39]. In the MLS technique a least square fit restricted to the subset of ab initio values falling within a given closeness to the geometry of interest is performed.

More consolidated are the so called global methods in which a proper functional representation of the interaction is adopted and its parameters are adjusted to minimize the difference between the fitted and the available *ab initio* values $V(\{r_{ij}\})$. The resulting PES can then be used to investigate in detail the topology of the potential channels to single out the characteristics of the potential which determine the dynamics of the process. An advantage of adopting the procedure of fitting the *ab initio* points using a

[9]The methods can be categorized as Local Methods and Global Methods. In the first the PES is determined at each point based only on *ab initio* values which are available for geometries close to that point. In the second the PES at each point is determined by all the *ab initio* values which are input to the fit.

ISSN: 2279–8773

functional form is associated with the fact that this allows a fast evaluation of the potential energy and its derivatives at any given geometry accessed during dynamical calculations. Moreover it simplifies the incorporation of corrections into *ab initio* points when they are found to be inadequate. However, the above described fitting energy, all reasonably well applicable to small systems, become difficult to apply to large systems. They are however important also for large systems since the most popular procedures for fitting their PES expand the potential into few body terms.

1.4.1 Potential Energy Surface for small systems

The simplest global functional forms used for fitting the PES of the atom systems are either derived from drastically simplified algebraic expressions of the *ab initio* formulation of the interaction (such as LEPS [40, 41]) or from empirical models (such as diatomic rotating potentials [42, 43]). However, both these functional forms have shown to have built in some strong limitations (bias for certain geometries, difficulty to be generalized, etc. . .) which have prevented their generalized use.

A popular approach stems out from the manybody expansion method, also known as Sorbie-Murrell method that is a global method able, in principle, to fit potential energy surfaces for reactions involving any number of atoms [44]. The expression of the many-body expansion for a three-atom system is given as a sum of one-body, two-body and three-body terms:

$$\begin{aligned} V(r_{AB}, r_{BC}, r_{AC}) &= V_A^{(1)} + V_B^{(1)} + V_C^{(1)} + \\ &\quad V_{AB}^{(2)}(r_{AB}) + V_{BC}^{(2)}(r_{BC}) + V_{AC}^{(2)}(r_{AC}) + \\ &\quad V_{ABC}^{(3)}(r_{AB}, r_{BC}, r_{AC}) \end{aligned} \tag{1.96}$$

The one-body terms, $V_A^{(1)}$, $V_B^{(1)}$ and $V_C^{(1)}$, are the electronic energies of the atoms in the dissociation configurations. Since we normally deal only with electronic ground state potential energy surfaces these terms are set equal to zero:

$$V_A^{(1)} = 0; V_B^{(1)} = 0; V_C^{(1)} = 0 \tag{1.97}$$

Functional forms used for fitting higher order terms are polynomials in the relevant coordinates multiplied by a damping function of the exponential type. Popular two body functional forms in MD studies are the Lennard-Jones (n, 6) ones which read

$$V^{LJ}(r_{ij}) = 4\varepsilon_{ij}\left[(\frac{\sigma}{r_{ij}})^{12} - (\frac{\sigma}{r_{ij}})^{6}\right] \tag{1.98}$$

or in the equivalent form

$$V^{LJ}(r_{ij}) = \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} \tag{1.99}$$

The alternative approach followed in our group is to make use of the Bond Order (BO) variables n. The BO variable n_{ij} for the ij atom is related to the internuclear distance r_{ij} as follows:

$$n_{ij} = \exp[-\beta_{ij}(r_{ij} - r_{eq,ij})] \tag{1.100}$$

In Eq. 1.100 β_{ij} is an empirical parameter and $r_{eq,ij}$ is the equilibrium distance for the ij diatom. If the potential energy curve of a diatomic molecule is plotted against the related BO coordinate, it has a parabolic shape (see Fig. 1.6, where the plot of a diatomic potential as a function of the internuclear distance (right hand side panel) and as a function of the related BO coordinate (left hand side panel) is given).

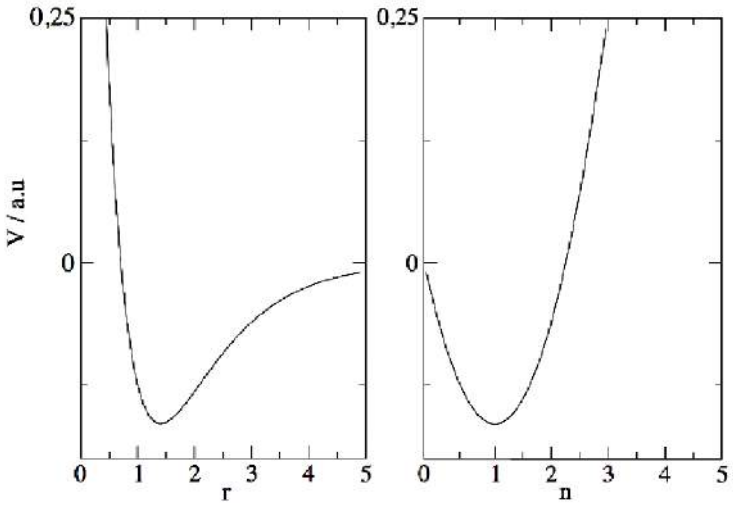

Figure 1.6: An attractive-repulsive diatomic potential represented as a function of the internuclear distance r (left hand side panel) and as a function of the corresponding BO coordinate n (right hand side panel).

The figure singles out some peculiar properties of the BO formulation of the diatomic interaction:

- the minimum of a diatomic potential energy curve is always located at $n = 1(r_{ij} = r_{eq,ij})$, as shown in Fig. 1.7 for both the reactant and the product fragments of H + ICl system;
- in physical coordinate plots the dissociation limit (A + B) ($r_{ij} \to \infty$) lies outside the graph and the repulsive region is located at short distance. On the contrary, in BO coordinates the dissociation limit is located at the coordinate origin and the repulsive potential is at large (though finite) values of n_{ij}.

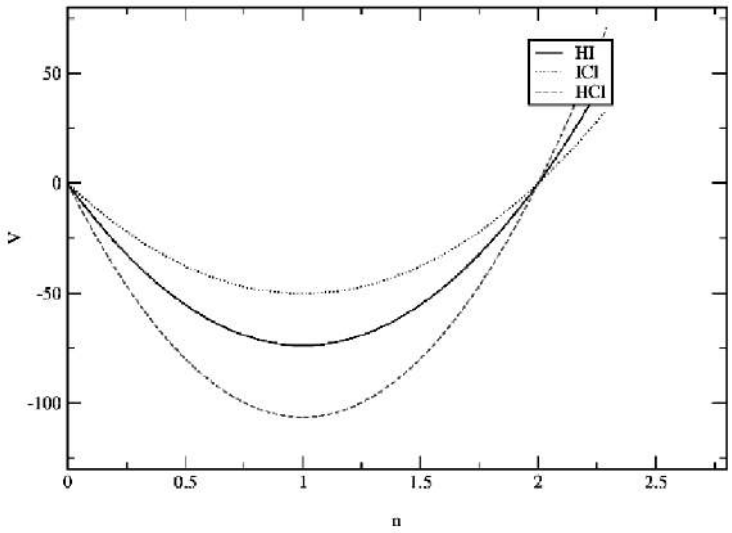

Figure 1.7: Shape of the three asymptotic (diatomic) curves of the LEPS PES of the $HICl$ system plotted as a function of the related BO variables.

These considerations evidence the advantage of using BO coordinates for modeling potential energy functions: the zero asymptotic limit is naturally built into the BO representation of the potential, the finiteness of the space makes it easier to build grids, minimum energy paths (MEPs) are easier to locate.

The two body component

When using BO coordinates, the two-body terms are expressed as follows [45]:

$$V_{ij}^{(2)}(n_{ij}) = D_{ij} \sum_{k=0}^{N} a_{ij}^{(k)} n_{ij}^{k} \qquad ij = AB, BC, AC \tag{1.101}$$

where D_{ij} is the dissociation energy of the ij diatom, while the a_{ij} are the expansion coefficients. These coefficients, jointly with the β_{ij} parameter of Eq. 1.100, are chosen so as to minimize the rms deviation of the fitted values from the input points (i.e. the *ab initio* values). It is obvious that, when the expansion of Eq. 1.101 is truncated to $k = 2$, the BO potential becames the usual Morse potential if β_{ij} coincides with the Morse β_{ij} parameter

$$\beta_{ij} = w_{e,ij}\sqrt{\frac{\pi\mu_{ij}}{D_{ij}}} \tag{1.102}$$

(with $w_{e,ij}$ being the harmonic vibrational constant of and μ_{ij} the reduced mass of the considered diatom) and a_1 and a_2 have the values 2 and -1, respectively.

When in Eq. 1.101 some higher terms are included the parameters of the BO functional loose the physical meaning associated with the parameters of the Morse potential. If expansion 1.101 is truncated to the fourth power, a relationship between the BO coefficients and the force constants of the diatom can still be established. The method called (FCBO) equates the derivatives of the potential at the minimum to the related force constants provided by the spectroscopy. This leads to the following system of equations (for simplicity we drop the index ij):

$$\begin{aligned} c_1 &= \frac{1}{6}(G_3/\beta^3) + (G_2/\beta^2) + 4 \\ c_2 &= -\frac{1}{2}(G_3/\beta^3) - 4(G_2/\beta^2) - 6 \\ c_3 &= \frac{1}{2}(G_3/\beta^3) + (G_2/\beta^2) + 4 \\ c_4 &= -\frac{1}{6}(G_3/\beta^3) - (G_2/\beta^2) - 1 \\ 0 &= G_4 + 10G_3\beta + 32G_2\beta^2 + 24\beta^4 \end{aligned} \tag{1.103}$$

where $G_k = -F_k/D_e$ with F_k being the k-*th* force constant of the diatom ij. The fifth equation is solved numerically in order to work out the β parameter. In general, such an equation, has only one positive solution. In the few cases when multiple one positive solutions are available the largest one, in better agreement with extrapolation from solutions obtained by second and third power truncation of the expansion, is adopted.

The three body term

After working out the diatomic terms, the three body interaction is estimated out of the calculated *ab initio* values by subtracting them the two body com-

ISSN: 2279–8773

ponents (after modifying the *ab initio* values to reproduce the main features of the reaction channel, including spectroscopic and reactivity properties). Then the three body values are fitted using a polynomial of order M in the relevant three BO variables:

$$V^{(3)}_{ABC}(n_{AB}, n_{BC}, n_{AC}) = \sum_{l=0}^{M}\sum_{m=0}^{M}\sum_{n=0}^{M} c_{lmn} n_{AB}^{l} n_{BC}^{m} n_{AC}^{n} \qquad (1.104)$$
$$l + m + n \leq M,$$
$$l + m + n \neq k \neq m \neq n$$

in which all single variable terms are excluded. It is worth noting that the values of the β parameters used for in Eq. 1.104 are still the same as those estimated for the two body terms. This fact implies that the surface has the correct asymptotes and that the spurious features which can arise in the long range region when other functional forms are used for diatomic and three body interaction terms do not showing in our case. Moreover, in this formulation the damping function, normally used in the many-body expansion to make sure that the three body term becomes zero at asymptotic configurations (i.e. when any of the internuclear distances becomes large), is not needed because the three-body term is zero when either n_{AB} or n_{BC} or n_{AC} is zero (i. e. for atom+diatom and full dissociation configurations). The optimization of the c_{lmn} coefficients is performed using a linear regression. An example of the N+N_2 potential energy surface used in section 3 drown using BO coordinates is shown in Fig. 1.8.

The HYBO coordinates and their relaxed variable

Picture of the PES like the one of Fig. 1.8 have suggested the use of a different type of BO coordinates which go under the name of Hyperspherical BO (HYBO) coordinates. HYBO coordinate for the process A + BC → AB + C (HYBO coordinates as the BO ones are process coordinates) are defined as:

$$\rho = \sqrt{n_{AB}^2 + n_{BC}^2} \qquad (1.105)$$
$$\alpha = \arctan(n_{BC}/n_{AB})$$

with Φ_B being the angle formed by n_{AB} and n_{BC} (the same as the angle formed by the internuclear distances [47]). These coordinates describe the

ISSN: 2279-8773

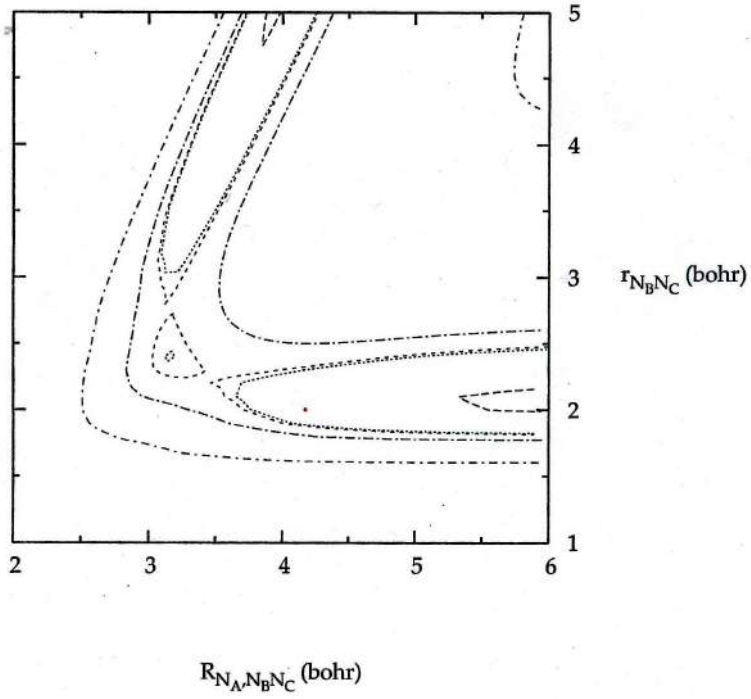

Figure 1.8: Isoenergetic contours of the N + N_2 BOPES.

reaction channel as the channel formed by a rotating pseudo diatomic potential whose parameters vary with Φ_B and the reaction coordinate α. The corresponding functional representation

$$V(\rho, \alpha, \Phi_B) = D(\alpha, \Phi_B) \left[\frac{\rho^2}{\rho_0(\alpha, \Phi_B)} - \frac{\rho}{\rho_0(\alpha, \Phi_B)} \right] \tag{1.106}$$

where D and ρ_0 are appropriate functions of α and Phi_B ensuring the symmetry properties of the system as well as the reproduction of the asymptotic and strong interaction features of the *ab initio* (if this is the case also modified) values. Switch from one process to the others is usually obtained using suitable switching functions [47] bears therefore the advantage not only of not needing damping functions but also of making use of simple functions to smoothly connect reactant to product asymptotes. The HYBO coordinates have also shown recently to be ideally suited for relaxed variable representations of the reaction paths [48]. In fact the plot of the potential minimum along ρ found at fixed Φ and α values allows a proper representation of different reactive paths when plotted as a function of these two angles. Attempts

to use these coordinates for carrying out dynamical calculations are under way [48].

AP and APH Adiabatic coordinates

Another type of relazex (adiabatic) more traditional set of coordinates more based towards adequate the description of the geometries of the strong interaction region are the APH hyperspherical coordinates already mentioned in page 26. Jacobi coordinates can in fact be transformed into a set of coordinates which adiabatically evolve from a geometry to another by continuously redirecting R_τ along the principal axis of inertia (AP, Adiabatically adjusting Principal axis of inertia). In fact, if we consider the arbitrary angle χ_τ of the kinematic rotation (called kinematic angle) that transforms R_τ and r_τ in the generic vectors $\mathbf{Q}$ and $\mathbf{q}$:

$$\begin{pmatrix} \mathbf{Q} \\ \mathbf{q} \end{pmatrix} = \mathbf{T}(\chi_\tau) \begin{pmatrix} \mathbf{R}_\tau \\ \mathbf{r}_\tau \end{pmatrix}, \tag{1.107}$$

we can choose the value of χ_τ which maximizes $\mathbf{Q}$ and minimizes $\mathbf{q}$. Such angle is defined by the relationships:

$$\sin(2\chi_\tau) = \frac{2\mathbf{R}_\tau \cdot \mathbf{r}_\tau}{\left[(R_\tau^2 - r_\tau^2)^2 + (2\mathbf{R}_\tau \cdot \mathbf{r}_\tau)^2\right]^{1/2}} \tag{1.108}$$

$$\cos(2\chi_\tau) = \frac{(R_\tau^2 - r_\tau^2)}{\left[(R_\tau^2 - r_\tau^2)^2 + (2\mathbf{R}_\tau \cdot \mathbf{r}_\tau)^2\right]^{1/2}} \tag{1.109}$$

The formulation of Q and q in terms of the Jacobian coordinates is therefore:

$$Q = \left(\frac{R_\tau^2 + r_\tau^2}{2} + \frac{\left[(R_\tau^2 - r_\tau^2)^2 + (2\mathbf{R}_\tau \cdot \mathbf{r}_\tau)^2\right]^{1/2}}{2} \right)^{1/2} \tag{1.110}$$

$$q = \left(\frac{R_\tau^2 + r_\tau^2}{2} - \frac{\left[(R_\tau^2 - r_\tau^2)^2 + (2\mathbf{R}_\tau \cdot \mathbf{r}_\tau)^2\right]^{1/2}}{2} \right)^{1/2} \tag{1.111}$$

Even if equations 1.110 and 1.111 are labeled after τ, Q and q do not depend on a specific arrangement. In fact, the angles χ_τ for different arrangements differ only for a given phase.

The choice of this particular angle lets the coordinate $\mathbf{Q}$ tend to the Jacobian coordinate $\mathbf{R}_\tau$ when the latter gets large (that is when the atom τ lies far away from the diatom). This behaviour is important, given the specificity

ISSN: 2279–8773

of the Jacobian coordinates to identfy a given arrangement. Unfortunately, however, q does not tend to r_τ at the same time. As a consequence, the coordinate system ($\mathbf{Q}$, $\mathbf{q}$) is not that suitable to describe the asymptotic zone of an atom-diatom process. However, out of the AP coordinates one can define the already utilized (see page 26) APH coordinates (Adiabatically adjusting Principal axis of inertia Hypersferical coordinates) which are the symmetric version of the hyperspherical coordinates particularly suited for the treatment of the strong interaction region. The expressions that links the APH coordinates to the AP ones are:

$$\begin{aligned} \rho^2 &= Q^2 + q^2, \\ \theta &= \frac{\pi}{2} - 2\arctan q/Q, \end{aligned} \tag{1.112}$$

where $\rho \in [0, \infty)$ is the hyperradius and $\theta \in [0, \pi/2]$ is one of the two hyperangles when specializing $\mathbf{Q}$ and $\mathbf{q}$ as $\mathbf{S}_\tau$ and $\mathbf{s}_\tau$. The remaining hyperangle, $\chi_\tau \in (-\pi/2, \pi/2]$, is such that maximizes $\mathbf{Q}$.

The coordinate ranges cover the upper half of a sphere and can be visualized as polar spherical coordinates of the internal space. The line undefined in χ_τ is now the line $\theta = 0$ which, like in polar spherical coordinates, does not cause any problem.

The three APH coordinates can also be directly expressd as functions of the Jacobian coordinate (as seen in page 26) and related inverse formulae are:

$$\mathbf{S}_\tau = \frac{\rho}{\sqrt{2}}\{1 + \sin\theta\cos[2(\chi_i - \chi_{\tau i})]\}^{1/2}, \tag{1.113}$$

$$\mathbf{s}_\tau = \frac{\rho}{\sqrt{2}}\{1 - \sin\theta\cos[2(\chi_i - \chi_{\tau i})]\}^{1/2} \tag{1.114}$$

and

$$\cos\Theta_\tau = \frac{\sin\theta\sin[2(\chi_i - \chi_{\tau i})]}{\{1 - \sin^2\theta\cos^2[2(\chi_i - \chi_{\tau i})]\}^{1/2}}. \tag{1.115}$$

1.4.2 The Potential Energy Surface for large systems

The use of Force Fields

For larger systems the construction of the interaction represents a really difficult task of any molecular dynamics simulation. This is usually simplified by building an **AMBER** like force field [49], which estimates the interaction by calculating (in a largely empirical way) the force acting on every particle

ISSN: 2279–8773

of a given chemical system. In fact, the force field is worked out by taking into account informations coming from both experiments and theoretical calculations. In particular, thanks to the recent advances in computational technologies, the calculation of a force field is worked out by considering the contributions to the force on a single particle i due to all its neighbours (not only from two body but also sometimes from three and four body contributions). It is obvious that the particles which can be considered as neighbours to particle i are those confined into a given space around the considered particle.

An important feature of the force field approaches is the fact that the total energy of a molecule is formulated as the sum of two terms representing the bond (E_{bond}) interactions and the non-bonded ($E_{non-bonded}$) interactions respectively

$$V(\mathbf{r}_1, \ldots, \mathbf{r}_n) = E_{bond} + E_{non-bonded} \tag{1.116}$$

with $(\mathbf{r}_1, \ldots, \mathbf{r}_n)$ representing the set of position vectors of the N-atom system. Both terms of Eq. 1.116 can be further partioned into a sum of different contributions, as follows:

$$E_{bond} = E_l + E_\Phi + E_\omega \tag{1.117}$$

$$E_{non-bonded} = E_{vdw} + E_{coulomb} \tag{1.118}$$

The first term of the right hand side of Eq. 1.117 corresponds to the energy of the single bond, often expressed as a harmonic (or also a Morse) potential:

$$E_l = \sum_{ij=bond} V(r_{ij}) = \sum_{ij=bond} K_{r,ij}(r_{ij} - r_{eq,ij})^2 \tag{1.119}$$

In this equation $K_{r,ij}$ is the force constant of bond ij, r_{ij} is the covalent bond distance between atom i and atom j and $r_{eq,ij}$ is the related equilibrium distance. The second term on the right hand side of Eq. 1.117, E_Φ , represents the energy associated with the variation of the planar angle formed by two bonds, usually having an harmonic representation:

$$E_\Phi = \sum_{ijkl=angle} V(\Phi_{ijkl0}) = \sum_{ijkl=angle} K_{\Phi,ijkl}(\Phi_{ijkl} - \Phi_{eq,ijkl})^2 \tag{1.120}$$

where $K_{\Phi,ijkl}$ is the force constant associated to the bending of the Φ_{ijkl} angle formed as already pointed out before by the ij and jk bonds, while

ISSN: 2279-8773

$\Phi_{eq,ijkl}$ is the related equilibrium value. The last term on the right hand side of the same equation is associated with the variation of the dihedral angle formed by the two planes determined by two consecutive pair of bonds:

$$E_\omega = \sum_{i=dihedral-angle} A_i[1 + \cos(n_i\tau_i - \phi_i)] \tag{1.121}$$

Eq. 1.118 deals with the non-bonded interaction and is made of van der Waals interactions usually formulated as a 12-6 Lennard-Jones potential:

$$E_{vdw} = \sum_{ij=non-bonded} 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] \tag{1.122}$$

The second term in Eq. 1.118 is related to the Coulomb interaction:

$$E_{coulomb} = \sum_{ij} \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \tag{1.123}$$

where q_i and q_j are the atomic charges and 0 is the permittivity in the medium. It is important to point out here that the calculation of the non-bonded interactions goes with the square of N (the number of total particles in the system) and that *cutoff radius* is adopted.

The various types of force fields can be categorized as follows:

1. First class

   Amber was initially developed for calculating the properties of the biomolecules. The bond interactions are expressed using harmonic potentials. A non-bonded term, of Lennard-Jones 12-10 type, is also included in order to describe the hydrogen bonds [50, 51]:

   $$E_{H-bond} = 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{10}\right] \tag{1.124}$$

   In the field of biological sciences **Amber** is the most popular force field, due to the fact that such a force field has been developed together with the homonymous molecular dynamics software.

   MM2 originated from **MM1** [53] that has been the first force field developed in the group of Allinger to describe the modeling of hydrocarbons. Like **Amber**, **MM2** uses harmonic potentials to model the covalent bond interactions with some additional terms for the calculation of the torsional energy, in order to reproduce the differences of conformational energy between the trans and cis configurations of some organic molecules.

ISSN: 2279-8773

2. Second class

Dreiding [54] has force constants and geometrical parameters depending on the hibridization state of constituent atoms. This force field is more generic than those described above but reduces the number of the terms needed to work out the total energy system. The dreiding force field provides two terms for describing the bond stretching: a harmonic and Morse potential.

CFF (Consistent Force Field) [55] this is a recent proposal that is more accurate than some first-class force fields. The novelty is given by the addition of a cross energy term (See Eq. 1.116):

$$E_{crossed} = E_{l-l} + E_{\theta-\theta} + E_{l-\theta} + E_{\theta-\theta-\omega} \tag{1.125}$$

together with some additive terms to the part of the force field describing the bond interactions, consisting of terms of higher degree terms with respect to those given in Eqs. 1.119, 1.120 and 1.123. Terms given in Eq. 1.19 have the following form:

$$E_{l-l} = \sum_{r.ij}\sum_{r,kl}[F_{r,ij,kl}(r_{ij} - r_{eq,ij})(r_{kl} - r_{eq,kl})] \tag{1.126}$$

$$E_{\theta-\theta} = \sum_{\theta,ij}\sum_{\theta,kl}[F_{\theta,ij,kl}(\theta_{ij} - \theta_{eq,ij})(\theta kl - \theta_{eqkl})] \tag{1.127}$$

$$E_{l-\theta} = \sum_{r,ij}\sum_{\theta,ij}[F_{r,\theta,ij}(\theta_{ij} - \theta_{eq,ij})(r_{ij} - r_{eq,ij})] \tag{1.128}$$

where the last term of Eq. 1.125 is related to two planar angles and a dihedral one:

$$E_{\theta-\theta-\omega} = \sum_{\theta,ij}\sum_{\theta,kl}\sum_{\omega,ijkl}[K_{\theta_{ij},\theta_{kl},\omega_{ijkl}}(\cos\omega_{ijkl})(\theta_{ij}-\theta_{eq,ij})(\theta_{kl}-\theta_{eq,kl}) \tag{1.129}$$

The introduction of crossed terms in the calculation of the force field has shown to be very useful in order to work out the total energy of the system.

AMPF A different solution to the need of introducing cross energy terms is the one proposed in Ref. [56] in which many body components of the Van der Waals non-bonded contributions to the

ISSN: 2279–8773

interaction are formulated in a pseudo two body fashion using a Lennard-Jones whose parameters depend on the atom-diatom distances and orientations so as to reproduce polarizability effects.

1.5 The Statistical treatment of many particle systems

MD treatments are well suited for computer simulations of many-particle systems and the calculation of their properties. However, not all the observable properties can be directly evaluated in a simulation. Conversely, not all the quantities which are calculated during a simulation have a corresponding observable. To give a specific example: in a Molecular Dynamics simulation of liquid water, we could measure the instantaneous positions and velocities of all molecules in the liquid. However, this kind of information cannot be compared with experimental data, because no experiment can provide us with such detailed information. Whereas, typical experimental measurements are those referring to properties averaged over a large number of particles and, moreover often, also averaged over a typical time duration of the measurement.

The microscopic state of a model system is uniquely determined by the specification of the complete set of microscopic variables of its particles. The number of such variables is of the same order of the number of the particles of the system. On the contrary, the macroscopic state is specified by a small number of measurable quantities (such as the net mass, energy, or volume). In general, a large number of different microstates are compatible with a given macrostate. It is a primary task of statistical mechanics to find out how many microstates are needed for a given set of macroscopic conditions. Interpreting the microscopic variables as coordinates in a high-dimensional space we may represent a particular microstate as a point or a vector in that space. This space is called Gibbs phase space and the state vector is often represented by the symbol Γ. Consider a simple, classical many-particle system - say, an ideal gas, or a fluid made of hard spheres or Lennard-Jones molecules. The state vector is then defined by all position and velocity coordinates:

$$\mathbf{\Gamma} \equiv \{\mathbf{r}_1, \ldots, \mathbf{r}_N; \mathbf{v}_1, \ldots, \mathbf{v}_N\},\ \mathbf{r}_i \in V; v_{i,\alpha}(\pm\infty) \tag{1.130}$$

The number of degrees of freedom (d.o.f.) of a system of N particles is 6N in 3 dimensions (or 4N in 2 dimensions). The number of velocity d.o.f. is $3N$ and $2N$ (in 3 and 2 dimensions respectively). It is often allowed - and always advantageous - to treat the sub- spaces $\mathbf{\Gamma}_r \equiv \{\mathbf{r}_i\}$ (position

space) and $\mathbf{\Gamma}_v \equiv \{\mathbf{v}_i\}$ (velocity space) separately. The representation of a microstate of the entire system by a single point in $6N$-dimensional Gibbs space must not be confused with the $\hat{\mathrm{I}}\frac{1}{4}$-space introduced by Boltzmann. The μ-space has only 6 dimensions $\{\mathbf{r}, \mathbf{v}\}$, and each particle in the system has its own representative point. Thus the microstate of a system of N particles corresponds to a swarm of N points $\{\mathbf{r}_i, \mathbf{v}_i\}$. In the case of an ideal quantum gas it suffices to specify all quantum numbers to define a state:

$$\mathbf{\Gamma} \equiv \{(n_{ix}, n_{iy}, n_{iz});\ i = 1, \ldots N\} \tag{1.131}$$

Model systems made up of N spins are specified by

$$\mathbf{\Gamma} \equiv \{\sigma_i; i = 1, \ldots N\} \tag{1.132}$$

with $\sigma_i = \pm 1$.

1.5.1 The Statistical ensembles

Let us consider a collection of M systems, all having the same energy E, the same number of particles N, and - in the case of a gas or fluid - the same volume V. All microstates $\mathbf{\Gamma}$ compatible with these macroscopic conditions are then assumed also to be equally probable, that is, they have the same relative frequency within the ensemble of systems. The size M of the ensemble is assumed to be so large to be considered infinite. The assumption that all microstates which are compatible with the condition $E = E_0$ have the same probability is one of the solid foundations Statistical Mechanics is built upon. It is called the "postulate of equal a priori probability". For a mathematically exact formulation of this axiom we use the phase space density which is supposed to have the property

$$\rho(\mathbf{r}, \mathbf{v}) = \begin{cases} \rho_0 & \text{for } E(\mathbf{r}, \mathbf{v}) = E_0 \\ 0 & \text{elsewhere} \end{cases} \tag{1.133}$$

The condition $E = const$ defines an $(n-1)$-dimensional "surface" within the n-dimensional phase space of the system. The set of all points upon that "energy surface" is named microcanonical ensemble. To take an example, in the $3N$-dimensional velocity space of a classical ideal gas the equation

$$E_{kin} \equiv \frac{m}{2} \sum_i v_i^2 = const = E_0 \tag{1.134}$$

defines the $(3N-1)$-dimensional surface of a $3N$-sphere of radius $r = \sqrt{2E_0/m}$.

ISSN: 2279–8773

The integration of Newton's equations of motion allows the exploration the constant-energy surface of a system. However, most natural phenomena occur under conditions where a system is exposed to external pressure and/or exchanges heat with the environment. Under these conditions, the total energy of the system is no longer conserved, and extended forms of molecular dynamics are required. Several methods are available for controlling temperature and pressure. Depending on which state variables (the energy E, enthalpy H (i.e., $E + PV$), number of particles N, pressure P, stress S, temperature T, and volume V) are kept fixed, different statistical ensembles can be generated. A variety of structural, energetic, and dynamic properties can then be calculated from the averages or the fluctuations of these quantities over the ensemble generated. Both isothermal (exchange heat with a temperature bath to maintain a constant thermodynamic [not kinetic] temperature) and adiabatic (do not exchange heat) ensembles are available:

Constant number of particles, volume and energy	(NVE)
Constant number of particles, volume and temperature	(NVT)
Constant number of particles, pressure and temperature	(NPT)

NVE ensemble The constant-energy, constant-volume ensemble (NVE), also known as the *microcanonical ensemble*, is obtained by solving the standard Newton equations without any temperature and pressure control and it may be considered as the natural ensemble for molecular dynamics simulations. Energy is conserved when this (adiabatic) ensemble is generated. However, because of rounding and truncation errors during the integration process, there is always a slight fluctuation or drift in energy. Although the temperature is not controlled during true NVE dynamics, one might want to use NVE conditions during the equilibration phase of the simulation. For this purpose, all the MD softwares allow us to hold the temperature within specified tolerances by periodic scaling of the velocities. If no time dependent external forces are considered, the system's Hamiltonian is constant, implying that the system's dynamics evolves on a constant energy surface. The corresponding probability density in phase space is therefore given by

$$\rho(\mathbf{q}, \mathbf{p}) = \delta(\mathcal{H}(\mathbf{q}, \mathbf{p})E) \tag{1.135}$$

In a computer simulation this theoretical condition is generally violated, always due to limited accuracy and to roundoff errors. True constant-energy conditions (i.e., without temperature contol) are not recommended for equilibration because, without the energy flow facilitated by temperature control, the desired temperature cannot be achieved. However, during the data col-

ISSN: 2279–8773

lection phase, if we are interested in exploring the constant-energy surface of the conformational space, or for other reasons do not want the perturbation introduced by temperature- and pressure-bath coupling, this is a useful ensemble.

NVT ensembles The constant-temperature, constant-volume ensemble (NVT), also referred to as the *canonical ensemble*, is obtained by controlling the thermodynamic temperature.

- The differential thermostat. Direct temperature scaling [57] (also called Gaussian thermostat) should be used only during the initialization stage, since it does not produce a true canonical ensemble (it is not truly isothermal), because the velocities are scaled according to $\mathbf{p}_i \rightarrow \sqrt{T_0/T}\mathbf{p}_i$, where T_0 is the reference temperature and T the actual temperature, calculated from the velocity of the particles. This method leads to discontinuities in the momentum part of the phase space trajectory due to the rescaling procedure. It can be shown for this method that the configurational part of the phase space density is of canonical form, i.e.

$$\rho(\mathbf{q},\mathbf{p}) = \delta(T - T_0)\ e^{-\beta U(\mathbf{q})} \tag{1.136}$$

- The proportional thermostat. Any of the other temperature-control methods available is used during the data collection phase (production run); like for instance the Berendsen thermostat [58], that differs from the Gaussian thermostat because it allows for the temperature to fluctuate, thereby not fixing it to a constant value. In each integration step it is insured that the T is corrected to a value close to T_0 . By means of a weak coupling to an external bath the momenta undergo a modification: $\mathbf{p}_i \rightarrow \lambda\mathbf{p}_i$, where

$$\lambda = \left[1 + \frac{\delta t}{\tau_T}\left(\frac{T_0}{T} - 1\right)\right]^{\frac{1}{2}} \tag{1.137}$$

  The proportional thermostat preserves a Maxwell distribution and the phase space distribution is

$$\rho(\mathbf{q},\mathbf{q}) = f(\mathbf{p})\ e^{-\beta(U(\mathbf{q}) - \alpha\beta\delta(\mathbf{q})^2\ 3N)} \tag{1.138}$$

  where $\alpha \simeq (1 - \delta E/\delta U)$ and δU, δE are the mean fluctuations of the potential and total energy. $f(\mathbf{p})$ is, in general, an unknown function of the momenta, so that the full density cannot be determined. For

ISSN: 2279–8773

$\alpha = 0$, which corresponds in Eq. 1.137 to $\tau_T = \delta t$, the fluctuations in the kinetic energy vanish and Eq. 1.138 reduces to Eq. 1.136, i.e. it represents the canonical distribution. The other extreme ($\tau_T \to \infty$) corresponds to an isolated system and the energy should be conserved, i.e. $\delta E = \delta K + \delta U = 0$ and $\alpha = 1$. In this case, Eq. 1.138 corresponds to the microcanonical distribution. Eq. 1.138 may therefore be understood as an interpolation between the canonical and the microcanonical ensemble.

- The integral thermostat. This method has been devised by Nosé [59] and it is founded on the wish to confine the effect of an external system acting as heat reservoir (to keep the temperature of the system constant) to just one additional degree of freedom into the system's Hamiltonian (for which equations of motion can be derived). Nosé divided the variables into two sets: real and so called virtual ones. The transformation from virtual to real variables is then performed using the relationship:

$$\mathbf{p}_i = \phi_i s \quad \text{and} \quad \mathbf{q}_i = \rho_i \tag{1.139}$$

The resulting Hamiltonian, when expressed in virtual coordinates reads

$$\mathcal{H}^* = \sum_{i=1}^{N} \frac{\pi_i^2}{2m_i s^2} + U(\rho) + \frac{\pi_s^2}{2M_s} + g k_B T \ln s \tag{1.140}$$

where $g = 3N + 1$ is the number of degrees of freedom (system of N free particles). The Hamiltonian in Eq. 1.140 was shown to lead to a probability density in phase space, corresponding to the canonical ensemble.

NPT ensembles The constant-temperature, constant-pressure ensemble (NPT) allows the control over both the temperature and pressure. The unit cell vectors are allowed to change, and the pressure is adjusted by varying the volume (i.e., the size and also, in some programs, the shape of the unit cell). This method applies only to periodic systems. The constant-temperature, constant-stress ensemble (NST) is an extension of the constant-pressure ensemble. In addition to the hydrostatic pressure which is applied isotropically, the constant-stress ensemble allows the control of the xx, yy, zz, xy, yz, and zx components of the stress tensor. Pressure can be controlled by the Hoover and Berendsen (and Parrinello-Rahman) method. Temperature can be controlled by any method available (except, of course, the temperature scaling

method, since it is not truly isothermal). NPT is the ensemble of choice when the correct pressure, volume, and densities are important in the simulation. This ensemble can also be used during equilibration to achieve the desired temperature and pressure before changing to the constant-volume or constant-energy ensemble when data collection starts. The NST ensemble is particularly useful if you want to run a simulation at incremented tensile loads to study the stress-strain relationship in polymeric or metallic materials.

- The proportional barostat. The proportional thermostat was introduced as an extension for the equation of the momentum, since it couples to the kinetics of the particles. Since the barostat acts on a volume change, which may be expressed in a scaling of particles' positions, a phenomenological extension for the equation of motion of the coordinates may be formulated. Accordingly one has

$$\frac{\partial \mathbf{q}_i}{\partial t}\frac{\mathbf{p}_i}{m_i} + \alpha \mathbf{q}_i, \tag{1.141}$$

while for a change in volume one has

$$\dot{V} = 3\alpha V \tag{1.142}$$

A change in pressure is related to the isothermal compressibility κ_T

$$\dot{P} = \frac{1}{\kappa_T V}\frac{\partial V}{\partial t} = -\frac{3\alpha}{\kappa_T} \tag{1.143}$$

If we call L the length of the box, the scaling of both the coordinates and the boxlength is $\mathbf{q} \rightarrow s\mathbf{q}$ and $L \rightarrow sL$, where

$$s = 1 - \frac{\kappa_T \delta t}{e\tau_P}(P_0 - P) \tag{1.144}$$

The time constant τ_P was introduced as a characteristic timescale on which the system pressure will approach the desired pressure P_0. It also controls the strength of the coupling to the barostat and therefore the strength of the volume/pressure fluctuations. A drawback of the proportional thermostat is the fact that the statistical ensemble is unknown. As in the case of the thermostat, it may be assumed to interpolate between the microcanonical and the constant-pressure/constant-enthalpy ensemble, depending on the coupling constant τ_P.

ISSN: 2279–8773

- The integral barostat. In line with the integral thermostat one can introduce a new degree freedom into the systems Hamiltonian which controls volume fluctuations. This method was developed by Hoover and modified by Melchionna [60]. The idea is to include the volume as an additional degree of freedom and to write the Hamiltonian in a scaled form, where lengths are expressed in units of the boxlength $L = V^{1/3}$, i.e. $\mathbf{q}_i = L\rho_i$ and $\mathbf{p}_i = L\pi_i$. Since L is also a dynamical quantity, the momentum is not related to the simple time derivative of the coordinates but $\partial t\mathbf{q}_i = L\partial_t\rho_i + \rho_i\partial_t L$. The extended system Hamiltonian is then written as

$$\mathcal{H}^* = \sum_{i=1}^{N} \frac{\pi_i^2}{2m_i s^2} + U(\rho) + \frac{\pi_s^2}{2M_s} + gk_BT\ln s \tag{1.145}$$

where P_{ex} is the prescribed external pressure and π_V and M_V are a momentum and a mass associated with the fluctuations of the volume. From that Hamiltonian we can derive the equations of motion and, by means of a transformation to real variables, we have

$$\frac{\partial \mathbf{p}_V}{\partial t} = \frac{1}{3V}\left(\sum_{i=0}^{N}\frac{\mathbf{p}_i}{m_i} - \mathbf{q}_i\frac{\partial U(\mathbf{q})}{\partial \mathbf{q}}\right) - P_{ex} \tag{1.146}$$

In the last equation the term in brackets corresponds to the pressure calculated from the virial theorem. The associated volume force, introducing fluctuations of the box volume is therefore controlled by the internal pressure, originating from the particle dynamics and the external pressure, P_{ex}.

1.5.2 The evaluation of observable properties

The following properties are considered desirable for a classical mechanical ensemble especially with respect to the evaluation of observable properties.

1. representativeness: The chosen probability measure on the phase space should be a Gibbs state of the ensemble, i.e. it should be invariant under time evolution. A standard example of this is the natural measure on a constant energy surface for a classical mechanical system. Liouville's theorem states this measure is invariant under the Hamiltonian flow.

ISSN: 2279–8773

2. ergodicity: Once a probability measure μ on the phase space Λ is specified, one can define the ensemble average of an observable, i.e. real-valued function f defined on Λ via this measure by

$$\langle f \rangle = \int_\Lambda f d\mu \tag{1.147}$$

where we have restricted to those observables which are μ-integrable. On the other hand, let $x(0)$ denote a representative point in the phase space, and $x(t)$ be its image under the flow, specified by the system in question, at time t. The time average of f is defined to be

$$\lim_{T\to\infty} \int_0^T f(x(t))dt, \tag{1.148}$$

provided that this limit exists $\hat{\mathrm{I}}\frac{1}{4}$-almost everywhere and is independent of $x(0)$.

The ergodicity requirement is that the ensemble average coincides with the time average. A sufficient condition for ergodicity is that the time evolution of the system is a mixing (see below for the ergodic hypothesis.) Not all systems are ergodic. For instance, it is unknown at this time whether classical mechanical flows on a constant energy surface are ergodic in general. Physically, when a system fails to be ergodic, we may infer that there is more macroscopically discoverable information available about the microscopic state of the system than what we first thought. In turn this may be used to create a better-conditioned ensemble.

Ergodic hypothesis In physics and thermodynamics, the ergodic hypothesis says that, over long periods of time, the time spent in some region of the phase space of microstates with the same energy is proportional to the volume of this region, i.e., that all accessible microstates are equally probable over a long period of time. Equivalently, it says that time average and average over the statistical ensemble are the same. Liouville's Theorem shows that, for conserved classical systems, the local density of microstates following a particle path through phase space is constant as viewed by an observer moving with the ensemble (i.e., the total or convective time derivative is zero). Thus, if the microstates are uniformly distributed in phase space initially, they will remain so at all times. Liouville's theorem ensures that the notion of time average makes sense, but ergodicity does not follow from Liouville's theorem. In macroscopic systems, the timescales over which a system can

ISSN: 2279–8773

truly explore the entirety of its own phase space can be sufficiently large that the thermodynamic equilibrium state exhibits some form of ergodicity breaking.

Measuring the Statistical Quantities

Measuring quantities in molecular dynamics usually means performing time averages of physical properties over the system trajectory. Physical properties are usually a function of the particle coordinates and velocities. So, for instance, one can define the in- stantaneous value of a generic physical property A at time t:

$$A(t) = f(\mathbf{r}_1(t), \ldots, \mathbf{r}_N(t), \mathbf{v}_1(t), \ldots, \mathbf{v}_N(t)) \tag{1.149}$$

and then obtain its average

$$\langle A \rangle = \frac{1}{N_T} \sum_{t=1}^{N_T} A(t) \tag{1.150}$$

where t is an index which runs over the time steps from 1 to the total number of steps N_T. There are two equivalent ways to do this in practice: $A(t)$ is calculated at each time step by the MD program while running. The sum $\sum_t A(t)$ is also updated at each step. At the end of the run the average is immediately obtained by dividing by the number of steps. This is the preferred method when the quantity is simple to compute and/or particularly important. An example is the system temperature. Positions (and possibly velocities) are periodically dumped in a "trajectory file" while the program is running. A separate program, running after the simulation program, processes the trajectory and computes the desired quantities. This approach can be very demanding in terms of disk space: dumping positions and velocities using 64-bit precision takes 48 bytes per step and per particle. However, it is often used when the quantities to compute are complex, or combine different times as in dynamical correlations, or when they are dependent on other additional parameters that may be unknown when the MD program is run. In these cases, the simulation is run once, but the resulting trajectory can be processed over and over. The most commonly measured physical properties are discussed in the following list:

- Potential energy. The average potential energy V is obtained by averaging its instantaneous value, which is usually obtained straightforwardly at the same time as the force computation is made. For instance, in the case of two-body interactions

ISSN: 2279–8773

$$V(t) = \sum_i \sum_{ij} \phi(\mid \mathbf{r}_i(t) - \mathbf{r}_j(t) \mid) \tag{1.151}$$

Even if not strictly necessary to perform the time integration (forces are all is needed), knowledge of V is required to verify energy conservation. This is an important check to do in any MD simulation.

- Kinetic energy. The instantaneous kinetic energy is of course given by

$$K(t) = \frac{1}{2} \sum_i m_i [v_i(t)]^2 \tag{1.152}$$

and is therefore extremely easy to compute. We will call K its average on the run.

- Total energy. The total energy $E = K + V$ is a conserved quantity in Newtonian dynamics. However, it is common practice to compute it at each time step in order to check that it is indeed constant with time. In other words, during the run energy flows back and forth between kinetic and potential, causing $K(t)$ and $V(t)$ to fluctuate while their sum should remain fixed (in practice there could be small fluctuations in the total energy). These fluctuations are usually caused by errors in the time integration, and can be made smaller by reducing the time step if considered too large. Slow drifts of the total energy are also sometimes observed in very long runs. Such drifts could also be originated by a too large time step. Drifts are more disturbing than fluctuations because the thermodynamic state of the system is also changing together with energy, and therefore time averages over the run do not refer to the same state. If drifts over long runs tend to occur, they can be prevented, for instance by breaking the long run into smaller pieces and restoring the energy to the nominal value between the various pieces. A common mechanism to adjust the energy is to modify the kinetic energy via a rescaling of the velocities. A final word of caution: while one may be tempted to achieve "perfect" energy conservation by reducing the time step as much as desired, working with an excessively small time step may result in waste of computer time. A practical compromise would probably allow for small energy fluctuations and perhaps slow energy drifts, as a price to pay to work with a reasonably large time step.

- Temperature. The temperature T is directly related to the kinetic energy by the well-known equipartition formula, assigning an average kinetic energy $k_B T/2$ per degree of freedom:

ISSN: 2279–8773

$$K = \frac{3}{2} N k_B T \tag{1.153}$$

An estimate of the temperature is therefore directly obtained from the average kinetic energy K. For practical purposes, it is also common practice to define an instantaneous temperature $T(t)$, proportional to the instantaneous kinetic energy $K(t)$ by a relation analogous to the one above.

- Mean square displacement (MSD). The MSD of atoms in a simulation can be easily computed by its definition

$$MSD = \mid r(t) - r(0) \mid^2 \tag{1.154}$$

where $\langle \dots \rangle$ denotes here averaging over all the atoms (or all the atoms in a given subclass). The MSD contains information on the atomic diffusivity. If the system is solid, MSD saturates to a finite value, while if the system is liquid, MSD grows linearly with time. In this case it is useful to characterize the system behavior in terms of the slope, which is the diffusion coefficient D:

$$D = \lim_{t \to \infty} \frac{1}{6t} \langle \mid \mathbf{r}(t) - \mathbf{r}(0) \mid^2 \rangle \tag{1.155}$$

The 6 in the above formula must be replaced by 4 in two dimensional systems. Some cautions, however, should be used in using eq. 1.155. As an example in the case of the penetration of a membrane a substance should go through three stages: 1) absorption into the membrane 2) diffusion through the membrane and 3) desorption of the penetrant out from the opposite surface of the membrane. It is well known that the slowest, and therefore the rate determining, stage is the diffusion which is well described by the diffusion coefficient D. It is important to emphasize here that equation 1.155 is valid only when the motion of the diffusing particle follows a random walk i.e. its motion is not correlated with that of any previous time. This means that the Einstein diffusion regime has been reached. If the surroundings inhibit the free motion of the particle, (for instance it remains trapped for a while into a small space limited by the layers of a membrane), the diffusion is called anomalous diffusion. In this case $\langle \mid R_i(t) - R_i(0) \mid^2 \rangle \propto t^n$, where $n < 1$, and equation 1.155 is not valid. When $\langle \mid R_i(t) - R_i(0) \mid^2 \rangle \propto t^n$, where $n > 1$, the motion of the particle is not diffusive and other

transport mechanism is effective. It is possible to test the region in which equation 1.155 is valid by plotting log(MSD) against log(t) and in the case of Einstein diffusion the slope of the curve is one:

$$\frac{\Delta \log(MSD)}{\Delta \log(t)} = 1 \tag{1.156}$$

Using the Einstein equation we can determine the ionic conductivity σ,

$$\sigma = \frac{e^2}{6tVk_BT}\Big(\sum_i z_i^2 \langle \mid R_i(t) - R_i(0) \mid^2 \rangle + 2\sum_{j>i} z_i z_j \langle \mid R_i(t) - R_i(0) \mid\mid R_j(t) - R_j(0) \mid \rangle\Big) \tag{1.157}$$

where t is time, V is the volume of the cell, k_B is Boltzmann's constant, T is the temperature and R is the position vector of the diffusing ion. The first term on the right hand side is the sum over individual MSD weighted with the charges and the second is the sum of correlation of displacements of ions describing the interactions between different ions.

- Radial distribution function (rdf). In statistical mechanics the rdf $g(\mathbf{r})$ is defined making use of the Dirac delta functions:

$$g(r) = \frac{V}{N^2}\langle \sum_{i>j}^{N} \delta(\mathbf{r}_{ij} - \mathbf{r}) \rangle \tag{1.158}$$

where V is the system volume, N is the number of particles, $\mathbf{r}_{ij} = \mid \mathbf{r}_i - \mathbf{r}_j \mid$ are particle coordinates and also appear as integration variables in the statistical average, the vector $\mathbf{r}$ is parameter with a value of our choice (thus appears as a true argument on the left hand side) and the angular brackets denote an integral over the configurational probability distribution function $P_N(\mathbf{r}^N)$, containing all configurational information there is to know. So, in explicit form:

$$g(\mathbf{r}) = \frac{V}{N^2}\sum_{i>j}^{N} \int d\mathbf{r}_1 d\mathbf{r}_2 \ldots d\mathbf{r}_N \; P_N(\mathbf{r}^N) \; \delta(\mathbf{r}_{ij} - \mathbf{r}) \tag{1.159}$$

ISSN: 2279-8773

$g(\mathbf{r})$ has a direct probabilistic interpretation: it is proportional to the probability for observing any two particles separated by a vector $\mathbf{r}$ (proportional but identical because of the difference in normalization factor); for instance liquids are isotropic systems with no preference for a direction in space. After some mathematical transformation one can obtain the following expression of rdf as a function of only $\mathbf{r}$:

$$g(r) \approx \frac{V}{4\pi r^2 \Delta r N^2} \langle \sum_i n_i(r, \Delta r) \rangle \tag{1.160}$$

which becomes exact in the limit $\Delta r \to 0$. Radial distribution functions are essentially histograms of two-particle distances and they are dimensionless quantities.

- Pressure. The measurement of the pressure in a molecular dynamics simulation is based on the Clausius virial function

$$W(\mathbf{r}_1, \ldots, \mathbf{r}_N) = \sum_{i=j} \mathbf{r}_i \dot{F}_i^{TOT}, \tag{1.161}$$

where F_i^{TOT} is the total force acting on atom i. Its statistical average $\langle W \rangle$ will be obtained, as usual, as an average over the molecular dynamics trajectory:

$$\langle W \rangle = \lim_{t \to \infty} \frac{1}{t} \int \cdots \sum_{i=0}^{N} \mathbf{r}_i(\tau) \dot{m}_i \ddot{\mathbf{r}}_i(\tau), \tag{1.162}$$

where use has been made of Newton's law. By integrating by parts,

$$\langle W \rangle = - \lim_{t \to \infty} \frac{1}{t} \int_0^t d\tau \sum_{i=0}^{N} m_i \mid \dot{\mathbf{r}}_i(\tau) \mid^2 . \tag{1.163}$$

This is twice the average kinetic energy, therefore by the equipartition law of statistical mechanics

$$\langle W \rangle = -DNk_BT, \tag{1.164}$$

where D is the dimensionality of the system (2 or 3), N the number of particles, k_B the Boltzmann constant. Now, one may think of the total force acting on a particle as composed of two contributions:

ISSN: 2279-8773

$$\mathbf{F}_i^{TOT} = \mathbf{F}_i + \mathbf{F}_i^{EXT}, \tag{1.165}$$

where $\mathbf{F}_i$ is the internal force (arising from the interatomic interactions), and $\mathbf{F}_i^{EXT}$ is the external force exerted by the container's walls. If the particles are enclosed in a parallelepiped container of dimensions L_x, L_y, L_z, volume $V = L_x L_y L_z$, and with the origin origin of the coordinates on one of its corners, the part $\langle W^{EXT} \rangle$ due to the container can be evaluated using the definition:

$$\langle W^{EXT} \rangle = L_x(-PL_yL_z) + L_y(-PL_xL_z) + L_z(-PL_xL_y) = -DPV, \tag{1.166}$$

where $-PL_yL_z$ is, for instance, the external force $\mathbf{F}_x^{EXT}$ applied by the yz wall along the x directions to particles located at $x = L_x$, etc. can then be written

$$\langle \sum_{i=1}^{N} \mathbf{r}_i \dot{\mathbf{F}}_i \rangle - DPV = -DNk_BT$$

$$PV = Nk_BT + \frac{1}{D} \langle \sum_{i=1}^{N} \mathbf{r}_i \dot{\mathbf{F}}_i \rangle. \tag{1.167}$$

This important result is known as the virial equation. All the quantities except the pressure P are easily accessible in a simulation.Note how the Eq. 1.167 reduces to the well-known equation of state of the perfect gas if the particles are non-interacting. In the case of pairwise interactions via a potential $\phi(r)$ the Eq. 1.167 becomes

$$PV = Nk_BT - \frac{1}{D} \langle \sum_i \sum_{ij} \mathbf{r}_{ij} \mid \frac{d\phi}{dr} \mid_{r_{ij}} \rangle. \tag{1.168}$$

This expression has the additional advantage over to be naturally suited to be used when periodic boundary conditions are present: it is sufficient to take them into account in the definition of r_{ij}.

ISSN: 2279-8773

1.6 Multiscale Modeling

Multiscale modeling and computation is a rapidly evolving area of research that will have a fundamental impact on computational science and applied mathematics and will influence the way we view the relationship between mathematics and science. Even though multiscale problems have been studied in mathematics since long the current excitement is driven mainly by the use of mathematical models in the applied sciences: like material science, chemistry, fluid dynamics, and biology. Problems in these areas are often multiphysics in nature. Namely, the processes occurring at different scales are governed by physical laws having a different character. For example, quantum mechanics at one scale and classical or fluid mechanics at another. Emerging from this intense activity is a need for new mathematics and new ways of interacting with mathematics. Fields such as mathematical physics and stochastic processes, which have so far remained in the background as far as modeling and computation of complex systems starting from first principles is concerned, will move to the front. New questions will arise, new priorities will be set as a result of the rapid evolution in the computational fields.

There are several reasons for the timing of the current interest. Modeling at the level of a single scale, such as molecular dynamics or continuum theory, is becoming relatively mature. Our computational capability has reached the stage when serious multiscale problems can be considered, and there is an urgent need from science and technology, nano-science being a good example, for multiscale modeling techniques.

An example with multiple time scales is that of protein folding. While the time scale for the vibration of the covalent bonds is on the order of femtoseconds (10^{-15} s), folding time for the proteins may very well be on the order of seconds. Well-known examples of problems with multiple length scales include turbulent flows, mass distribution in the universe, and vortical structures on the weather map [61]. At the same time different physical laws may be required to describe the system at different scales. Take the example of fluids. At the macroscale (meters or millimeters), fluids are accurately described by the density, velocity, and temperature fields, which obey the continuum Navier-Stokes equations. On the scale of the mean free path, it is necessary to use kinetic theory (Boltzmann's equation) to get a more detailed description in terms of the one-particle phase-space distribution function. At the nanometer scale, molecular dynamics in the form of Newton's law has to be used to give the actual position and velocity of each individual atom that makes up the fluid. If a liquid such as water is used as solvent for protein folding, then the electronic structures of the water molecules become

ISSN: 2279–8773

important, and these are described by Schrödinger's equation in quantum mechanics. The boundaries between different levels of theories may vary, depending on the system being studied, but the overall trend described above is generally valid. At each finer scale, a more detailed theory has to be used, giving rise to more detailed information on the system.

1.6.1 Approaches to multiscaling

There is a long history in mathematics for the study of multiscale problems. Fourier analysis has long been used as a way of representing functions according to their components at different scales. More recently, this multiscale, multiresolution representation has been made much more efficient through wavelets. On the computational side, several important classes of numerical methods have been developed which address explicitly the multiscale nature of the solutions. These include multigrid methods, domain decomposition methods, fast multipole methods, adaptive mesh refinement techniques, and multiresolution methods using wavelets. From a modern perspective, the computational techniques described above are aimed at efficient representation or solution of the fine-scale problem. For many practical problems, full representation or solution of the fine-scale problem is simply impossible for the foreseeable future because of the overwhelming computational costs. Therefore we must seek alternative approaches which are more efficient.

One classical approach is to use analytic techniques to derive effective models at the scale of interest. As an example of the application of such a technique let us consider the averaging method in which the multiscale nature of the problem can be formulated as a system of ordinary differential equations in the action-angle variables by writing it as follows

$$\varphi_t = \frac{1}{\varepsilon}\omega(I) + f(\varphi, I)I_t = g(\varphi, I)$$

where φ is the fast variable, which varies on the time scale of $O(\varepsilon)$, while $\varepsilon \ll 1$; I is the slow variable, which mainly varies on the time scale of $O(1)$ (f and g are assumed to be 2π-periodic in φ).

Another example of a mathematical technique for approaching multiscale problems is the homogenization method for which we can consider the problem

$$\frac{\partial u^\varepsilon}{\partial t} = \nabla \cdot (a(x, \frac{x}{\varepsilon})\nabla u^\varepsilon(x,t)), \quad x = \Omega \tag{1.169}$$

with the boundary condition $u^\varepsilon\|_{\partial\Omega} = 0$. In this problem the multiscale nature comes from the coefficients $a(x, \frac{x}{\varepsilon})$, which contain two scales: a scale

ISSN: 2279–8773

of $O(\varepsilon)$ and a scale of $O(1)$. Not only is (1.169) a nice model problem for the homogenization technique, it also describes important physical processes such as heat conduction in a composite material (for simplicity let us assume that $a(x, y)$ is periodic in y). It can be shown [62] that for $\varepsilon \ll 1$, $u^\varepsilon(x, t)$ can be expressed in the form

$$u^\varepsilon(x,t) = U(x,t) + \varepsilon u_1(x, \frac{x}{\varepsilon}, t) + \varepsilon^2 u_2((x, \frac{x}{\varepsilon}, t)) + \ldots, \tag{1.170}$$

where U satisfies a homogenized equation

$$\frac{\partial U}{\partial t} = \nabla \cdot (A(x)\nabla U(x,t)). \tag{1.171}$$

Here $A(x)$ may be thought of as being the effective coefficient describing the effective properties of the system on the scale of $O(1)$. Determining $A(x)$ usually requires solving families of so-called cell problems. In the onedimensional case, however, $A(x)$ is simply given by the harmonic average

$$A(x) = (\int_0^1 \frac{1}{a(x,y)} dy)^{-1}. \tag{1.172}$$

Many other mathematical approaches have been developed to study multiscale problems, including boundary-layer analysis [63], semiclassical methods [64], geometric theory of diffractions [65], stochastic mode elimination [66], and renormalization group methods [67], [68]. Despite this progress, purely analytical techniques are still very limited when it comes to problems of practical interest. As a result, the overwhelming majority of problems have been approached using empirical techniques to model the small scales in terms of the macroscale variables using empirically derived formule. As a matter of fact, a large part of the progress in physical sciences lies in such empirical modeling. A familiar example is the case of the continuum theory of fluid dynamics. To derive the system of equations for fluids, we apply Newton's law to an arbitrary volume of fluid denoted by Ω:

$$\frac{D}{Dt} \int_\Omega \rho u dV = F(\Omega) \tag{1.173}$$

where $\frac{D}{Dt}$ is the material derivative, ρ and u are the density and velocity fields respectively, and $F(\Omega)$ is the total force acting on the volume of fluid in Ω. The forces consist of body forces such as gravity, which we neglect for the present argument, due to the long-range interaction of the molecules that make up the fluid, and forces due to the macroscopic-range interaction

ISSN: 2279–8773

between the molecules, such as the Van der Waals interaction. In the continuum theory, the short-range forces are represented as a surface integral of the stress tensor τ, which is a macroscopic idealization of the small scale effects,

$$F(\Omega) = \int_{\partial\Omega} (\tau\hat{n})ds \tag{1.174}$$

where $\hat{n}$ is the unit outward normal of Ω. The stress τ can be expressed as $\tau = -pI + \tau_d$, where p is the pressure, I is the identity tensor, and τ_d is the dissipative part of the stress. In order to close the system, we need to express τ_d in terms of u. In the simplest empirical approximation, τ_d is assumed to be a linear function of ∇u. This leads to

$$\tau_d = \mu \frac{\nabla u + (\nabla u)^T}{2} \tag{1.175}$$

where μ is called the viscosity of the fluid. Substituting this into Newton's law and adding the incompressibility condition gives rise to the wellknown Navier-Stokes equation:

$$\rho(u_t + (u \cdot \nabla)u) + \nabla p = \mu\Delta u \qquad \nabla \cdot u = 0 \tag{1.176}$$

In such a macroscopic description, all molecular details of the liquid are lumped into a single parameter, the viscosity. Fluids for which Eq. 1.175 gives an accurate description of the small-scale effects are called Newtonian fluids. This simple derivation illustrates how, in general, continuum models in the form of partial differential equations are derived. One typically starts with some universal conservation laws such as Eq. 1.173. This requires introducing certain currents or flux densities, which are then expressed by some postulated constitutive relations such as 1.175. In this way, we obtain the heat equation for thermal conduction by postulating Fourier's law, the diffusion equation for mass transport using Fick's law, and the porous medium equation using Darcy's law. Such empirical ad hoc descriptions of the small scales are used almost everywhere in science and engineering. In molecular dynamics, empirical potentials are used to model the forces between atoms, mediated by the electrons. In kinetic theory, empirical collision kernels are used to describe probabilistically the short-range interaction between the atoms and the molecules. Other examples include plasticity, crack propagation, and chemical reactions. While much progress has been made using such empirical approaches, their shortcomings have also been recognized, especially so in recent years, since numerical simulations based on the empirical models are now accurate enough that the modeling error

ISSN: 2279–8773

can be clearly identified. Microscale simulation methods such as electronic structure calculations have matured, enabling us to ask more ambitious questions. Moreover, the empirical approach often lacks information about how microstructural changes, such as the conformation of polymers in a polymeric fluid, affect the macroscale properties of the system.

1.6.2 Examples of Multiscale Problems

In view of the limitations of the empirical approach, several "first principle"-based multiscale methods have been proposed in recent years. Some of these methods are discussed below.

Molecular Dynamics: This has been already exaustively treated in Section 1 of this Chapter.

The Quasicontinuum Method: In the continuum theory of nonlinear elasticity, we are often interested in finding the displacement field by solving a variational problem

$$\min_u E(u) = \int_\Omega f(\nabla u) dx \tag{1.177}$$

where E is the total elastic energy, u is the displacement field, and f is the stored energy functional, subject to certain loading or boundary conditions. This approach takes for granted that the function f is explicitly given. Actually the process of finding f is rather empirical and often even crude. A different methodology called the quasicontinuum method has been proposed in Refs. [69] and [70] for the analysis of crystalline materials. In this case the microscopic model comprises molecular mechanics of the atoms that make up the crystal. Given a macroscopic triangulation of the material, let V_H be the standard continuous piecewise-linear finite-element space over this triangulation. For $U \in V_H$, ∇U is constant on each element K. Let $E_K(U)$ be the energy of a unit cell in an infinite volume uniformly deformed according to the constant deformation gradient $\nabla U \mid_K$. In the quasicontinuum approximation, the total energy associated with the trial function U is then given by

$$\tilde{E}(U) \sum_K n_K E_K(U) \tag{1.178}$$

where n_K is the number of unit cells in the element K. This approach bypasses the necessity of modeling f empirically. Instead, the effective f is computed on the fly using microscopic models. What we have described is the simplest version of the quasicontinuum method. There are many improvements, in particular to deal with defects in the crystal [70].

Kinetic-Hydrodynamic Models of Complex Fluids: Consider, for example, polymers in a solvent. The basic equations follow again from that of mass and momentum conservation:

$$\rho(u_t + (u \cdot \nabla)u) + \nabla p = \mu_s \Delta u + \nabla \cdot \tau_p \quad \nabla \cdot u = 0 \tag{1.179}$$

Here we have decomposed the total stress into two parts: one part, τ_p, due to the polymer and the other part due to the solvent, for which we used Newtonian approximation; μ_s is the solvent viscosity. Traditionally, τ_p is modeled empirically using constitutive relations. The most common models are a generalized Newtonian model and various viscoelastic models. It is generally acknowledged that it is an extremely difficult task to construct such empirical models in order to describe the flow under all experimental conditions. An alternative approach was proposed in the classical work of Kramers, Kuhn, Rouse et al. [71]. Instead of using empirical constitutive relationships, this approach makes use of a simplified kinetic description for the conformation of the polymers. In the simplest situation, the polymers are assumed to be dumbbells, each of which consists of two beads connected by a spring. Its conformation is therefore described by that of the spring. The dumbbells are convected and stretched by the fluid, and at the same time they experience spring and Brownian forces:

$$\gamma(Q_t + (u \cdot \nabla)Q - (\nabla u)^T Q) = F(Q) + \sqrt{k_B T \gamma}\dot{W}(t) \tag{1.180}$$

Here Q denotes the conformation of the dumbbell, $F(Q)$ is the spring force, γ is the friction coefficient, $\dot{W}(t)$ is temporal white noise, k_B is the Boltzmann constant, T is the temperature, and I is the identity tensor. If we have Q, we can compute the polymer stress τ_p via

$$\tau_p = nk_B T I + \mathbb{E}(F(Q) \bigotimes Q), \tag{1.181}$$

where n is the polymer density and $\mathbb{E}$ denotes expectation over the Brownian forces. These equations are valid in the dilute regime when

ISSN: 2279–8773

direct interaction between polymers can be neglected. The dumbbell model is a very simplified one and in many cases needs to be improved. This can be done in a number of ways (see Ref [71]).

Many other multiscale methods which are similar to those mentioned above have been developed in the past few years. We mention in particular the work of Abraham et al. on coupling finite element continuum analysis with molecular dynamics and tight binding [72], the work on coupling kinetic equations with hydrodynamic equations, Vanden-Eijnden's method for solving stochastic ordinary differential equations with multiple time scales [73], superparametrization techniques in meteorology in which the parameters for turbulent transport are determined dynamically by local microscale simulations, and the work of Kevrekidis et al. on bifurcation analysis based on microscopic models [74]. By explicitly taking advantage of the separation of scales, these methods become much more efficient than solving the full fine-scale problem. This is a common feature of the new class of multiscale methods we are interested in. In contrast, traditional multiscale techniques such as the original multigrid methods are rather blind to the special features of the problem, since they are aimed at solving the full fine-scale problem everywhere in the macroscale domain. Of course many practical problems such as turbulent flows do not have separation of scales. For these problems, other special features, such as selfsimilarity in scales, must be identified first before we have a way of modeling them more efficiently than simply solving the fine-scale problem by brute force or resorting to ad hoc models.

The most used traditional and modern computational multiscale techniques are given in Table 1.1.

ISSN: 2279–8773

Table 1.1: Traditional and modern computational multiscale techniques. Traditional multiscale techniques focus on resolving the fine-scale problem. Modern multiscale techniques try to reduce the computational complexity by using special features in the fine-scale problem, such as scale separation.

Traditional Techniques	Recent Techniques
Multigrid Method	Car-Parrinello Method
Domain Decomposition	Quasi-Continuum Method
Multiresolution Methods	Superparametrization
Adaptive Mesh Refinement	Heterogeneous Multiscale Method
Fast Multipole Method	Vanden-Eijnden's Method
Conjugate Gradient Method	Coarse-Grained Monte Carlo Models
	Adaptive Model Refinement
	Patch Dynamics

ISSN: 2279-8773

ISSN: 2279-8773

Chapter 2

Concurrent computing for molecular sciences

2.1 Introduction

As we have already mentioned the European Union is promoting the assemblage of large scale distributed computing platforms facilities to support the solution of the so called Grand Challenges in computational sciences. Accordingly, the evolution of computer technologies is at present going beyond the policy of individual large computer centers machines in favour of clusters of out of the shelves PCs. At the some time Computational applications become increasingly complex and need the converged effort of different expertise and multiscale modelling. This has led to the present effort as creating Grid platforms and fostering the development and implementation on the Grid of ICT applications in all fields of human activities including Molecular and material science and technology.

For applications concerning the lowest level usage of the Grid plarform (data transfer) a broad bandwidth, a safe and secure transfer protocol and authenticated access are needed. A distributed usage of the information on the Grid dramatically requires new services mainly consisting of intelligent tools for information representation and handling. Moreover, the shared and distributed usage of knowledge is one of the most complex tasks that can be performed on the Grid. In fact, it implies the coordination of the expertise, the cross fertilization of the know how, the protection of the intellectual property and the stimulation of teamed innovative research to be designed and implemented on a highly distributed and heterogeneous context. All three levels of usage are of paramount importance in natural sciences since they have to deal with the interpretation of transformations in matter. In this

ISSN: 2279–8773

contest, Computational Chemistry plays a key role because of its ability to carry out realistic *a priori* simulations of the multiscale type, starting from the microscopic level of the molecular interaction. The complexity of this type of multiscale simulations, ranging from molecular to human level (and necessarily founded on competences and skills scattered over geographically dispersed laboratories), requires the exploitation of large computing resources that only the Grid can supply.

In this chapter, the evolution of computer technologies and platforms towards distributed computing and its impact on the way Computational Chemistry research is carried out.

2.2 Concurrency on a single processor

The single-processor architectures are based on the well known Von Neumann model, sketched in Fig. 2.1. In this model the processing unit first fetches an instruction from the memory that subsequently is interpreted and executed after the fetching of the operands involved. Therefore, in a single-cycle the CPU performs the following operations:

- ***Fetch the next instruction***: the next instruction referenced by the program counter is transferred from memory to CPU;
- ***Decode***: the fetched instruction is interpreted and related circuitry activated;
- ***Fetch the operands***: the referenced operands are transferred from memory and stored in the registers;
- ***Execute***: the decoded instruction is executed;
- ***Check for interrupt***: the processor checks for signal of interrupt issued by the operating system or any other process and decides accordingly to stop or to solve the interrupt and continue;
- ***Store***: the results are transferred to the appropriate memory locations;
- ***Increment*** the counter and go to fetch to do a new iteration.

The Von Neumann architecture has represented the first step in the process of building efficient automatic handling of information. Year by year, the original architecture has been modified to increase its speed and improve its efficiency. The progress towards faster computers has been largely based, in

ISSN: 2279–8773

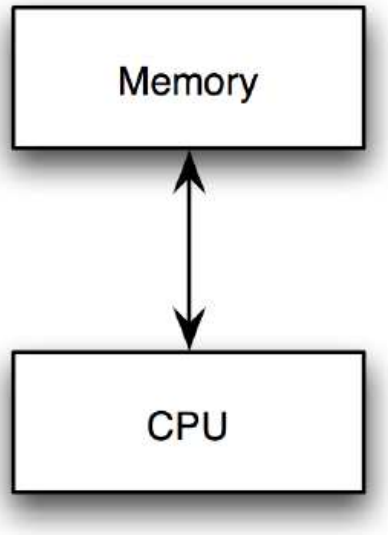

Figure 2.1: The Von Neumann model.

the past, on technology advances in producing increasingly more integrated circuitry. This has made the CPU more efficient, the memories more capable, the buses more transfer-performant.

The ability of further integrating an increasing number of circuitry elements on a chip has led to a silicon design integrating into the same processor socket multiple execution cores (multicore processors).

Despite the fact that the multi-core processor plugs multiple "execution cores" directly into a single processor socket, the operating system perceives each of them as a discrete logical processor with all the associated execution resources. The idea behind this implementation is the strategy of *divide et impera*. This means that by dividing the computational work traditionally performed by the single processor's core in traditional processors and spreading it over multiple execution cores, a multi-core processor can perform more work within a given clock cycle. To enable this improvement, the software running on the platform must be written in a way that it can spread its workload across multiple execution cores. This functionality is called thread-level parallelism or "threading". Applications and operating systems that are written to support it are referred to as "threaded" or "multi-threaded". A processor equipped with thread-level parallelism can execute completely separate threads of code. This can mean one thread running from an application and a second thread running from an operating system, or parallel threads running from within a single application.

These improvements have fuelled a constant advance in computing speed that has been quantified in the past by the Moore law: "the computing speed doubles every two years". However, due to the intrinsic limit of this process (a signal cannot be faster than light speed in the related medium), progress has involved other key aspects of the computer and has prompted innovative research in several relevant fields, including optical circuitry, molecular devices, quantum computers, etc.

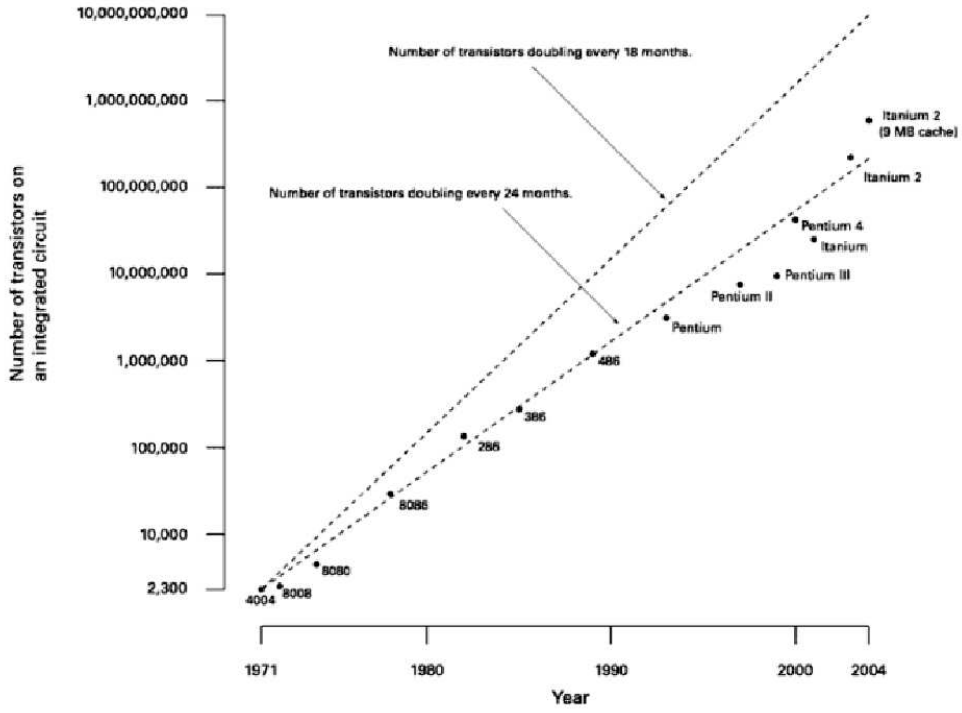

Figure 2.2: A graphical representation of the Moore's law.

2.2.1 Management concurrency

The first move towards concurrency is already built into the Von Neumann architecture that makes use of buses for parallel bit transfers. The concept of concurrent executing was, after all, already discussed by Babbage and proposed by Von Neumann for dealing with the various grid points of his differential computer. The first vacuum tube computer (ENIAC [75]) was made of 25 independent computing units. However, most of the early days progress was obtained at management level. The concepts themselves of

ISSN: 2279–8773

typing ahead, multiprogramming and time sharing were building up software and hardware concurrency in sequential (single CPUs) computers. Along the same line moved the use of dedicated (low level) CPUs for dealing with I/O internal communications, etc.

The real progress, however, was reached by making the circuitry extensively concurrent in several ways including CPU duplication, processing units segmentation, memory partitioning and, very recently, as already mentioned multi-core architecture. This single machine concurrency is more frequently referred to as "*parallelism*".

2.2.2 Instruction level parallelism

When the parallelism is exploited at instruction level (Instruction Level Parallelism, **ILP**) the concurrent execution of a multiple flux of instructions is adopted in order to maximize the performance. This is obtained by making use of the pipelining. In a pipelined architecture the CPU is usually partitioned up into stages for each operation (including instruction decoding, arithmetic, and register fetching) with each stage processing one instruction at a time. In some cases such a multiple flux can be sorted out (by the compiler as well as by the hardware) on the basis of a partial sorting induced by the logical dependencies of instructions, depending on the semantic of the program. The **ILP** paradigm was born in the seventies (together with superscalar architectures) by adopting a fine grain parallelism and evolving later in multiscalar and Very Long Instruction Word (VLIW) architectures.

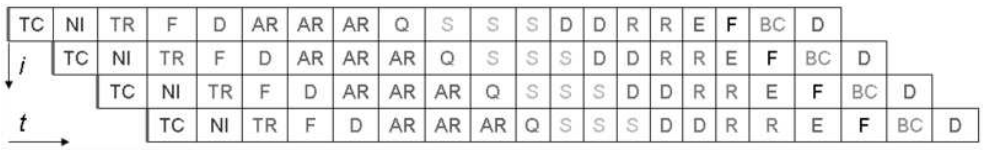

Figure 2.3: A graphical representation of a pipeline.

Superscalar Architectures

In **ILP** architectures, even if instruction are pipelined, in each stage of the pipeline only one instruction can be executed. In superscalar architectures (see Fig. 2.4) the machine-cycle phases are concurrently executed by simultaneously dispatching multiple instructions to redundant functional units on the processor. Each functional unit is not a separate CPU core but an execution resource within a single CPU such as an arithmetic logic unit, a bit

ISSN: 2279–8773

shifter, or a multiplier. The superscalar architecture is traditionally associated with several identifying characteristics applied within a given CPU core.

- Instructions are issued from a sequential instruction stream
- CPU hardware dynamically checks for data dependencies between instructions at run time (versus software checking at compile time)
- Multiple instructions per clock cycle are accepted

All executing functional units, due to the fact that each of them is implemented to perform a well established class of machine operations, are totally dedicated. The involved operations include floating/fixed-point operations as well as load/store operations (the last class involves data movements from/to the main memory). The degree of parallelism of these architectures is confined between three and six; processors belonging to this class are, for example, the MIPS 15000, DEC 21164, IBM Power4, ULtra Sparc III.

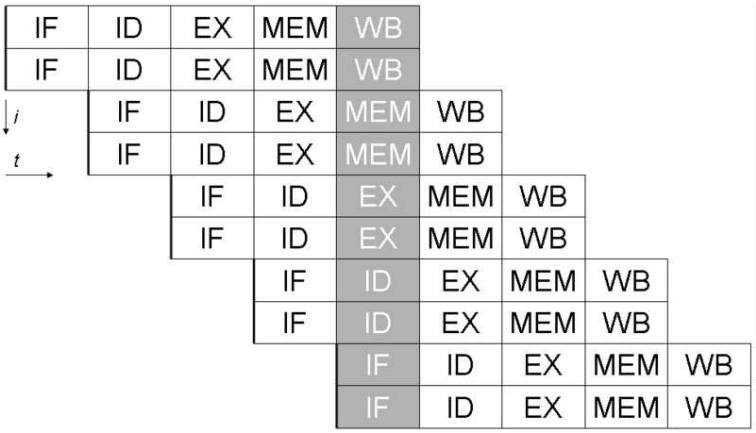

Figure 2.4: Superscalar architecture.

Multiscalar architectures

The most recent **ILP** paradigm is the Multiscalar [77]. In Multiscalar **ILP** architectures the granularity of the parallelism, exploited at instruction level, is greater than that of superscalar architectures. In these architectures the program loaded in the main memory is partitioned into many independent (in terms of logic) tasks which are distributed to the functional units, where the cycle-machine phases are applied to the instructions of the assigned task.

ISSN: 2279-8773

VLIW architectures

The VLIW (Very Long Instruction Word) architecture takes advantage of the capability of the compiler to compact independent operations in a bigger single instruction word, executing it on different functional units. The VLIW processor executes operation in parallel based on a fixed schedule determined when programs are compiled. Since determining the order of execution of operations (including which operations can execute simultaneously) is handled by the compiler, this means that the processor does not need the scheduling hardware. As a result, VLIW CPUs offer significant computational power with less hardware complexity (but greater compiler complexity) than is associated with most superscalar CPUs. Thus, it requires the adoption of refined compilers able to implement advanced techniques, such as Software pipelining and trace scheduling [78].

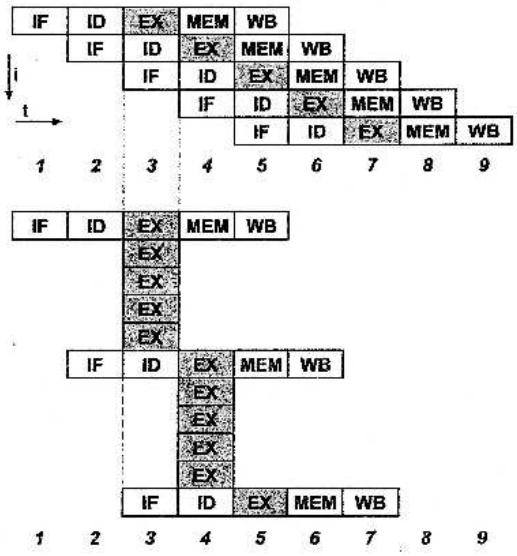

Figure 2.5: VLIW architecture.

2.2.3 Data Level Parallelism

The parallelism exploited at data level (data level parallelism, **DLP**) differs from the **ILP** in the granularity of the operands involved in the operations. Arithmetic operations are executed on data item arrays: in this way the paradigm becomes useful when one has to deal with applications involving a great number of vector and matrix elements operations.

ISSN: 2279-8773

Vector processors

Vector computers have been the first data parallel high performance computing systems. It has also been proposed to add a vector functional unit [79] to a superscalar processor. In a vector machine the flux of instructions passes through the Scalar Control Unit that takes care of submitting the instruction, if of the vector type, to the Vector Control Unit. Such an instruction is executed by a functional pipeline. In Fig.2.6 a register-register vector architecture is sketched. The vector operands as well as the vector results are stored in vector registers having a fixed (as in the CRAY computers) or dynamic (as in the Fujitsu VP200 Series) length [80]. The memory-memory vector architecture (adopted by the CDC CYBER 205) differs from the previous one since a streaming vector unit is used instead of the vector registers, in order to avoid memory-register traffic. However, such a model has not been successful since the memory access is very time consuming, even if it is useful when dealing with large arrays. In fact, in these processors the special vector registers can store up to N array values at the same time and each operation performed on a single register is propagated to the whole set of values stored on it. Obviously, in order to speed up this mechanism, the availability of a fast link with the main memory is crucial.

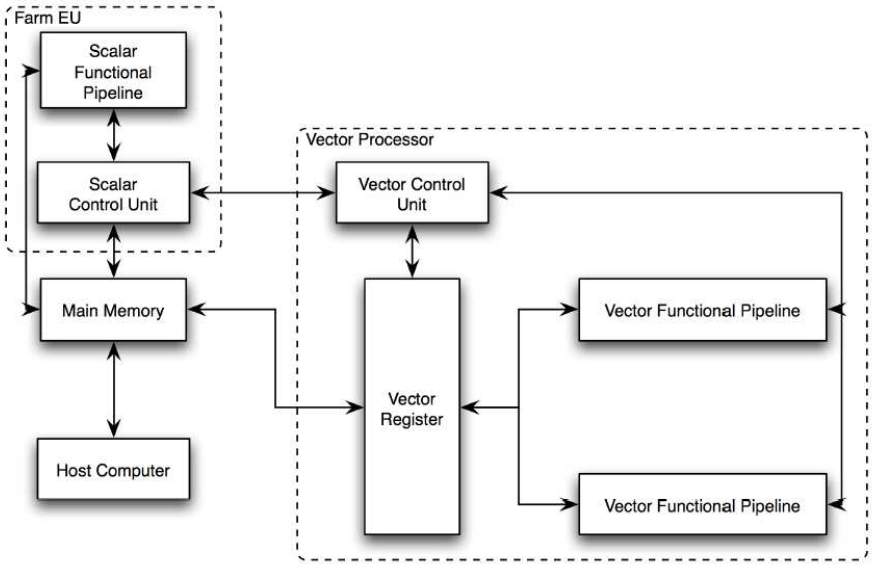

Figure 2.6: Vector architecture.

Array processors

An array processor performs only vector instructions, in fact the array processor architecture exploits the approach of executing concurrently the same instruction on several data. This approach guarantees high performances for programs involving a great number of vector instructions. An example of vectorial machine is represented by a vector co-processor (recently even PlayStation® and video cards are used) linked to an host-computer that loads in the main memory of the vector computer the program, together with the relevant data for the computation. The Instruction Unit (IU) fetches and decodes the instructions to be executed from the main memory and sends them to the Execution Units (EUs) (only if they involve arrays). The IU loads also into the Data Memories (DM) the operands involved in the array operations. Each EU-DM pair represents a Processing Element (PE). Such a model is sketched in Fig.2.7 and uses a distributed memory organization. As an alternative, we have the shared memory organization that is illustrated in Fig.2.8 which differs from the distributed memory organization because it can be simultaneously accessed by different EU with an intent to provide communication among them or avoid redundant copies.

Systolic array

A systolic array is a pipe network arrangement of processing units called cells. It is a specialized form of parallel computing, where cells (i.e. processors), compute data and store it independently of each other. A systolic array is composed of matrix-like rows of Data Drocessing Units (DPU) called cell which are similar to Central Processing Units (except for a program counter, since operation is transport-triggered, i.e., by the arrival of a data object). Each cell shares the information with its neighbours immediately after processing. The systolic array is often rectangular where data flows across the array between neighbour DPUs, often with different data flowing in different directions. The data streams entering and leaving the ports of the array are generated by Auto-Sequencing Memory units (ASM). ASM is an essential part of the anti machine paradigm. It is part of the instruction sequencer and is co-located with the datapath.

An example of a systolic algorithm might be designed for matrix multiplication. One matrix is fed in a row at a time from the top of the array and is passed down the array, the other matrix is fed in a column at a time from the left hand side of the array and passes from left to right. Dummy values are then passed in until each processor has seen one whole row and one whole column. At this point, the result of the multiplication is stored in

ISSN: 2279–8773

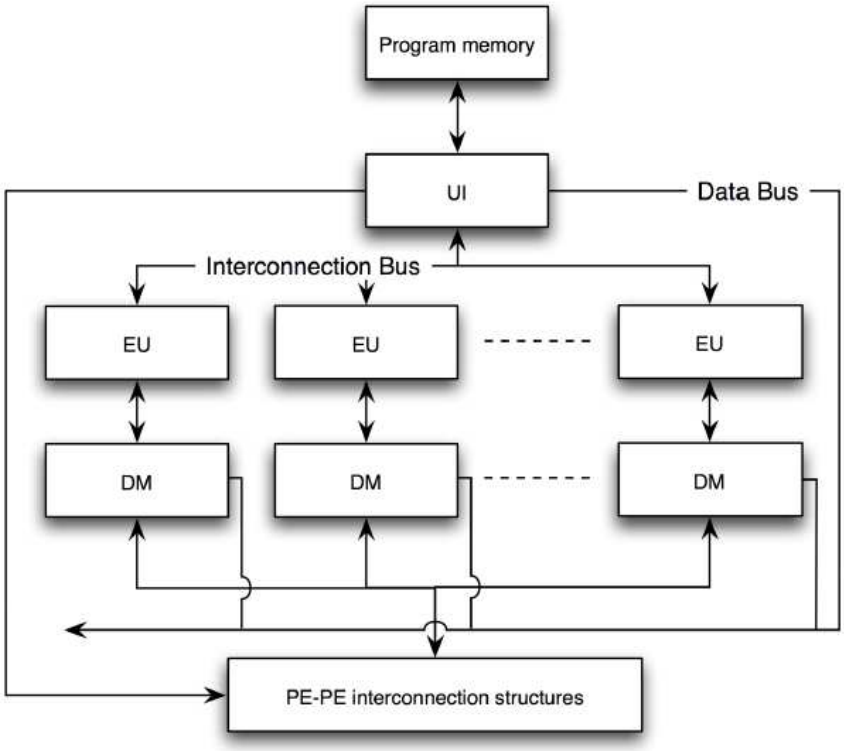

Figure 2.7: Array processor architecture with distributed memory.

the array and can now be output a row or a column at a time, flowing down or across the array.

2.2.4 Limits of the sequential architectures

Scalar architectures described in the previous sections are hitting the limit of their performances and most of the modern scientific advances could never be met using sequential computers (as an example, the simulation of a climatic model for a period of ten years, needs the execution of 10^{16} floating point operations, that needs, if performed on a superscalar architecture, about three years). In fact, in addition to the execution speed, that can be increased using the just mentioned management concurrency that includes **ILP** and **DLP**, there is a physical limit in directly connecting the single CPU to a sufficiently large memory. Let us consider the case of a scalar computer that has to execute in one second the following cycle (Teraflops speed):

```
Do i= 1 to 1000000000000
```

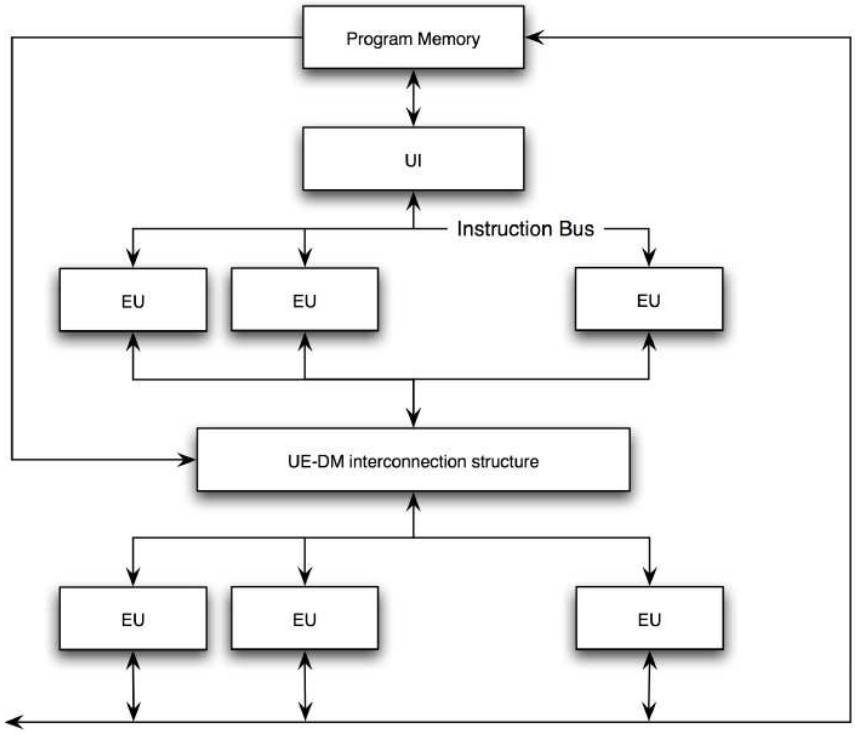

Figure 2.8: Array processor architecture with distributed memory.

```
      a(i)=b(i)+c(i)
EndDo
```

we need to transfer $3{\cdot}10^{12}$ variables from the memory to the CPU registers in one second. This implies that, if r is the mean distance of a word of memory from the CPU, the overall distance to be covered while transferring $3 \cdot 10^{12}$ variables in one second is $3 \cdot 10^{12} \cdot r$. Since the speed of light is $3 \cdot 10^8$ m/s one gets $r = 10^{-4}$ m. If we have $3 \cdot 10^{12}$ memory cells (each containing a word) packed as a matrix on a board around the CPU, then we have about 106 cells per row. This means that each cell cannot have a size larger than $10^{-6} \cdot r$ or 10^{-10} m that is the mean dimension of an atom. Therefore, since we are not able to store a 32/64 bit number into a location of the size of an atom we cannot build a scalar computer with a peak performance of 1 T *flops*. This leads to the conlusion that, in order to increase hardware performances we need to build platforms having many processors each surrounded by a local memory.

ISSN: 2279-8773

2.3 Concurrency on multiple processor

As we have already mentioned parallelism can be achieved in several ways by exploiting different architectural features. To this end, it is very useful to introduce a classification (taxonomy) that labels various architectures by the nature of the adopted parallelism.

2.3.1 Flynn taxonomy

Starting from the von Neumann machine model, that consists of one processor which executes sequentially a set of instructions to produce a single result, a classification can be based on the concept of streams. Two are the basic streams of a computer: instructions stream and data stream. This represents the basis of the taxonomy proposed by Michael J. Flynn in the sixties that reads as follows:

> "*The multiplicity is taken as the maximum possible number of simultaneous operations (instructions) or operands (data) being in the same phase of execution at the most constrained component of the organization*" M.J. FLYNN, 1966 [81]

Following these considerations, the taxonomy introduced by Flynn is articulated as follows:

- **SISD** Single Instruction stream Single Data stream
- **SIMD** Single Instruction stream Multiple Data stream
- **MISD** Multiple Instruction stream Single Data stream
- **MIMD** Multiple Instruction stream Multiple Data stream

SISD

The SISD class of architectures (see Fig.2.9) is the simplest one. In fact, it consists of a sequential machine which computes only one instruction on a single data item. In other words, the SISD family consists only of von Neumann computers. This definition however, does not consider any parallel organization of CPUs, such as the integration of multiple elaboration units on the same chip that, as we have already seen, is a popular practical solution.

ISSN: 2279–8773

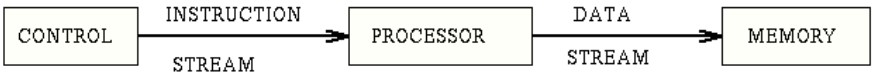

Figure 2.9: Sketch of SISD (Single Instruction, Single Data) machine.

SIMD

SIMD systems (see Fig.2.10) have a single control unit that executes one operation at time. They consist of several homogeneous processing units which can execute simultaneously that operation on different data sets. The central unit (control unit), in fact, acts like an emitter broadcasting the current instruction to be executed to the Processing Elements (PEs). **Cray 1**, **NEC SX-2**, **Fujitsu VP**xx and **APE** (or **QUADRIX**) are some examples of SIMD architectures. SIMD architectures can also be split in two classes:

- vector SIMD
- parallel SIMD

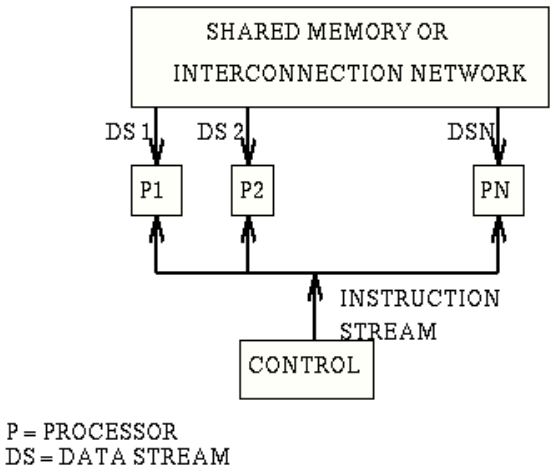

Figure 2.10: Sketch of SIMD (Single Instruction, Multiple Data) machine.

MISD

In the MISD case (see Fig.2.11) many instructions act simultaneously on the same data item. In this case, the granularity is represented by the processes

(by the multiple instruction orograms). This architectural model has never been popular due to its complexity as well as its futility.

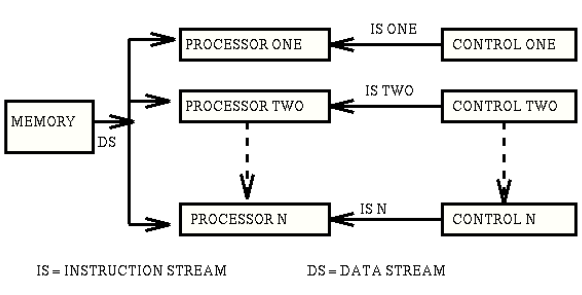

Figure 2.11: Sketch of MISD (Multiple Instruction, Single Data) machine.

MIMD

In MIMD machines (see Fig.2.12) the parallelism is exploited at a very coarse grain level on the execution tasks. This means that many PEs interpret different instructions on different data sets. Moreover, MIMD computers which make use of distributed memory are often referred to as thightly coupled machines (multiprocessors) while in case of a shared use of memory they are called loosely coupled machines (multicomputers). This class represents the multiprocessor version of SIMD architectures. Examples of computers belonging to MIMD multiprocessors class are ENCORE, MULTIMAX, SEQUENT & BALANCE; examples of computers belonging to MIMD multicomputers class are INTEL iPSC and NCUBE/7.

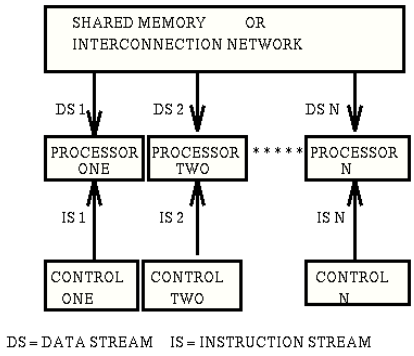

Figure 2.12: Sketch of MIMD (Multiple Instruction, Multiple Data) machine.

ISSN: 2279-8773

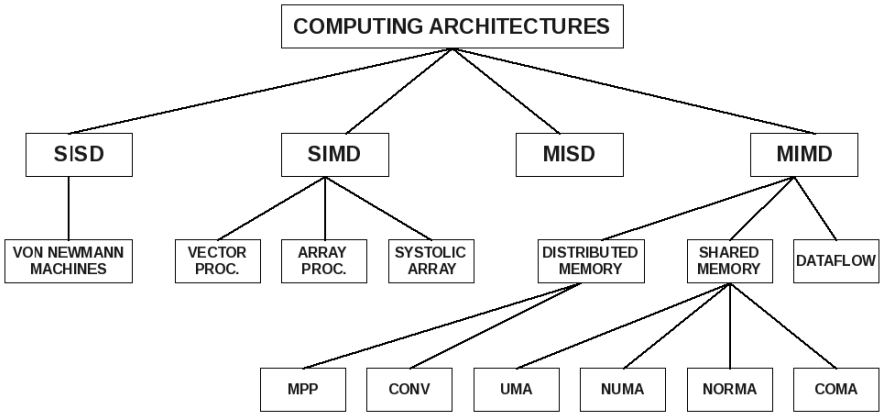

Figure 2.13: A summary sketch about computing architectures.

2.3.2 Other taxonomies

By blending flux, memory, interconnection and other parameters one can construct hibrid taxonomies:

Handler's classification [82] In 1977 Handler proposed a fairly elaborate notation for expressing the pipelining and parallelism of computers. Handler's taxonomy addresses the computer at three distinct levels: the processor control unit (PCU), the arithmetic logic unit (ALU) and the bit-level circuit (BLC). The PCU corresponds to a processor or a CPU, the ALU corresponds to a functional unit or a processing element in an array processor and the BLC corresponds to the logic unit needed to perform one-bit operations in the ALU. In particular, Handler's taxonomy makes use of three pairs of integers to describe a computer:

```
Computer = (k*k', d*d', w*w')
where
k = number of PCUs
k'= number of PCUs that can be pipelined
d = number of ALUs controlled by each PCU
d'= number of ALUs that can be pipelined
w = number of bits in ALU or PE word
w'= number of pipeline segments on all
    ALUs or in a single PE
```

For example, the **CRAY-1** is a 64-bit single processor computer whose ALU has twelve functional units, eight of which can be chained together

to form a pipeline. Different functional units have from 1 to 14 segments, which can also be pipelined. Handler's description of such a computer reads as:

```
Cray-1 = (1, 12*8, 64*(1 ~ 14))
```

Shore's taxonomy [83] Shore proposed his taxonomy in 1973. It is based on the structure and number of functional units incorporated into the computer. This taxonomy is characterized by six different categories, each of them associated to a number (**Type-I** $\sim$ **Type-VI**).

Hockney and Jesshope's taxonomy [84] Hockney and Jesshope developed an elaborate notation called Algebraicstyle Structural Notation (ASN) in order to describe parallel computers. This notation is the basis of their structural taxonomy. The taxonomy is arranged as several trees with a machine being described by all the labels in the parent nodes all the way back to the root node.

2.3.3 Memory architectures

As already mentioned above, Flynn's taxonomy is poor in classifying present parallel machines even if it is a very popular scheme. As already pointed out, the organization of the main memory is a key element in defining and classifying parallel architectures. By taking into account the memory structure of the machine (that is the organization of the main memory) three different models can be considered:

1. Distributed memory
2. Shared memory
3. Virtual Shared memory

This is particularly important for MIMD systems which exploit parallelism at coarse grain level. The organization of the memory can be, at an abstract level, either distributed or shared. The distributed memory model is illustrated in Fig. 2.14.

This model is referred in the literature as NORMA (*NO Remote Memory Access*). In this case the memory is distributed among the PEs, which are called computational (elaboration) nodes, each of which can be a multiprocessor. If all nodes are single processor machines, each node is made of a CPU that refers to its own memory. This means that a node can only access

the memory directly attached to it. The communication between different nodes is managed by a message-passing interface, thanks to the interconnection network. In this case, however, the user has to implement the needed communication patterns, at the programming level. NORMA systems are generally made of many CPUs neither very powerful nor expensive.

An example of the distributed memory machine model of the NORMA type is the **IBM SP5** at CINECA [85] made of 64 nodes p5-575 interconnected with a pair of connections to the Federation HPS (High Performance Switch). Globally the machine has 512 IBM Power5 processors, capable of 4 double precision floating point operations per clock cycle, and 1.2 TBs of memory. The peak performance of SP5 is 3.89 TFlops. A p5-575 node contains 8 SMP processors Power5 at 1.9GHz. 60 nodes have 16GBs of memory each, 4 nodes have 64GBs each. The IBM-SP5 runs AIX 5.3 Operating System.

Quite different is the shared memory model. In this case the PEs coordinate their activity, accessing to data sets and instructions, in a global, shared memory environment. All processing elements have direct access to the whole memory space via the interconnection structure. Moreover, nodes can be dedicated or not. In this respect, it is useful to single out two categories: UMA (*Uniform Memory Access*) and NUMA (*Non Uniform Memory Access*). In the UMA scheme (see Fig. 2.15) the processors work in an anonymous modality. Each process in a *ready* state (i.e. ready to be executed) is scheduled on the first available PE and the access time to the main memory is the same for all PEs. On the contrary, in the NUMA scheme (see Fig. 2.16) the PEs work in a dedicated modality. Accordingly, each node gets an allocated partition of the global machine resources. In particular, while the UMA model is closer to the shared memory scheme than the NUMA one, the NUMA model is made of subsets of processors with each subset having a local memory (**LM** in the figure) and I/O devices. The processors communicate among themselves via an interconnection structure which realizes a shared address space among the memory units (**M** in the figure). In other words, all the local memories share a global address space, accessible to the whole set of PEs (in other words, each PE can access, via remote operation, the local memory of the remaining PEs). Accordingly, local memory access time is shorter than in remote memory access. In the NUMA model the characteristics of both the simple shared memory and the distributed memory schemes coexist. This allows the NUMA model to go beyond the poor scalability of the UMA scheme when the number of processors and/or of the memory units (**M**) increases. In fact, when in a NUMA model the number of processors attempting to access the memory increases, the latency of the memory gets larger leading to progressively more severe bottlenecks. This

ISSN: 2279-8773

problem is partially circumvented in the NUMA model by using the local memory of the processor (**LM**).

The last model of this family is the *virtual shared memory* (VSM) architecture. The VSM architecture has a memory organization similar to both the distributed and to the shared ones. In fact, each node refers to a MMU (*Memory Management Unit*) linked with both the local memory and an interconnection structure implementing the global address space. The MMU decides if a referred object has to be fetched from the local memory or from the remote one, once the processor address is known. The remote access is made possible by a support circuitry handling the communication, independently from the address of the processor. Sometimes these architectures are implemented as all cache machines. Such a model, in particular, is a special case of the NUMA in which local memories have been converted into caches. This model is also referred to as COMA (*Cache-Only Memory Architecture*).

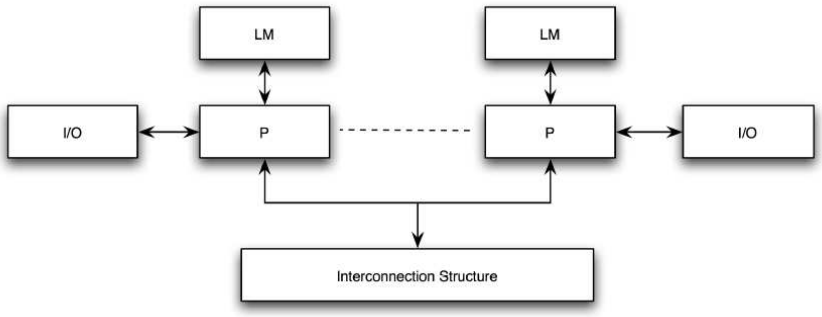

Figure 2.14: Distributed memory model.

2.3.4 The Interconnection infrastructure

The main difference between different MIMD architectures is concerned with the modality of data exchange. In fact, a MIMD system can be implemented in a variety of ways, depending on the adopted interconnection network, that represents a key point with respect to the performance in data exchange processes. Obviously, the choise of the Message Passing paradigm to be used for data, information and signal exchange is also very important.

The main topologies of interconnection networks can be grouped as follows:

Full connection : this topology represents the most powerful interconnection system since each node is directly connected to the others. If we

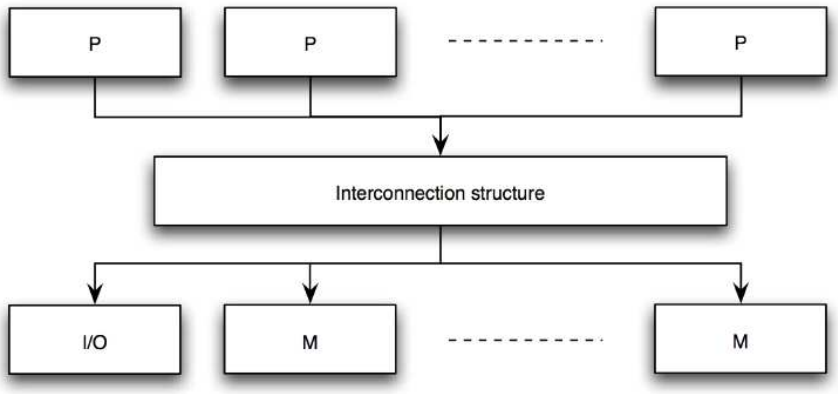

Figure 2.15: UMA model.

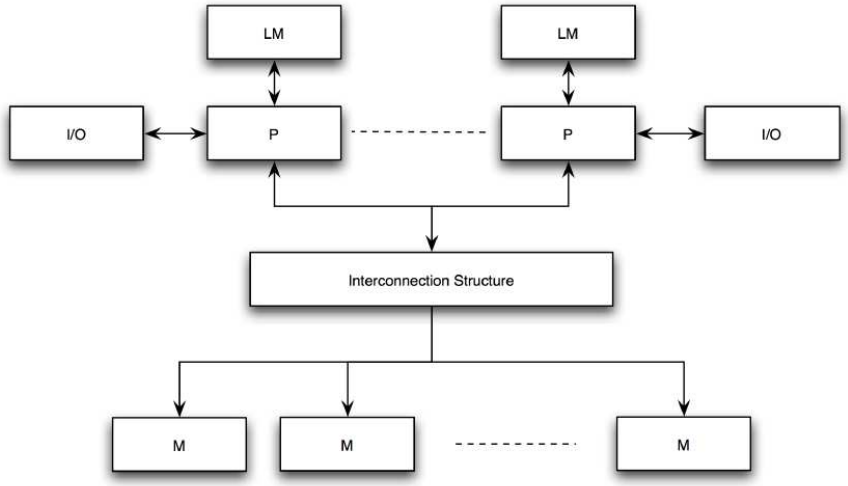

Figure 2.16: NUMA model.

have N processors, each node has $N-1$ connections giving an overall total number of connections equal to $N(N-1)/2$. Obviously, despite the great advantage of having all the nodes simultaneously connected (the bandwidth is proportional to N^2), this model becomes impractical if N is large.

Single shared bus : tis type of network topology in which all of the nodes of the network are connected to a common transmission medium which

has exactly two endpoints (this is the "bus", which is also commonly referred to as the backbone, or trunk). All data that is transmitted between nodes in the network is transmitted over this common transmission medium and is able to be received by all nodes in the network virtually simultaneously.

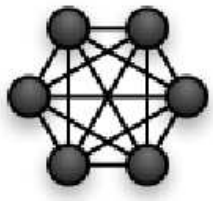

Figure 2.17: Full connection.

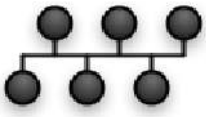

Figure 2.18: Bus.

k-dimensional Grid (Mesh) : in such a network nodes are collocated on a grid of dimension k and width w, thus we have a total amount of nodes equal to w^k (see Fig. 2.19). Communications are performed only with neighbours (each node is interconnected to $2k$ nodes). Some versions of this topology present *wrap-around* connections between the border nodes (toroidal topology).

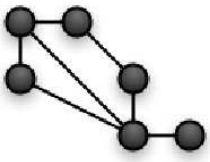

Figure 2.19: Mesh.

Tree and pyramid : a pyramidal network of dimension p is a complete quaternary tree with $\log_4 P$ levels, where the nodes of each level are connected by making use of a $2-D$ mesh (see Fig. 2.20).

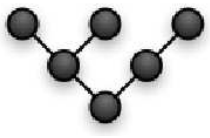

Figure 2.20: Tree.

Ring : it consists of a one dimensional array in which the final nodes are directly connected between them. In two dimension this is a thorus.

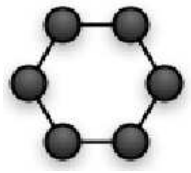

Figure 2.21: Ring.

Butterfly : a butterfly network has $(k+1)2^k$ nodes distributed among $(k+1)$ rows (ranks), each consisting of 2^k interconnected nodes.

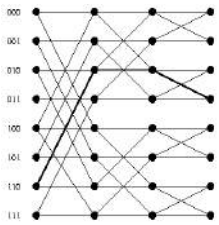

Figure 2.22: Butterfly.

Hypercube (Binary n-Cube) : such a topology consists of 2^k nodes arranged as an Hypercube of dimension k (see Fig. 2.23). The nodes are numbered from 0 to $2k-1$ and two nodes are connected only if their binary representations differ only for one bit.

Star : the type of network topology in which each of the nodes of the network is connected to a central node with a point-to-point link in a "hub" and "spoke" fashion, the central node being the "hub" and the nodes that are attached to the central node being the "spokes" (e.g., a collection of

point-to-point links from the peripheral nodes that converge at a central node). All data that is transmitted between nodes in the network is transmitted to this central node, which is usually some type of device that then retransmits the data to some or all of the other nodes in the network, although the central node may also be a simple common connection point (such as a "punch-down" block) without any active device to repeat the signals.

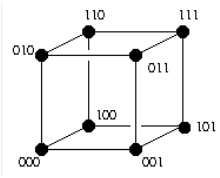

Figure 2.23: Hypercube.

Figure 2.24: Star.

Cross-bar switch : a crossbar switch is a switch topology where every node can be connected to any other node in the system. This topology has traditionally been used for high-performance computing systems since it can provide concurrent independent data paths between pairs of nodes in the system for maximum total system bandwidth. This topology is also flexible because the connections between node pairs can be changed dynamically as needed for optimum system communication.

Network of computers : this is the case of a single communication channel shared by all nodes of the system. Each node can be either a workstation or a PC (or both if we are not interested into having homogeneity) and the interconnection network is, typically, a LAN. Although as an individual platform this model is rather poorly performing, it is the reference scheme for Grid computing. In the case of Grid computing each computing apparatus is considered as a working unit and the public network must provide access for work coordination.

ISSN: 2279-8773

Clusters : this is a particular cost effective platform representing a new trend in concurrent computing. Clusters are essentially farms i.e., a set of independent servers each connected to a central unit, called front end, that is directly interfaced with the outer network. The nodes are normally Pentium, or equivalent PC like units having good memory sizes and equipped with low cost hard disks. The interconnection network consists of a switch, typically of Gigabit technology. This solution makes it feasible to build a parallel system in a simple and cheap way.

2.4 Concurrent computing

Along with the evaluation of computer architectures and platforms, program design and organization models are needed to provide users with suitable tools to implement concurrency in their applications.

The basic sequential machine programming paradigm is the execution of a set of instructions. An instruction can specify, in addition to various arithmetic operations, the address of data to be read or written in memory and/or the address of the next instruction to be executed. While it is possible, in principle, to program a computer using this basic scheme directly in machine language, this is for most purposes impractical since one needs to keep track of millions of memory locations and manage the execution of thousands of machine instructions. Hence, modular design techniques are applied, whereby complex programs are constructed from simple components, and components are structured in terms of higher level abstractions such as data structures, iterative loops, alternative sequences and procedures. Abstractions (like the procedures) make the exploitation of modularity easier by allowing objects to be manipulated without concern for their internal structure. So do high-level languages such as Fortran, Pascal, C, C++. These high level (artificial) languages allow program design expressed in terms of abstractions to be translated automatically into executable code.

On the other hand, parallel programming introduces additional sources of complexity with respect to sequential programming if we were to program at the lowest level. In this case, in fact, not only would the number of instructions to be executed increase, but the execution of thousands of processes and the coordination of millions of interprocess interactions would also need to be managed explicitly. Hence, abstraction and modularity are at least as important as in sequential programming. In order to develop a parallel application one has to look at well established programming paradigms. Three are the main paradigms in parallel computing:

Message passing: this is probably the most widely used parallel program-

ISSN: 2279–8773

ming paradigm today. Message-passing applications create multiple tasks, with each task encapsulating local data. Each task is identified by a unique name, and tasks interact by sending and receiving messages to and from named tasks. The message-passing paradigm does not preclude the dynamic creation of tasks, the execution of multiple tasks per processor, or the execution of different programs by different tasks. However, in practice most message-passing systems create a fixed number of identical tasks at program startup and do not allow tasks to be created or destroyed during program execution.

Data parallelism: another commonly used parallel programming paradigm that calls for the exploitation of the concurrency deriving from the application of the same operation to multiple elements of a data structure. A data-parallel application consists of a sequence of such operations. As each operation on each data element can be thought of as an independent task, the natural granularity of a data-parallel computation is coarse, and the concept of locality does not arise naturally. Hence, data-parallel compilers often require the programmer to provide information about how data are to be distributed over processors, in other words, how data are to be partitioned into tasks. The compiler can then translate the data-parallel program into an SPMD formulation, thereby generating communication code automatically. The implementation of the data-parallel paradigm is represented, as an example, by the High Performance Fortran (HPF) parallel programming language.

Shared memory: for this programming paradigm tasks share a common address space, which they read and write asynchronously. Various mechanisms such as locks and semaphores may be used to control access to the shared memory. An advantage of this paradigm from the programmer's point of view is that there is no notion of data ownership, and hence there is no need to specify explicitly the communication of data from producers to consumers. This paradigm can simplify program development. However, understanding and managing locality becomes more difficult and this is an important consideration that need to be considered on most shared-memory architectures. It can also be more difficult to write deterministic programs.

In this section only the message passing paradigm will be taken into account since it represents the *de-facto* standard in directive parallel programming.

2.4.1 The *a priori* design of a parallel application

Apart from the adopted paradigm, the methodological design of a scalable parallel application must follows well determined steps. In fact, most programming problems can be tackled using different parallel approaches. The best solution may differ from that suggested by existing sequential algorithms. The design methodology described here and proposed by Ian Foster [86] is intended to foster an exploratory approach in which machine-independent issues (such as concurrency) are considered early and machine-specific aspects are delayed until late (or if possible left with specific software of the machine). This methodology structures the design process as four distinct stages: partitioning, communication, agglomeration, and mapping, as sketched in Fig. 2.25.

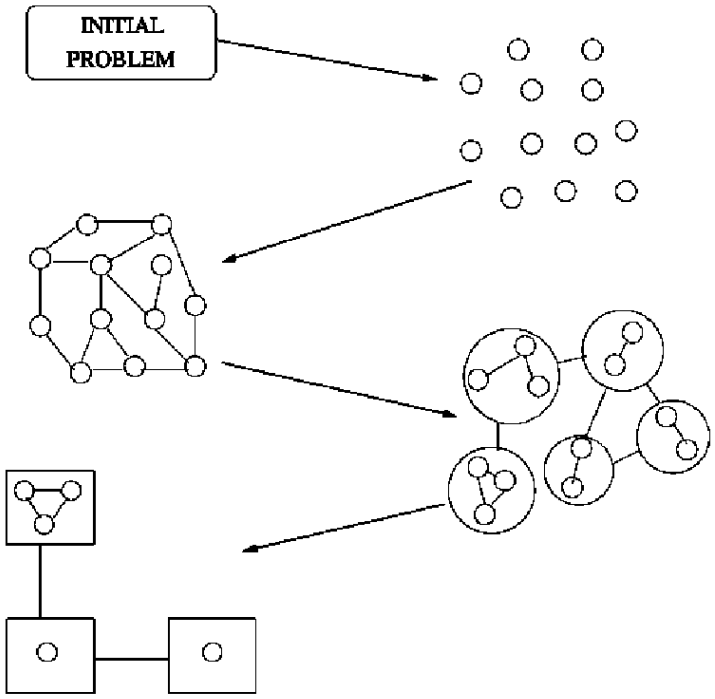

Figure 2.25: A design methodology for parallel applications.

Partitioning

The computation that is to be performed and the data involved are decomposed into small tasks. Practical issues such as the number of processors

in the target computer are ignored, and attention is focused on recognizing opportunities for parallel execution. In particular, this stage consists in defining the computational grain of the parallel application, taking care of pushing the partitioning at the lowest level ensuring a good flexibility. Such a partition can be undertaken at both data and computation levels:

- *domain decomposition:* this technique first decomposes the data associated with a problem and then partitions the computation that is to be performed, typically by associating each operation with the data on which it operates.

- *functional decomposition:* this technique represents a different and complementary way of thinking about problems. In this approach, the initial focus is on the computation that is to be performed rather than on the data manipulated by the computation. After being successful in dividing the computation into disjoint tasks, one proceeds into examining the data requirements of these tasks. These data requirements may be disjoint, in which case the partition is complete. Alternatively, they may overlap significantly, in which case considerable communications will be required to avoid replication of data.

Communication

Once the partition into very small tasks of data and computations has been performed, the communication required to coordinate the execution of the farious tasks needs to be determined, and appropriate communication structures and algorithms need to be defined. The tasks generated by the partitioning can, in general, execute concurrently though they cannot execute independently. Usually the computation to be performed in one task will require the communication of data associated with another task. Communication requirements differ depending on whether one has performed a domain or a functional decomposition. In fact in the former case, the implementation of communications between different tasks can be difficult establish since partitioned data often remains tightly coupled. On the contrary, if a functional decomposition has been adopted, communications can be easly decoupled.

Agglomeration

The tasks and communication structures defined above need to be evaluated with respect to performance requirements and implementation costs. This may request that individual tasks are combined into larger tasks to improve the performance or to reduce the development costs. At this stage one has

ISSN: 2279-8773

to decide if the granularity, derived by the partitioning stage (in which as many tasks as possible have been defined), is acceptable or if one has to push the granularity to a coarser level. In general, one looks for tasks (and communications among them) to be associated into greater tasks. At the end of this stage it is advised to minimize the overall communication and to obtain a number of tasks larger than the number of processors.

Mapping

Tasks are assigned to the various processors in a way that attempts to maximize the processor utilization and minimize the communication costs. The main goal of this stage is to ensure an optimal load-balancing between different nodes (processors) of the platform. In this final stage we have to decide where each task has to be executed, depending on the used platform. In general two strategies can be adopted:

- Tasks which can be executed independently are mapped on different physical processors
- Tasks which keep a high degree of coupling are mapped on the same processor.

Clearly, these two strategies might sometimes conflict. In this case the design will require some tradeoffs. Also, resource limitations may restrict the number of tasks that can be mapped on a single processor or the number of processors that the tasks may use. In order to perform the mapping of the tasks there are many load balancing algorithms that help the user. Obviously the user based on his/her knowledge of the application may decide to apply one of them like recursive bisection, local algorithm, probabilistic method, etc. The user can also choose of leaving to the Load Balance utility of the machine to take care of the problem. Load Balancing struggle to avoid the unshared state in processors which remain idle while tasks compete for service at some other processor equalizing the load on all processors. Algorithms for load balancing have to rely on the assumption that the on hand information at each node is accurate to prevent processes from being continuously circulated about the system without any progress. This is one of prerequisites to utilize the full resources of parallel and distributed systems. Load balancing may be centralized in a single processor or distributed among all the processing elements that participate in the load balancing process.

Such load balancing algorithms are the *Recursive Bisection*, the *Local algorithm*, the *Probabilistic Method*, the *Round-Robin* and the *biasing algorithm*.

ISSN: 2279-8773

2.4.2 Models of parallelism

The *a priori* design of a parallel application is driven by the adoption of some standard models of parallelism. These models can be classified on the basis of the exploited parallelism:

*F*lux parallelism:

- *seq* in which parallelism is not exploited
- *farm* in which each process is assigned a determined work-load, by a central unit that manages the distribution of tasks. This model is also called Master-Slave or Master-Worker paradigm)
- *pipe* the whole computation is partitioned into different stages each of which is assigned to a process as in a chain. Each stage of the pipe can be parallelized at a finer grain
- *loop* in which a block of processes is iterated in time

On the contrary, in *data parallelism* models there are few dependencies among data and there is no need for a great number of interactions:

- *map* completely independent data
- *reduce* subsequent associative reduction of data
- *comp* sequential composition of modules with some interactions

In Fig. 2.26 a sketch of the *farm* model is given.

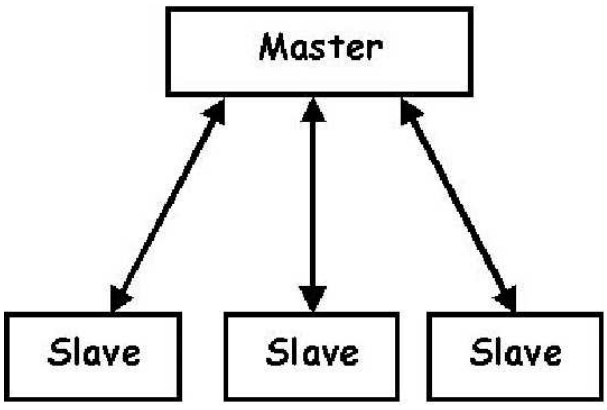

Figure 2.26: The farm model.

ISSN: 2279-8773

2.4.3 Tools for parallel programming

Message Passing Interface

As already pointed out above, an important paradigm for parallel computing is Message Passing. This paradigm has been developed in order to be used on distributed memory platforms although it can be also implemented efficiently on a shared memory architecture. In the message passing paradigm a computation consists of one or more processes that communicate by calling library routines to send and receive messages to/from other processes. In this respect, two are the main libraries that implement the paradigm: PVM (Parallel Virtual Machine) [87] and MPI (Message Passing Interface) [88]. We are interested here only into MPI implementations (in particular, MPI-1 implementation) since MPI is increasingly becoming the standard library both in homogeneous (MIMD platforms) and heterogeneous (GRID systems) environments. Accordingly, we can summarize the main goals of MPI as follows:

- provide source-code portability
- allow efficient implementations
- offer a great deal of functionality
- support heterogeneous parallel architectures

In the message passing programming model, each process has a local memory and no other process can directly read from or write to that local memory. Parallel programming by definition involves cooperation between processes to solve a common task. There are two sides to the question of programming processes that cooperate each other. The programmer has first to define the processes that will be executed by the processors and then to specify how those processes synchronize and exchange data with one another (as already specified at an abstract level in the previous section). A central point of the message-passing model is that, obviously, the processes communicate and synchronize by exchanging messages. As far as the processes are concerned, the message passing operations are just calls to a message passing interface that is responsible for dealing with the physical communication network linking the processors.

Prior to exchanging messages it is needed to determine the "communication universe", in which point-to-point or collective operations can act, by defining a communicator. Each communicator contains a group of processors and the source and the destination of a message is identified by the process

ISSN: 2279–8773

rank within that group. In MPI there are two types of communicator here discussed

- **Intracommunicator** used for communicating within a single group of processes,
- **Intercommunicator** used for communicating within two or more groups of processes (in MPI-1, an intercommunicator is used for point-to-point communication between two disjoint groups of processes).

Messages are central to the message passing programming model. They are exchanged between processes. When two processes exchange a message, data is copied from the memory buffer (or memory locations) of one process into the memory buffer of the other process. The data sent in a message comes under two headings: *contents* and *envelope*. The contents of the message are pure user data and are not interpreted neither by the communication interface nor by the communication system that lies behind that interface. The data on the envelope, however, is used by the communication system to copy the content of the message between local memories.

The simplest form of message is a *point-to-point* communication in which a message is sent from a sending process to a receiving process. Only these two processes need to know anything about the message. The communication itself consists of two operations: send and receive. The send operation can be either synchronous or asynchronous depending on whether or not it can be completed[1] before or after the corresponding receive operation has started. *MPI point to point* prinitives can be classified in blocking or non blocking procedures. The blocking ones return the control only when the corresponding communication is completed. The non blocking ones return the control straight-away and allow the process to continue performing other work. Many message-passing systems do also provide primitives allowing large numbers of processes to communicate. Such primitives implement the so-called *collective communications* and they are of blocking type. Examples of these are barrier, which synchronizes the processors, broadcast, which allows a one to many communication and reduction operations which takes data items from several processors and reduces them to a single data item that may or may not be made available to all of the participating processes. Moreover, the library includes primitives which are particularly designed to perform a collective domain decomposition. For example, the most commonly used are:

[1] the completion of the communication means that memory locations used for the message transfer can be safety accessed. MPI communication modalityes differ form the conditions needed to the completion.

ISSN: 2279–8773

- scatter (*one-to-all communication*): different data are sent from the root process to all the others in the MPI communicator (including the root process)

- gather (*one-to-all communication*): different data are collected by the root process from all other processes in the communicator (including the root process). It is the opposite of the scatter primitive.

A synopsis of the most commonly used MPI primitives are listed in Table 2.1.

Table 2.1: Most common MPI primitives.

Primitive	Description
MPI_comm size	Creates the communicator processes
MPI_comm rank	Gives the rank to each process of the communicator
MPI_send	Sends a message to another process
MPI_recv	Receives a message from another process
MPI_barrier	Blocks the caller until all processes have called it
MPI_bcast	Broadcasts a message form one process to all processes
MPI_gather	Each process sends a message to a root process that receives and stores it in rank order
MPI_scatter	The inverse operation of gather
MPI_reduce	Combines the elements of an incoming data using a pre-determined operators in the root memory space

Clearly, despite its great portability as well as its diffusion, MPI shows some severe limitations:

1. the management of the communication is entirely on the hands of the programmer

2. the library does not provide models and tools for efficiency evaluation (e.g. to predict the scalability of the application on the basis of the implemented granularity) and a profiling of the program in this sense is needed;

3. portability sometimes is difficult and a deep restructuring may be necessary to implement the parallel code;

ISSN: 2279–8773

4. due to its explicit nature, MPI is *error-prone*.

A pictorial representations of Broadcast, Gather and Scatter primitives are given in Fig. 2.27.

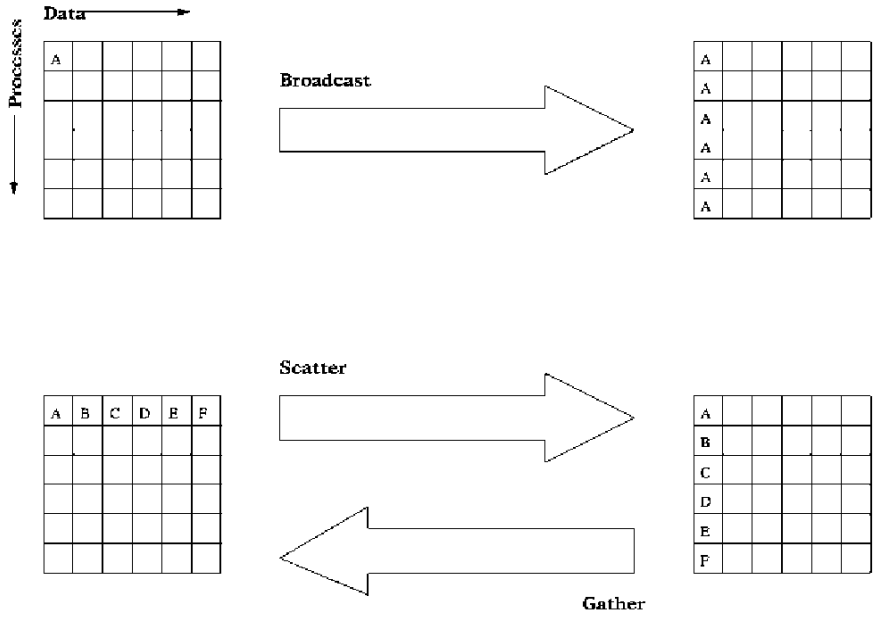

Figure 2.27: A pictorial view of some collective primitives.

Charm++

Very recently it has been developed at the University of Illinois, a machine independent parallel programming language written in C++ called Charm++. The design of the system is based on the following tenets:

- Efficient Portability: Charm++ programs run unchanged on MIMD machines with or without a shared memory. The programming model induces better data locality, allowing it to support machine independence without losing efficiency.

- Latency Tolerance: Message-driven execution, supported in Charm++ is a mechanism for tolerating or hiding communication latency. In message driven execution a processor is allocated to a process only when a message for the process is received. This means that when a process blocks, waiting for a message, another process may execute on the processor.

ISSN: 2279–8773

- Dynamic Load Balancing: Charm++ provides dynamic (as well as static) load balancing strategies.

The package consists of potentially medium-grained processes (called chares), a special type of replicated process, and collections of chares. These processes interact with each other via messages. There may be thousands of medium-grained processes on each processor, or just a few, depending on the application. The "replicated processes" can also be used for implementing novel information sharing abstractions, distributed data structures, and intermodule interfaces. The programming language can be considered a concurrent object-oriented system with a clear separation between sequential and parallel objects.

2.4.4 The evaluation of performances and scalability

Execution time is not always the most convenient metric by which to evaluate parallel algorithm performance. Since the first goal of parallel programming is to work out an application being faster than the related sequential implementation, one needs some parameters able to quantify the achievable performances. In this respect we must be able to evaluate the timing of both the parallel and sequential application in order to work out the so called *Speed up* (S) and *Efficiency* (E). The speed up S is the most used parameter to evaluate the performance of a parallel application. It gives an estimate of the variation of the performance of a program which the number of used processors. As a matter of fact the speedup is calculated as:

$$S(n) = \frac{T_s}{T_p} \tag{2.1}$$

where T_s represents the execution time of the best sequential implementation of a given algorithm and T_p is the execution time of the related parallel implementation. The speed up is a pure number and the greatest value it can reach is equal to the number of processors used in the calculation. At the same time, the efficiency of a parallel application is given by:

$$E = \frac{S(n)}{n} \tag{2.2}$$

The efficiency can be seen as the fraction of time that the processors spend in doing useful work. Sometimes it may happen that the calculated speedup is greater than the number of processor. This behaviour is called *superlinear*. This effect is ususally related to a better use of fast memory, such as cache memory. In fact, if data are kept in cache memory (rather than

ISSN: 2279–8773

in the physical memory) during the calculation, costs of memory access are reduced and the efficiency is greater than 1 (and the speedup superlinear).

A model able to predict the theoretical maximum speed-up using multiple processors is the Amdahl's law which compare the expected speedup of parallelized implementations of an algorithm relative to the serial algorithm, under the assumption that the problem size remains the same when parallelized. More technically, if P is the proportion of a program that can be parallelized and $(1 - P)$ is the proportion that cannot be parallelized (i.e. it is strictly serial), then the maximum speedup that can be achieved by using N processors is given by the equation

$$\frac{1}{(1-P) + \frac{P}{N}} \tag{2.3}$$

2.5 Concurrency on the network

In very recent times Grid computing has emerged as a new important field of concurrent computing in heterogeneous environments. Grid computing is focused on large-scale resource sharing, innovative applications and, clearly, high throughput computing. The Grid concept was effectively born in the mid '90s and it has been defined as [89]:

> *A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive and inexpensive access to high-end computational capabilities.*

In this respect, the Grid deserves to be considered separately from parallel computing. Grid computing is not aimed at high-performance computing. It is rather focused on high-throughput computing for complex computational applications which need the gathering of a large ensemble of computer resources and expertises. Accordingly, the basic problem lying behind the Grid concept is related to the coordinated sharing of the resources. This means easy access to all kind of available computing platforms, software, data and other resources like human skills.

2.5.1 The foundations of Computer Grids

Computer Grids are characterized by the heterogeneous nature of their wide ensemble of computing hardware components and by the composite nature of the applications considered. Therefore the establishing of a production Grid requires the implementation of several components.

ISSN: 2279–8773

The first characteristic feature of a Grid is the composition of different expertises related not only to the involved hardware, but also to the specific components of the problems tackled. The management of these competence based instrumentation and tools requires specific attention when building demonstrators for the Grid.

The second critical feature is given by the nodes of the Grid. The user can make direct use of just one type of computer, typically a desktop workstation, not only to write, debug and compile the codes but also to launch the simulation. A bunch of high performance computers or supercomputers can be chosen to take care of running the codes and crunching the data fed to them. Other computer may be used to take care of rendering the results in a graphical form or to power a virtual reality display device.

The third feature of a Grid is the communication software to make the whole collection of codes user-friendly. Communication software bridges all of the gaps, between different computers, between computers and people, even between different people. This turns the physical connections between computers from a collection of individual machines into an interconnected computing system.

The fourth feature of the computational Grid is the physical network that links the various machines, for example, via modem, ISDN, standard Ethernet, FDDI, ATM, or other networking technologies. Networks with high bandwidth and low latency are the most favored to provide rapid and reliable connections between the machines. To actually communicate over these physical connections it is also necessary to have some smart communication software running.

Given that we have an interconnected, communicating network of computers, processors with memory, one further component is needed. This fifth feature is something like an operating system that can be used to configure, manage, and maintain the Grid computing environment. This virtual environment needs to span the extent of the computational Grid and makes it usable by both administrators and individual users. Such an environment will enable machines and/or instruments that may be located in the same building, or separated by thousands of miles, to appear as one system. This virtual environment, therefore, must provide the administrators all the functions which allow her/him to time the system management tools to deal with a changing heterogeneous platform. This software needs also, to speak, on the side of the user, the language he is used to and has, therefore, to fall into the category of problem solving environments (PSE). More in detail such an environment has to allow the user to get the best from the platform both in preparing input data and at running time with no need to get involved in managing related technicalities. The most popular name for this type of

ISSN: 2279-8773

software is "middleware" and still needs a lot of effort to be designed and implemented. Presently, the standard in Grid computing is represented by the gLite middleware [90], developed through collaborative efforts of more than 80 people in 12 different academic and industrial research centers as part of the EGEE Project. gLite derives from the Globus middleware [91] developed at the Argonne National Laboratory.

2.5.2 The various middleware

The middleware that most contributed to the development of the computing Grids is Globus. Globus provides a toolkit based on a set of existing components with which build a metacomputing environment. Globus Metacomputing Toolkit allows to construct a Computational Grid, which means a distributed computing environment for high performances, and to execute a set of applications using different computational models. Each user can select the application suitable for its system or adapt it to its personal use. Graphic interfaces, called translucent, have been introduced to manage tools and applications and recognize and control mechanisms at low level. This gives the chance of optimizing performances and adapt all the configurations of the system to the user necessities. An information system is part of the toolkit. Thanks to the grid configuration, it is possible to use different networks and computers with respect to problems related to hardware and software facilities. The programmer needs not to define an *a priori* static configuration of the application environment, but the grid offers tools which dynamically find out resources and configurations of the system for an efficient execution, allocating them in a transparent way. The Globus Project is organized around four main activities:

1. **Research**: The study of basic problems in areas such as resource management, information services, security and data management is addressed. It focuses not only on the issues associated with the building of computational grid infrastructures, but also on problems arising from the design and the development of parallel applications that use grid services. Uniform and scalable mechanisms for naming, locating, and allocating computational and communication resources in distributed systems have been developed so far. Basic Grid services have been integrated into existing application development frameworks, environments and languages (e.g. CORBA, Java, Perl, Python). A collaborative effort has been launched to design and produce an infrastructure-level architecture for data management, which is called the data grid. Finally, requirements, designs and prototypes of a Grid information service as

ISSN: 2279–8773

well as an enabler for dynamic application configuration and adaption, have been worked out.

2. **Test beds**: The Globus team supports and assists initiatives for planning and building large-scale prototyped packages acting as test-beds, both for its own research and for production use by scientists and engineers.

3. **Software tools**: Pieces of software running on a variety of platforms acting as general instruments available on the grid have been developed and are continuously being updated.

4. **Applications**: Several large-scale packages have been designed and implemented in a cooperative fashion by scientists and engineers on the grid to serve as commonly used applicative software. To this end basic technologies enabling entirely new classes of applications have been developed. The net result is a set of programs allowing to advance in the understanding of how to build programs for the grid, how to focus the research efforts of the project and how to evaluate the utility of the tools developed.

The Globus toolkit adopts many software components that implement the above described services. Other middleware have been proposed. The one that has become the race horse of the Grid European projects is gLite that will be illustrated in detail later on.

2.6 From Metachem to GridChem

In order to experiment the assemblage of a suitable Grid infrastructure in Italy, a national FIRB project [92] called "Piattaforme abilitanti per griglie computazionali ad alte prestazioni orientate a organizzazioni virtuali scalabili (shortly called GRID.IT)" was launched in the year 2002 [93]. GRID.IT was aimed at gathering together at national level the efforts of desining advanced networking hardware, defining appropriate middleware and implementing grid enabled applications. On the side of designing grid enabled calculations codes for molecular simulation, a workpackage (WP13) was established. The main goal of WP13 was the assemblage of an *ab initio* Simulator called SIMBEX (SImulator of Molecular Beam EXperiments). For this purpose a prototype Grid infrastructure called CHEMGRID was also assembled. CHEMGRID was built around a cluster of computers owned by the Chemistry Departments of the Universities of Bari, Bologna, Naples and

ISSN: 2279-8773

Perugia, the Milan, Padua and Perugia local sections of the ISTM CNR Institutes and the Computational Chemistry laboratory of ENEA at the Casaccia location. CHEMGRID, managed by the homonimous cluster of computers of the University of Perugia, was also linked to the computing resources of the Department of Physical Chemistry of the University of the Basque Country in Vitoria Gasteiz (Spain) and to the computing resources of the Department of Physical Chemistry of the University of Barcelona in Barcelona (Spain).

2.6.1 From COST D23 METACHEM to GRID.IT

The involvement in Grid of the molecular science community begun some years before when the Computational Dynamics and Kinetics (CDK) research group of the Department of Chemistry of the University of Perugia launched the COST Action D23 " METACHEM: metalaboratories for complex computational applications in Chemistry" within the COST Chemistry domain. The idea was to set some Metalaboratories, defined as a set of geographically dispersed physical laboratories sharing expertise, software and hardware on a computing Grid to carry out collaborative work (relevant references can be found in the "Memorandum of Understanding" of the COST in Chemistry D23 Action [94] and in refs. [95, 96]).

The clustering together of geographically distributed laboratories willing to share expertise, software and hardware on a computing Grid was, indeed the main goal of the Metalaboratories gathered in METACHEM. The Metalaboratories being made of several physical research laboratories acting as reservoirs of specific chemical expertise were found to be the ideal environment for the realization of large computational projects on molecular structures and processes based on first principle. For this reason one or more computer centers (or laboratories particularly skilled in information and communication technologies) were acting as regulators of the Grid and, when it was the case, other laboratories having complementary expertise (for example experimental facilities) were also gathered together. The main goal of METACHEM Metalaboratories was to realize by gathering specific competences, programs, environments and hardware *a priori* multiscale simulators starting from the microscopic level. These multiscale *a priori* molecular simulators were intended to constitute the computational engine for usage in atomic and molecular sciences and related applications (like those for pharmaceutics, materials, environment, biology, life science).

Altogether METACHEM has implemented six Metalaboratories. They are made of more than 40 physical laboratories belonging to 17 European Countries. The main sharacteristics of the implemented Metalaboratories are given in Table 2.2.

ISSN: 2279–8773

2.6.2 From GRID.IT to COMPCHEM

The mentioned GRID.IT project was just one of the various European projects implemented to build a new scientific and technological ICT platform for large scale distributed computing.

Like most of these projects GRID.IT had a strong interdisciplinary character. Therefore one of its primary goals was that of defining, implementing and applying innovative solutions exploiting network computing enabling platforms, oriented towards the constitution of scalable Virtual Organization (VO)[2] operating on the shared Grid Computing platform.

This means also that a unifying approach was adopted in the development of the applications as well as in choosing programming models and environments. Such an approach was meant to take into account both the aspects related to the distribution of computations and resources. The programming environment was though to be characterized by absolute portability on different hardware-software systems (or different hardware-software combinations) in a heterogeneous and dynamic context an portability to be guaranteed not only for codes but also for tools and data.

GRID.IT was articulated in layers. At the basic layer an important role was played by research on high speed networks based on photonic technology for Grid platforms with high performance sites covering metropolitan areas.

At an upper layer there were research lines in Middleware concerned with programming environments, resource management and software technologies articulated as:

- Security: guarantee secure Grid environments and cooperation among Grid environments belonging to different organizations;

- Data intensive services: offer federated database services, visualization and hierarchical management of data and meta-data according to advances and high-performance techniques;

- Knowledge discovery services: provide Grid services (data mining, search engines, etc.) guaranteeing consistent, efficient and pervasive access to high end computational resources allowing exploitation of the Grid as a sophisticated Web computing tool;

- Grid portals: deliver Grid enabled applicative services like submitting tasks and collect results to remote jobs via a Web interface.

[2]VO: set of individuals or organizations that need to share resources to solve a given complex problem.

ISSN: 2279-8773

Programming tools and developments like the Design and implementation of scientific libraries suited to be efficiently used in heterogeneous and dynamic contexts represented a further higher level.

The top level was the one concerned with the development of applicative software. The project, in fact, took care also of the development of some demonstrators chosen among the applications of high interest not only for their intrinsic scientific value, but also as test-beds for high performance Grid platforms in the fields of Earth observation, Geophysics, Astronomy, Biology and Computational chemistry.

The research activities were organized in 14 Work Packages (WP) of which two in the area of High-speed Networks, five in the area of Middleware, two in the area of Programming Tools and Environment, five in the area of Applicative Demonstrators. As already mentioned one of these 5 Applicative Demostrators WPs (WP13) was devoted to molecular science. It was led by our laboratory and was named "Grid Applications For Molecular Virtual Reality".

The specific objective of WP13 was to promote the gridification of molecular computational applications in connection with the possible use in Molecular virtual reality to support theoretical and experimental investigation. The outcome of the activities carried out of CHEMGRID consists therefore of the logical scheme of the workflow of molecular simulator for beam experimnts (SIMBEX) and of an ensemble of suites of computer programs developed or implemented by the partner laboratories for that purpose. Each partner Laboratory was committed to develop and maintain (in addition to being responsible for) a set of software and computer codes according to the spirit of the Computational chemistry Metalaboratories of METACHEM.

The potentialities of grid techniques have been exploited in three areas of scientific knowledge relevance for the design of new drugs, of innovative materials, of computing devices and for the understanding of environmental fenomena and technological processes. These areas are, respectively, the gas phase, the condensed phase and large molecules. The molecular simulator has been in particular developed to simulate in *a priori* fashion molecular beams gas phase experiments.

For this purpose the following programs have been considered for implementation on the CHEMGRID segment of the Grid:

- **MOLPRO**: A program devoted to electronic structure calculations for isolated molecules;

- **GAMESS-UK**: A program devoted to electronic structure calculations for isolated molecules (English version);

ISSN: 2279-8773

- **GAMESS-US**: A program devoted to electronic structure calculations for isolated molecules (American version);
- **GAUSSIAN**: A program devoted to electronic structure calculations for isolated molecules;
- **ABC**: A program devoted to calculate single and multiple excitation energies for electrons using a time independent method;
- **RWAVEPR**: A program devoted to calculate reactive probabilities using a wavepacket approach;
- **VENUS**: A classical trajectory program devoted to calculate collisional properties of polyatomic systems;
- **CRYSTAL**: A program devoted to electronic structure calculations for solid state aggregates;
- **AMBER**: A program devoted to the assemblage of the force field of complex systems;
- **CPMD**: A car-Parrinello program devoted to the calculation of the dynamics of complex systems;
- **QM/MM**: A quantum mechanics molecular mechanics program devoted to the calculation of the dynamics of complex systems;
- **DL POLY**: A classical mechanics program for dynamical studies of molecular systems.

Some characteristics of the most popular of these programs are given in Tab. 2.3.

The model adopted for assembling the grid infrastructure and implementing on it the computer programs is described in detail in the reports of the project and was taken care by of the ISTM-CNR Laboratory of Perugia. This work has shown to be extremely heavy since there was no regular response from the various laboratories because of the high experimental nature of the project. A second aspect evidenced by the project has been the difficulty of keeping the standards aligned with the choices made by the ongoing European projects which have a wider spectrum of requirements though a slower pace in adopting innovative solutions. As a matter of fact this has created disalignments among the various laboratories whose activities had different response times with respect to the choices made by the production Grid environments.

ISSN: 2279-8773

More significant have been the achievements of the project in terms of designing workflows for molecular science applications. As a matter of fact the workflow designed for SIMBEX is articulated as:

1. Interaction: Information of the interaction was assumed to be already available as a LEPS fortran routine, whose parameters [97–99] were stored in a library. Other options were considered though not developed. The first case applies when the LEPS parameters have not been fitted though *ab initio* calculations are available on the web. For the first case a portal called FITTING was developed to carry out a best fit of the parameters of the adopted functional form to the calculate *ab initio* values. The second case applies when *ab initio* values are not available and need to be calculated. In this case the workflow has been structured so as to call the necessary procedures using SUPSIM (to perform *ab initio* calculations) and FITTING portals.

2. Dynamics: Dynamics calculations are performed using a package derived from the quantum program exchange library to carry out the integration of atom diatom classical trajectories. The program has been specialized to use a LEPS potential though it has built in the possibility of using various other functional representations of the interaction. Again, also this section of the workflow has been structured in a more general way. It is ready, in fact, to run both time dependent and time independent quantum reactive scattering programs developed in our laboratories. Options have also been inserted to include in future releases of SIMBEX larger systems molecular dynamics programs like DL_POLY

3. The virtual monitors. The activated virtual monitors of the demo SIMBEX version can manage internal energy distributions of the products (rotation and vibration) as well as translational energy and angular distributions. It can also create plots for non observable quantities like the "opacity function" useful for rationalizing the results. In WG13 it was possible also to develop through the interaction with other laboratories and workpackages more useful tools. Among them those shared with the "High performance component based programming environments", "Scientific libraries" and "Knowledge services for intensive data analysis, intelligent searching and intelligent query answering" workpackages for side activities of WP13.

ISSN: 2279–8773

2.7 The European Production Grid

The European Union has funded within the Framework Programs 6 and 7 (FP6 and FP7) the design and the construction of a Europe wide production Computing Grid. The project called EGEE (European Grid for E-sciencE) [1] has been first approved for two years and then renewed twice for additional two years each time (EGEE II and EGEE III respectively). As a result, at present, researchers in academia and industry already benefit from the EGEE e-Infrastructure that which simultaneously supports many applications from various scientific areas, providing a shared pool of resources, independent of geographic location, with round-the-clock access to major storage, compute and networking facilities. The EGEE project aims also at significantly extending and consolidating this infrastructure, that links national, regional and thematic Grid resources and interoperates as well as with other Grids around the globe. The resulting high capacity, world-wide infrastructure greatly surpasses the capabilities of local clusters and individual centres, providing a unique tool for collaborative computing in science ("e-Science"). So far, several large- and small-scale communities use the EGEE infrastructure as an every-day tool for their work. Applications deployed come from High Energy Physics, Life Sciences, Earth Sciences (including the industrial application EGEODE), Astrophysics, and Computational Chemistry. EGEE will expand the portfolio of supported applications to include Nuclear Fusion as well as other disciplines.

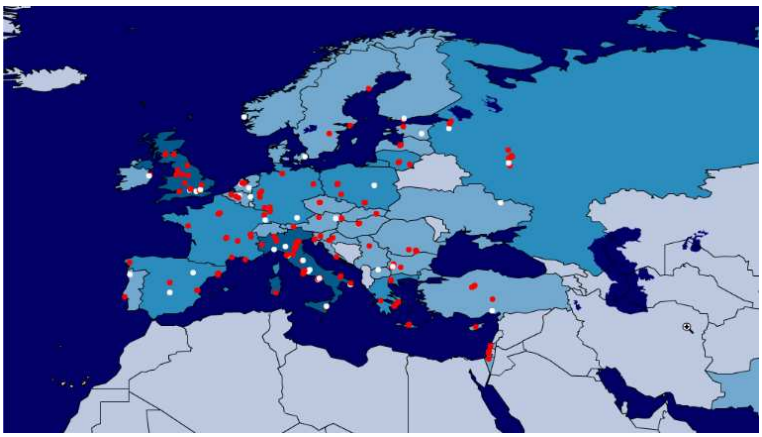

Figure 2.28: Partner countries of the EGEE project.

The EGEE has established a Consortium consisting of more than 90 partners from 32 countries, grouped into 13 federations and representing almost

ISSN: 2279–8773

all major and national Grid efforts in Europe, as well as projects from the US and Asia (see Fig. 2.28). In addition, a number of related projects will extend the infrastructure further, to the Mediterranean area, Baltic States, India, Latin America and China. Combined with other related projects spurred out from or affiliated with EGEE, EGEE-II, and the present EGEE III, this project has played around the world. With an expanded Consortium of enthusiastic participants and a large range of related projects, EGEE has also fostered the constitution of organized Virtual communities (called Virtual Organizations or VO) devoted to gather together in a structured way the members of a given scientific community. Thanks to all this EGEE will be able to further develop its infrastructure into a truly pervasive global platform for e-Science.

2.7.1 EGEE Activities

To illustrate the wealth of activities promoted by EGEE we give below a short description of some of them:

Project Activities : The main project activities of EGEE consist in supporting Networking, Service and Joint Research. In response to the more mature state of the Grid technology, the project has consolidated and developed even more Service and Networking activities. This will allow the project to have follow up in new countries, applications and sites joining the infrastructure as well as expand efforts in dissemination, training and application support. Efforts in software development have been concentrated on Grid foundation components and integrating the third party components coming from other projects and sources.

Networking Activities : The Networking Activities of EGEE include NA1 (Project Management); NA2 (Dissemination, Outreach and Communication); NA3 (User Training and Induction); NA4 (Application Identification and Support); and NA5 (Policy and International Cooperation).

Service Activities : The Service Activities consists of SA1 (European Grid Operations, Support and Management) and SA2 (Networking Support). A new activity, SA3 (Middleware Integration, Testing and Certification) combines software elements from a variety of sources to provide integrated releases for deployment on the infrastructure.

Joint Research Activities : JRA1 (Middleware Re-Engineering) will continue to develop and support the gLite middleware. JRA2 (Quality Assurance) will manage quality throughout the project, including overall security and coordination.

ISSN: 2279–8773

The project partners are given in Table 2.4.

A particular effort has been put in EGEE to develop Grid enabled applications. Actually EGEE from its very beginning has been built on the computing needs of the Large Hadron Collider (LHC) experiment of CERN (European Organization for Nuclear Research) near Geneva, Switzerland. In any event the High Energy Physics (HEP) and Biomedicine were integrant part of the original EGEE project. As EGEE has progressed into its second phase, other research domains were added and related Virtual Organizations were formed. A brief description of the application domains currently supported by EGEE is given below.

High Energy Physics (HEP) applications. The HEP community was one of the two pilot user domains for EGEE and remains a major user of the EGEE infrastructure, providing vital input that allows EGEE to ensure it provides a user-oriented service. The original EGEE HEP community was formed from the researchers of LHC. These experiments themselves are estimated to produce some 15 petabytes of data per year when the collider will enter its production phase. These data will be managed and processed using the EGEE infrastructure. Other international HEP experiments are also making use of the EGEE infrastructure, including the BaBar (the B and B-bar experiment), CDF (Collider Detector at Fermilab) and D experiments using particle accelerators in the USA, and the ZEUS and H1 experiments using the HERA collider at the DESY laboratory in Germany.

Biomedical applications. Applications in the biomedical field have been included in the EGEE project from the outset and are now exploiting the infrastructure in a sustained production mode. The biomedical community benefits from the Grid by enabling remote collaboration on shared datasets as well as carrying out high throughput calculations. The applications cover the fields of medical imaging, bioinformatics and drug discovery, with 23 individual applications deployed or being ported to the EGEE infrastructure. Notable among the biomedical sector applications is the WISDOM initiative, which has carried out a number of high profile drug discovery calculations. These verify the EGEE infrastructures ability to perform large, complex tasks and its usefulness as a tool in the fight against diseases such as malaria and avian flu.

Astro(-particle) Physics applications. The two major VOs in this domain, Planck and MAGIC, share problems of computation involving

ISSN: 2279-8773

large-scale data acquisition, simulation, data storage, and data retrieval. The Planck satellite of the European Space Agency (ESA) will be launched in 2008 and aims to map the microwave sky with an unprecedented combination of sky and frequency coverage, accuracy, stability and sensitivity. The MAGIC application simulates the behaviour of air showers in the atmosphere, originated by high energetic primary cosmic rays. These simulations are needed to analyze the data of the MAGIC telescope, located in the Canary Islands, to study the origin and the properties of high energy gamma rays.

Earth Science Research (ESR) applications. Earth Science covers a large range of topics related to the solid earth, atmosphere, ocean and their interfaces as well as planet atmospheres and cores. Recently, members of the ESR Virtual Organization have worked on rapid earthquake analysis, helping the scientific community to better understand these devastating natural disasters.

Geophysics applications. The Geophysics domain is closely related to the Earth Sciences domain and supports EGEODE (Expanding GEOsciences on DEmand), EGEEs first industrial application. EGEODE was initiated by the private company CCG (Compagnie Gnrale de Gophysique). It allows academic researchers to use the companys Geocluster software on the EGEE infrastructure.

Fusion applications. The capability of Grids for meeting the needs of the fusion community has been demonstrated. Several applications are already running on the EGEE infrastructure: massive ray tracing to estimate the trajectory of a microwave beam in plasma; kinetic transport and optimization of special magnetic confinement fusion devices (stellerators). Several computational tasks related to the ITER (International Thermonuclear Experimental Reactor) project were successfully ported to the EGEE infrastructure and will be further expanded with the main technical work expected to start in 2007.

Computational Chemistry applications. The main user in the field of computational chemistry is COMPCHEM, the VO managed by Perugia whose horserace is the set of programs being part of the *a priori* molecular simulator GEMS. Several Computational Chemistry applications have already been ported to the Grid and have been run in production to calculate observables for chemical reactions, to simulate the molecular dynamics of complex systems, and calculate the structure of molecules, molecular aggregates, liquids and solids.

ISSN: 2279–8773

Finance and Multimedia applications. There are also two minor application domains more recently started out with EGEE. The multimedia domain is currently in testing through EGEEs GILDA Grid test-bed. The financial applications involve work with the Abdus Salam International Center for Theoretical Physics.

However EGEE supports also a number of related European and national projects wishing to use the EGEE middleware or the EGEE infrastructure or both. DILIGENT develops Grid software for creating and maintaining digital libraries. DEGREE aims at promoting Grid technologies throughout the large and diverse Earth Sciences community. GRIDCC aims at integrating instrumentation with the Grid. BEinGrid fosters the adoption of Grid technologies by realizing several business experiments and creating a toolset repository of Grid middleware. For more information about the applications running on EGEE, visit the User and Application Portal at Ref. [100].

EGEE is fully committed to support the maximum range of research domains and applications and, as a result, it is always keen to attract others research laboratories to participate in its work. This might be at one of several levels, as an end-user, manager of a virtual organization, or as a resource provider. To participate as an end-user, the user must have a certificate from an accepted certificate authority (see the EUGridPMA at http://www.eugridpma.org/) and must join an existing virtual organization. To obtain a certificate the user has to contact the appropriate certificate authority, by looking at
http://www.eugridpma.org/members/worldmap/ or
http://www.eugridpma.org/members/ on the EUGridPMA site. The user should then search the list of existing virtual organizations (online at
https://cic.in2p3.fr/index.php?id=vo) to find the one suited for the user. Each entry contains a link for enrollment information. The enrollment process usually takes a couple of days for verifications. To start a new a Virtual Organization, the VO manager must fill in the VO Registration Form online at
https://cic.in2p3.fr/index.php?id=vo&subid=vo registration. The user must have a Grid certificate to access and submit this form. The approval process takes a minimum of three business days. After that the minimum VO services must be deployed. If the new VO is in the vo.eu-egee.org domain, then the EGEE project can deploy the necessary VO Membership Service. If the user chooses another domain, then he may have to deploy this service for his organization, although the Operations Advisory Group (OAG) will try to find a site willing to do so. To participate as a resource provider, the user should have a look to the EGEE SA1 information for resource centres (online

ISSN: 2279–8773

at http://cern.ch/egee-sa1/participate.html). One notes that each virtual organization is expected to integrate computational resources into the EGEE infrastructure generally equivalent to its average consumption, although this can be relaxed in exceptional circumstances. Members of business and industry are also encouraged to join the project, and several structures have been set up for these interactions. For more details one can see the information sheet on industrial involvement or one can visit the "EGEE and industry" section on the public website at www.eu-egee.org.

2.7.2 From EGEE to EGI

Given the success of the EGEE programme, it has become unavoidable to build on its achievements and prepare the transition towards a sustainable infrastructure. EGEE III project is already a step forward in the transition to the new European Grid Initiative (EGI). The design of EGI and of the procedure for implementing it has been the goal of the European Grid Initiative Design Project (EGI_DS), approved under FP7. EGEE-III will work closely with EGI_DS to transfer its broad experience in operating large-scale international Grid infrastructures. Also, EGEE-III will start implementing the required structural changes to allow a seamless transition to the EGI model, ensuring the continued provision of the Grid service. Essentially EGI will be a European coordination body managing the relationships with the National Grid Infrastructures (NGI). In Italy the NGI is at present being structured for formal recognition and will be named IGI (Italian Grid Initiative)

A second major goal of EGEE-III is to maintain and enhance the production quality computing infractructure to an increasing range of researchers in diverse scientific fields, thus strengthening Europe's leading position in high quality research. A key aspect of that is the establishment of the EGEE reference open-source middleware "gLite" as an open standard to be spread from research to public institutions and industries.

Initially EGEE used a middleware based on the outcomes of the European DataGrid (EDG) project, later developed into the LCG middleware stack. In parallel, EGEE has developed and re-engineered most of this middleware stack into a new middleware solution, gLite, now being deployed on the preproduction service. The gLite stack combines low level core middleware with a range of higher level services. Distributed under a business friendly open source license, gLite integrates components from the best of current middleware projects, such as Condor and the Globus Toolkit, as well as components developed for the LCG project. The product is a best-of-breed, low level middleware solution, compatible with schedulers such as PBS, Condor and LSF, built with interoperability in mind and providing foundation services that

ISSN: 2279–8773

facilitate the building of Grid applications from all fields.

At present several academic and industrial research centres are collaborating in the development of the software, organized in a number of different activities: Security, Resource Access (Computing and Storage Elements), Accounting, Data Management, Workload Management, Logging and Bookkeeping, Information and Monitoring, and Network Monitoring and Provisioning. Development and deployment are also supported by EGEEs extensive infrastructure (training infrastructure) programme. This provides support ranging from online documentation to live seminars and webcast tutorials. Training is also available on the dedicated GILDA dissemination testbed, featuring its own Certification Authority (CA), and allowing users and system administrators to test all aspects of deployment and use of gLite. A project for building a gLite consortium is under way.

The gLite Grid services follow a Service Oriented Architecture, meaning that it will be easy to connect the software to other Grid services, and also that it will facilitate compliance with upcoming Grid standards, for instance the Web Service Resource Framework (WSRF) from OASIS and the Open Grid Service Architecture (OGSA) from the Global Grid Forum. The gLite stack is envisaged as a modular system, allowing users to deploy different services according to their needs, rather than being forced to use the whole system. This is intended to allow each user to tailor the system to their individual situation. Building on experience from EDG and LCG middleware development, gLite adds new features in all areas of the software stack. In particular it features better security, better interfaces for data management and job submission, a refactored information system, and many other improvements that make gLite easy to use as well as effective. Already deployed on the various testing and pre-production Grids of the project, the roll out of gLite over the pre-production service is in progress.

2.8 COMPCHEM: the molecular science Virtual Organization

EGEE provides a variety of services to the scientists clustered into Virtual Organisations. The services range from training and user support, through to the software infrastructure necessary to access the resources. In the field of Computational Chemistry these services are provided by COMPCHEM VO.

COMPCHEM [2] has been assembled by a group of molecular and material sciences laboratories committed to implement their computer codes on

ISSN: 2279–8773

the section of the production EGEE Grid infrastructure available to the VO.

2.8.1 Structure of COMPCHEM

COMPCHEM Virtual Organization offers to its members clear advantages for carrying out their computational campaigns (especially when they are so complex to not be feasible using other computing platforms). Only in this way the laboratories will take the burden of carrying out the extra duties necessary to work within a collaborative environment. Therefore the entry level of the VO offers to the user the possibility of implementing a code at wish for personal use. This entry level membership situation has a limited validity and is targeted to check the laboratories on their real willingness to operate on a Grid platform. Already at this level, in fact, several competences necessary to restructure the code to run in a distributed way by exploiting the advantages of using a Grid platform need to be acquired. In return one gets the advantage of distributing the code on a much larger platform and an easier interaction with the codes of other users of the VO.

As sketched in Table 2.5 one becomes real member of COMPCHEM only after committing him/ herself to open the code implemented on the Grid to a shared use by the other members of the VO. This implies the validation of a stable version of the code and the assemblage of all the necessary GUIs for use by other researchers. It also implies software maintenance services and user support. It may also imply the commitment to confer to the Grid additional hardware (especially for those suites of codes which need special devices) after a negotiation with the Management Committee (MC) of the VO about the relevance of such a commitment to the strategic choices of the virtual organization. Obviously, the conferring of both software and hardware to COMPCHEM will take place gradually due to the time needed to validate the software and to gridify the machines. A member will likely devote to VO related activities other unshared resources (e.g. for development work). To become member of the VO and acquire the status of "COMPCHEM stakeholder" a user should place a specific application to the MC. While the user status has a limited time validity (after which a user may become either a paying customer and/or a paid supplier of services) the status of member has no time limit (though its terms could be periodically revised). The status of COMPCHEM member may imply further levels of involvement. The stakeholder, in fact, should take care of maintaining the software and the local segment of Grid hardware (a particular attention is needed for the conferring of software, either commercial or not, with special constraints like the payment of fees since in this case commercial, legal and financial aspects are better dealt centrally).

ISSN: 2279–8773

The members of the VO are requested to be proactive in providing either their own work or attract financial resources specifically for the development of the VO. As to contributing by providing their own work this may be under the form of participation to the management of the Grid, to the development of WMs, etc.. As to attracting financial resources VO members should elaborate joint applications for funding, research projects and even develop within the VO commercial services. However, the most important contribution to the sustainability of COMPCHEM that is requested to the stakeholders is a high dynamisms in research and in the transfer of its outcomes into innovation and developments (R&D). This means that, ideally, all members of the VO should excell in basic and applied research and that a proper reward for that has to be given in the VO.

2.8.2 COMPCHEM applications

As already mentioned the COMPCHEM main computational application is GEMS. For this purpose several programs have been ported to the Grid and have been run in production. Efforts are also underway to port additional applications to the EGEE infrastructure and to promote wider collaboration between the computational chemistry research groups.

The GEMS application is used to implement a simulation environment to study reaction dynamics of complex chemical systems. To this end one makes use of the programs already implemented in GRIDCHEM and ported on the production Grid of EGEE through the computer cluster GRID of the Department of Chemistry devoted to this pourpose. More specific efforts have been those addressed at implementing grid empowered versions of quantum reactive scattering codes dealing with atom-diatom systems which is also the task of the QDYN working group of the COST Action D37. A brief description of these programs follows:

- **ABCtraj** [101] calculates the observables of the atom-diatom reactions in gas phase. These events are generated using Monte Carlo techniques. The program is linked to a molecular virtual reality environment that shows the outcomes of the simulation in virtual monitors.

- **VENUS** [102] calculates the cross-sections and rate coefficients for elementary chemical reactions by simulating the collisions between atoms and molecules whose initial conditions are sampled using a Monte Carlo scheme. This application is a modified version of the VENUS96 program by W.L.Hase (QCPE-671). It calculates the trajectory for two reactants (atoms or molecules) by integrating the Hamilton equation

ISSN: 2279–8773

in cartesian coordinates. Before the collision the molecules are selected at discrete internal energy states and after the collision a quantization of the internal energy is also enforced on the product molecule. Initial positions and momenta are selected by using a Monte Carlo method. A parallelized version using MPI has been also implemented.

- **DL_POLY** [10] is a package of subroutines, programs and data files, designed to facilitate molecular dynamics simulations of macromolecules, polymers, ionic systems, solutions and other molecular systems on a distributed memory parallel computer. The package was written to support the UK project CCP5 by Bill Smith and Tim Forester under grants from the Engineering and Physical Sciences Research Council and is the property of the Science and Technology Facilities Council (STFC). Two forms of DL_POLY exist. DL_POLY_2 is the earlier version and is based on a replicated data parallelism. It is suitable for simulations of up to 30.000 atoms on up to 100 processors. DL_POLY_3 is a domain decomposition version, written by I.T. Todorov and W. Smith, and is designed for systems beyond the range of DL_POLY_2 - up to 10.000.000 atoms (and beyond) and 1000 processors.

- **RWAVEPR** [103] it integrates rigorously the three-dimensional time-dependent Schrödinger equation for a generic atom-diatom reaction by propagating wave packets. It calculates the scattering S matrix elements for given values of the vibrational quantum number, the rotational quantum number, the total angular quantum number, the quantum number for the projection of the total angular momentum on the atom-diatom vector, for a given the parity and for a given range of total energies. From the **S** matrix elements the state-to-state reaction probabilities are calculated. The centrifugal sudden approximation (i.e. to neglect the Coriolis coupling) can be also invoked.

- **COLUMBUS** [104] is a collection of programs for high-level ab initio molecular electronic structure calculations. The programs are designed primarily for extended multi-reference calculations on electronic ground and excited states of atoms and molecules.

- **GAMESS** [105] is a program for ab initio molecular quantum chemistry. Briefly, GAMESS can compute SCF wavefunctions ranging from RHF, ROHF, UHF, GVB, and MCSCF. Correlation corrections to these SCF wavefunctions include Configuration Interaction, second order perturbation Theory, and Coupled-Cluster approaches, as well as

ISSN: 2279–8773

the Density Functional Theory approximation. Geometry optimization, transition state searches, or reaction path following, vibrational frequencies with IR or Raman intensities and a variety of molecular properties, ranging from simple dipole moments to frequency dependent hyperpolarizabilities may be computed. Most computations can be performed using direct techniques, or in parallel on appropriate hardware. A detailed description of the program is available in the following paper: "General Atomic and Molecular Electronic Structure System" M.W. Schmidt, K.K. Baldridge, J.A. Boatz, S.T. Elbert, M.S. Gordon, J.H. Jensen, S. Koseki, N. Matsunaga, K.A. Nguyen, S.Su, T.L. Windus, M. Dupuis, J.A. Montgomery J. Comput. Chem., 14, 1347-1363(1993).

- **ABC** [4] is a program that uses a coupled-channel hyperspherical coordinate method to solve the Schrödinger equation for the motion of the three nuclei (A, B, and C) on a single Born-Oppenheimer potential energy surface. The coupled-channel method used involves a simultaneous expansion of the wavefunction in the Delves hyperspherical coordinates of all three chemical arrangements (A+BC, B+CA, C+AB). The quantum reactive scattering boundary conditions are applied exactly at the asymtotes potential, and the coupling between orbital and rotational angular momenta is also implemented correctly for each value of the total angular momentum quantum number.

- **MCTDH** [106] is a program implementing the MultiConfigurational Time-Dependent Hartree (MCTDH) method is nowadays considered as one of the most powerful tools for the quantum dynamics simulation of multidimensional systems. Unlike conventional wave packets methods, in the MCTDH approach the wave function is expressed on a basis of time-dependent functions, which evolve along with the system. The use of this time-dependent basis set turns up into a much smaller basis dimension and thus a greater computational efficiency with respect to standard wave packet approaches.

- **FLUSS** [107] The fluss code performes a modified Lanczos iterative diagonalisation of the thermal flux operator. The output of the code is a set of eigenvalues and eigenstates which can afterwards be used to calculate the thermal rate constant of a chemical reaction. A Krylov space is generated by recursive application of the thermal flux operator onto an initial wave function, typically a Gaussian-type wave packet located in the vicinity of the transition state. The matrix representation of the operator in the Krylov-type basis is diagonalized to obtain

ISSN: 2279-8773

eigenstates and eigenvalues.

- **SC-IVR** [108] Semiclassical (SC) initial value representation (IVR) methods are used to calculate the thermal rate coefficents for the gas-phase reaction (like H + H_2, N + N_2, O + O_3). This program use Cartesian coordinates in the full space to carry out the calculations; it does not invoke the conservation of total angular momentum J to reduce the problem to fewer degrees of freedom and solve the problem separately for each value of J, as is customary in quantum mechanical treatments. The behaviours of the SC-IVR methodology are: first we used the semiclassical coherent-state propagator of Herman Kluk (HK). Second, the Bolzmannized flux operator can be tuned continuosly between the traditional half-split and Kubo forms. Third, the normalization integral is express in terms of simple constrained partition functions.

ISSN: 2279–8773

Table 2.2: Metalaboratories partners.

Name	Coordinator	Partner countries	Goals
MURQM	P. Carsky (CZ)	Poland (1) Greece (2) Germany (2) United Kingdom (1)	develop the *ab initio* approaches to the calculation of the molecular electronic structures at CI level
DIRAC	K. Faegri (NO)	Norway (1) Sweden (1) Denmark (1) Netherland (1) Israel (1)	develop relativistic computational approaches using 4 component techniques
SIMBEX	O. Gervasi (I)	United Kingdom (2) Spain (2) Sweden (1) Poland (1) Germany (1) Hungary (2)	build a molecular simulator for molecular experiments based on accurate potential energy surface calculations and rigorous dynamical treatments
ELCHEM	A. Laganà (I)	Austria (1) Denmark (1) Germany (2) Greece (2) Italy (1) United Kingdom (2)	develop ubiquitous electronic learning and teaching technologies and virtual laboratories based on human and molecular virtual reality
ICAB	E. Rossi (I)	France (1) Hungary (1) Spain (2)	develop computational tools for linear scaling approaches to computational chemistry methods and mark-up languages
DYSTS	A. Aguilar (E)	France (1) Italy (2) Spain (1)	build accurate computational approaches to the dynamics and the spectroscopy of systems relevant to environment and applied Chemistry

ISSN: 2279-8773

Table 2.3: Characteristics of the programs implemented on CHEMGRID. QC stands for Quantum Chemistry, CD for Classical Dynamics and QD for Quantum Dynamics.

APPLICATION			MIN. REQUIREMENTS			LIBRARIES
Env.	Name	Seq./Par.	Mem	Disk	CPU	
QC	Gaussian03	S/P	64MB	256MB	90s	Atlas (incl.d in G03)
QC	MOLPRO	S	80MB	500MB	120s	IntelMKL6.1
QC	GAMESS-UK	S	80MB	500MB	120s	LAPACK/ BLAS
QC	GAMESS-US	S	80MB	500MB	120s	LAPACK/ BLAS
QC	ADC	S/P	10MB	500MB	600s	IntelMKL6.1/ SCALAPACK
CD	DL_POLY	S/P	10MB	10MB	3h FFT	BLAS
QD	RWAVEPR	S	0.1-1GB	1.5MB	1-6*dd* FFT	BLAS, LAPACK
CD	ABCtraj	P	1 MB	1 MB	120s	
CD	VENUS	P	1 MB	100MB	1 week	

ISSN: 2279-8773

Table 2.4: Project partners.

AGSC	TW	DFN	DE	JINR	RU
UNIZAR	ES	DKRZ	DE	JKU	AT
BME	HU	ELETTRA	IT	JSI	SI
CCLRC	UK	ENEA	IT	KFKI-RMKI	HU
CEA	FR	FhG/SCAI	DE	KIAM RAS	RU
CERN	CH	FOM	NL	KISTI	KR
CESGA	ES	FZJ	DE	KTH	SE
CESNET	CZ	FZK	DE	LIP	PT
CGG	FR	GARR	IT	MTA SZTAKI	HU
CIEMAT	ES	Glasgow	UK	MTW	IT
CKSC	KR	GRNET	GR	NIIF	HU
CNES	FR	GSI	DE	Oxford	UK
CNR-ITB	IT	HEALTHGRID	FR	PIC	ES
CNRS	FR	ICI	RO	PNPI	RU
CRSA	FR	UW	PO	PSNC	PO
CS SI	FR	IHEP	RU	RED.ES	ES
CSC	FI	IISAS	SK	RENCI	US
ETHZ (CSCS)	CH	IMPB RAS	RU	RRC KI	RU
CSIC	ES	Imperial	UK	RUG	NL
CYFRONET	PO	INFN	IT	SARA	NL
DANTE	UK	IPB	YU	SINP MSU	RU
DATAMAT	IT	IPP-BAS	BUL	SRCE	HR
DESY	DE	ITEP	RU	SWITCH	CH
TAU	IL	TCD	IE	TID	ES
TUBITAK-ULAKBIM	TR	UChicago	US	UCM	ES
UCY	CY	UEDIN	UK	UH-HIP	FI
UiB	NO	UIBK	AT	UKBH	DK
ULB	BE	UNICAL	IT	UNILE	IT
UNIMAN	UK	UNINA	IT	UWisc-Madison	US
UPV	ES	USC	US	UvA	NL
VR	SE				

ISSN: 2279-8773

Table 2.5: Levels of membership in COMPCHEM.

Membership level	Description
User	*Passive*: Run a program implemented by other members of the VO *Active*: Implement at least one program for personal use
SW provider	*Passive*: Implement at least one program for use by other members *Active*: Interactive management of the implemented program for cooperative usege
Grid deployer	*Passive*: Confer to the Grid infrastructure at least a small cluster of nodes *Active*: Operates above the minimal level as support for the Grid deployment and management
Stakeholder	Take part to the development and the management of the VO

ISSN: 2279-8773

Chapter 3

Grid empowered Molecular Dynamics Applications

3.1 Introduction

To the end of analyzing in some detail the impact of Grid computing on molecular and material science, technologies and computational applications we have considered three study cases. The three cases range from exact quantum reactive scattering calculations, for an atom diatom system aimed at singhing out feartures of the exact quantum calculations that can be spotted by crossed beam experiments, to the investigation of the structural and thermodynamics properties of some hydrocarbon (namely methane and propane) bulks using MD methods, to the end of working out technological implications for their usage in refrigeration, and to the implementation of the CHIMERE package, to carry out multiscale simulations of the production of secondary pollutants in Central Italy.

3.2 The Quantum study of the F+HD and $N+N_2$ reactions

As just mentioned the first class of calculations we have considered to the end of exploiting the increasing availability of computer power on Grid platforms for molecular science applications is that of quantum reactive scattering codes.

The particularly high request of memory associated with Quantum reactive scattering calculations enables them to exploit only some segments of the grid platform. The implementation on the Grid of quantum reactive scatter-

ISSN: 2279–8773

ing codes is, indeed, the main goal of QDYN, the working group of the COST Action D37 [109] coordinated by our laboratory and is one of the missions of the virtual organization COMPCHEM [2]. More specifically, QDYN as part of the COST Action D37 aims at fostering the development of computing Grid tools and computer codes specifically designed to advance theoretical approaches and related algorithmic formulations in the field of quantum molecular dynamics. At the same time COMPCHEM aims at implementing on the grid the largest variety of computational applications relevant to molecular science among which, at present, the quantum dynamics part consists of some atom-diatom reactive scattering codes. For some of these atom-diatom reactive scattering codes COMPCHEM has already implemented grid empowered versions and is trying to design and develop suitable distributed workflows [3] which are meant to be part of the more ambitious project of building a Grid empowered simulator of molecular systems. As part of this effort in our laboratory we have investigated the porting of some quantum reactive scattering legacy applications onto the grid infrastructure of EGEE.

In particular, the application chosen for porting is the quantum mechanical atom-diatom reactive scattering program called ABC [4] that carries out accurate calculations of the quantum S matrix elements to evaluate reaction probabilities as well as state-to-state integral and differential cross sections. The ABC code in addition to a significant memory request has a quite large CPU demand. Depending on the input parameters, in fact, one fixed energy execution of the ABC code can take about 10 hours on a single PC. Moreover, the ABC code is seldomly used for just one set of parameters. In a typical use case the ABC program must be executed several times for different sets of input parameter, consuming a large amount of hours of CPU time as typical of parameter study approaches of the Grid.

For this purpose it was chosen to use also gridification tools of higher level (this work has been carried out in collaboration with the Grid Application Support Centre (GASuC) of the MTA SZTAKI [110]). The higher level gridification work was carried out using the P-GRADE Grid Portal [5, 6]. PGRADE is an open source tool that provides intuitive graphical interfaces to develop the porting and does not require the modification of the original code.

P-GRADE is available on all the major Globus, LCG and gLite based production grids [90]. This makes the present study easy replicable by other communities for other computationally intensive applications and makes highly effective the efforts to creating user friendly grid applications which are beneficial to the computational chemistry community. This makes it easier for the other members of the COMPCHEM VO execute the code.

In this sub section we refer to the version of ABC implemented in COM-

ISSN: 2279–8773

PCHEM. The work carried out is concerned both with the bench study of the F + HD system and the new investigation on the N + N_2 reaction.

3.2.1 The ABC program

As already mentioned the ABC computer code is designed to carry out exact and approximate quantum dynamical calculations for reactive atom diatom sistems. Its theoretical background differs from the scheme given in the first section mainly for the use of a different set of coordinates. ABC is probably the most popular atom-diatom quantum reactive scattering code and, thanks also to some members of our laboratory, exibits interesting computational features.

The ABC program integrates the Schrödinger equation for an atom-diatom reactive scattering problem using the Delves hyperspherical coordinates [111] and a coupled channel method. The program makes use of a time independent technique to integrate the atom-diatom nuclear Schrödinger equation:

$$[\hat{H} - E]\psi = 0 \tag{3.1}$$

on a single potential energy surface (PES) according to a Born-Oppenheimer approach.

Delves hyperspherical coordinates

Delves hyperspherical coordinates, though referring to one definite atom-diatom geometry, are not particularly suited to describe the related asymptotic situation. In fact, as true for all the hypershperical coordinate systems, due to the fact of using an angle for the final diatom description (rather than the related arc which would tend instead to the internuclear distance), they do not have a decoupled formulation at large values of ρ. Therefore, at large values of ρ, it is preferable to perform a transformation into Jacobi coordinates.

The volume element and coordinate ranges of Jacobi and Delves coordinates are, for an arbitrary function F [112],

$$\int F \, d\mathbf{R}_\tau \, d\mathbf{r}_\tau = \int_0^\infty R_\tau^2 \, dR_\tau \int_0^\infty r_\tau^2 \, dr_\tau \int d\hat{R}_\tau \int F \, d\hat{r}_\tau \tag{3.2}$$

and

$$\int F \, d\mathbf{R}_\tau \, d\mathbf{r}_\tau = \frac{1}{4} \int_0^\infty \rho^5 \, d\rho \int_0^{\pi/2} \sin^2(2\theta_{D_\tau}) \, d\theta_{D_\tau} \int d\hat{R}_\tau \int F \, d\hat{r}_\tau. \tag{3.3}$$

ISSN: 2279–8773

Then, in the reference SF system, the following relationship can be used:

$$d\hat{R}_\tau d\hat{r}_\tau = \sin\vartheta_{R_\tau}\, d\vartheta_{R_\tau}\, d\varphi_{R_\tau}\, \sin\vartheta_{r_\tau}\, d\vartheta_{r_\tau}\, d\varphi_{r_\tau},$$

and in the BF_τ system

$$d\hat{R}_\tau d\hat{r}_\tau = d\alpha_\tau\, \sin\beta_\tau\, d\beta_\tau\, d\gamma_\tau\, \sin\Theta_\tau\, d\Theta_\tau,$$

so that both angular integrations cover the usual $(4\pi)^2$ sr.

The surface of the hypersphere is the five-dimensional surface defined by the angles $\theta_{D_\tau}, \Theta_\tau, \alpha, \beta$ and γ, though sometimes, improperly, the hypersphere is meant to be only the two-dimensional surface of the internal coordinate half-sphere defined by θ_{D_τ} and χ_i.

While ρ gives an idea of the overall size of the ABC system (and thus of its moving towards fragmetation), θ_{D_τ} and Θ_τ describe the shape of the triangle formed by the three particles. For these reasons, it is often very helpful to view things on the surface of the internal sphere as functions of θ_{D_τ} and Θ_τ at ρ fixed. Since θ_{D_τ} and Θ_τ cover the upper half of the surface of a sphere, we can make plots using the projection most commnly used in making maps of the Earth as viewed from the North pole, and that is the *stereographic projection*, in wich the X and the Y on the plot are defined by

$$X = \tan(\frac{1}{2}\theta_{D_\tau})\cos\Theta_\tau, \qquad Y = \tan(\frac{1}{2}\theta_{D_\tau})\sin\Theta_\tau. \tag{3.4}$$

Quantum dynamics approach

In eq. 3.1 ψ is the nuclear wavefunction (depending on nuclear coordinates only) and $\hat{H}$ is the related electronically adiabatic Hamiltonian of the nuclei. In the approach adopted by ABC ψ is expanded in terms of the hyperspherical arrangement channel τ basis functions $B^{JM}_{\tau\upsilon_\tau j_\tau\Omega_\tau}$. The basis functions $B^{JM}_{\tau\upsilon_\tau j_\tau\Omega_\tau}$ which are also labeled after J (the total angular momentum quantum number), M and Ω_τ (the space- and body- fixed projections of $\mathbf{J}$), υ_τ and j_τ the τ asymptotic vibrational and rotational quantum numbers and depend on both the three Euler α, β, γ and the Jacobi Θ_τ angle as well as on the internal Delves hyperspherical angle $\theta_{D\tau}$. In order to carry out the propagation of the solution from small to large hyperradius values (the asymptotic region we need to integrate the equations linking the second derivative of the matrix of the coefficients ($\mathbf{g}$) of the expansion of ψ (see eq. 3.5) to the overlap matrix $\mathbf{S}$ and the coupling matrix $\mathbf{U}$ as follows:

$$\frac{d^2\mathbf{g}}{d\rho^2} = \mathbf{S}^{-1}\mathbf{U}\mathbf{g}. \tag{3.5}$$

ISSN: 2279-8773

where the **S** matrix elements are formulated as:

$$S^{\tau v'_\tau j'_\tau \Omega'_\tau}_{\tau v_\tau j_\tau \Omega_\tau} = \langle B^{JM}_{\tau v_\tau j_\tau \Omega_\tau} | B^{JM}_{\tau v'_\tau j'_\tau \Omega'_\tau} \rangle \tag{3.6}$$

and the coupling **U** matrix elements are formulated as:

$$U^{\tau v'_\tau j'_\tau \Omega'_\tau}_{\tau v_\tau j_\tau \Omega_\tau} = \langle B^{JM}_{\tau v_\tau j_\tau \Omega_\tau} | \frac{2\mu}{\hbar^2}(\bar{H} - E) - \frac{1}{4\rho^2} | B^{JM}_{\tau v'_\tau j'_\tau \Omega'_\tau} \rangle, \tag{3.7}$$

in which μ is the reduced mass of the system and $\bar{H}$ is the part of the Hamiltonian not containing derivatives with respect to ρ.

To integrate the set of coupled differential equations given in eq. 3.5 the interval of ρ has been divided as usual into various sectors, and solve for local (sector) bound state functions by diagonalizing a Hamiltonian which describes related $\theta_{D\tau}$ -dependent motions using a carefully chosen reference potential. The coupled-channel equations are integrated starting from small values of ρ (by propagating the solution first within the sector and then chaining its value at the end of the sector with that at the beginning of the next one) to the asymptotes where the solutions are matched to product states.

3.2.2 The bench F + DH reaction

The F + HD bench reaction was investigated computationally by carrying out the calculation of its reactive properties in the threshold region using an extremely fine grid of energy so as to single out sharp resonances for the zero total angular momentum ($J = 0$ given as `jtot` in Table 3.1 in which the namelist of all input data is given) case. This study is of particular relevance for the characterization of the F + HD reaction because a possible resonance falling in this energy region might survive to the total angular momentum averaging and show up in the plot of the integral cross section as a function of energy for a comparison with the experiment [115]. In Table 3.1 the maximum value of ρ (`rmax`), the number of sectors (`mtr`), the initial value of total energy (`enrg`) and the energy increment (`dnrg`) chosen for the calculations are also given.

The atomic masses (`mass`) given in the first line of the Table can be integer (the program adds more significant figure automatically) and if the PES used for the calculations is not specified the default one (SW of [139]) is chosen. The parameter `jpar` is not used for the F + HD reaction, since it has a meaning only for the symmetric A + B_2 reactions. The helicity truncation parameter `kmax` is also not used when J=0 since it has a meaning only for $J > 0$. The value of `kmax` given in the table is the one necessary to calculate

ISSN: 2279–8773

Table 3.1: Input parameters for the test calculation on the F + HD(v=0,j=0) reaction

Parameter	Explanation
`mass` = 19,1,2	Masses of the three atoms in atomic mass units.
`jtot = 0`	Total angular momentum quantum number J.
`ipar = 1`	Triatomic parity eigenvalue P.
`jpar = 0`	Diatomic parity eigenvalue p.
`emax = 1.7`	Maximum internal energy in any channel (in eV).
`jmax = 15`	Maximum rotational quantum number of any channel.
`kmax = 4`	Helicity truncation parameter kmax.
`rmax = 12.0`	Maximum hyperradius ρ_{max} (in bohr).
`mtr = 150`	Number of log derivative propagation sectors.
`enrg = 0.233`	Initial scattering energy (in eV).
`dnrg = 0.001`	Scattering energy increment (in eV).
`nnrg = 48`	Total number of scattering energies.
`nout = 0`	Maximum value of v for which output is required.
`jout = 0`	Maximum value of j for which output is required.

converged integral and differential cross sections for the F + HD ($v = 0$, $j = 0$) reaction at collision energies slightly higher than those considered here [140]. Due to the high number of scattering energies (`nnrg`) involved in this calculation and to the exothermiticity of the reaction that leads to a high number of open channels in the HF + D and DF + H product arrangements, we obtain a rather long output file whose final results are summarized in Fig. 3.1 and 3.2. An important aspect extracted from the calculation is the evidence that there is a pronounced quantum mechanical resonance in the reaction whose energy is close to 0.254 eV (see the upper panel of Fig. 3.1 where state to state probabilities for the two possible products are plotted as a function of the total energy E). The Figure shows, in fact, that the resonance peak is quite high (more than 0.5) only for the $v = 0$ to $v' = 2$ transition of the process leading to HF (see left hand side panels of Fig. 3.1). On the contrary, no resonant peaks are shown by the state to state probabilities of the process leading to DF (right hand side panels of Fig. 3.1). The product rotational distribution associated to the resonant transition is remarkably broad as shown in the lower panel of Fig. 3.1. A more detailed discussion of the characteristics of these resonances is given in ref. [115].

ISSN: 2279-8773

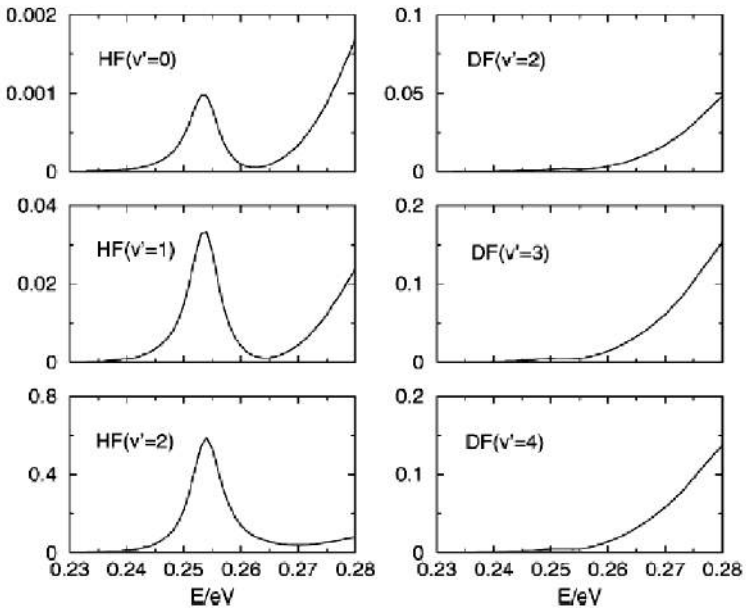

Figure 3.1: State to state reaction probabilities for F + HD($v = j = 0$) → HF(v') + D (left hand side panels) and DF(v') + H (right hand side panels) reactions calculated on the SW potential energy surface at $J = 0$.

3.2.3 The N + N_2 system

Previous theoretical investigations for the N + N_2 system

The accurate evaluation of the state to state cross section and rate coefficients of atom-diatom collisions is a highly demanding computational tasks since it requires full quantum calculations for a large number of initial states and a slowly converged sum of the fixed total angular momentum value J detailed S matrix elements. Such a tremendous computational effort is justified, however, only once the potential energy surface to be used for the calculations has already proven to be sufficiently accurate. Unfortunately this is not the case of the LEPS PES available from the literature and having a collinear barrier to reaction illustrated in Fig. 3.3. The LEPS PES has been used for the first dynamical calculations of the N + N_2 reaction [116] and for subsequent massive quasiclassical (QCT) and quantum infinite order sudden (RIOS) computational campaigns of the cross sections and rate coefficients [117–119] because of their relevance to the modelling of nitrogen plasmas and processes occurring around reentering spacecrafts [120]. More recently calculations on the same PES were extended to QCT vibrational relaxation and dissociation

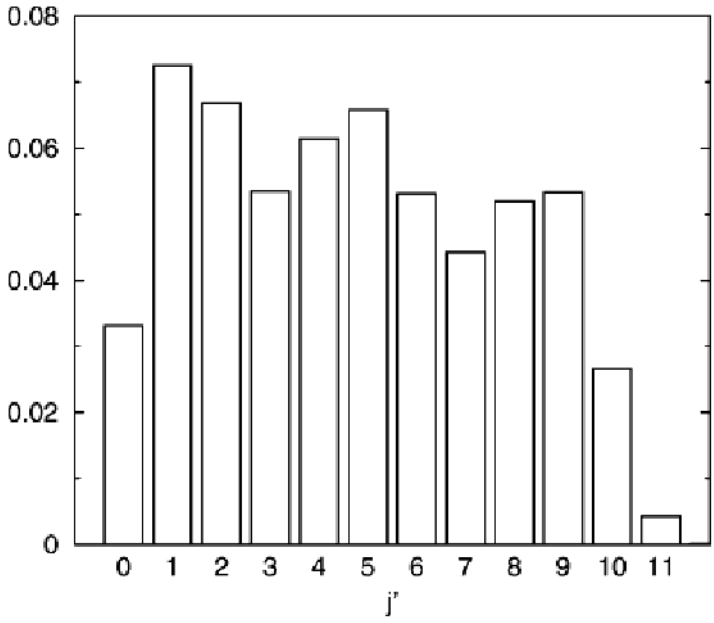

Figure 3.2: Product rotational distribution of the F + HD($\upsilon = j = 0$) $\rightarrow$ HF($\upsilon' = 2, j'$) + D reaction calculated at the energy E=0.254 eV of the maximum of the resonant peak at $J = 0$.

rates for the whole ladder of reactant vibrational states [121–123] and to semiclassical initial value representation thermal rate coefficients which were also compared to transition state and RIOS results [124].

The *ab initio* finding of a bent transition state reported in Refs. [125,126] proved the inadequacy of the LEPS PES in describing the main features of the interaction of the N + N_2 system in the region of the saddle to reaction, thus driving the attention of the researchers on the elaboration of a new and more accurate surface centered on a bent geometry (about 125°) of the transition state.

A time dependent quantal study of the reaction was then reported in ref. [127] on a high level *ab initio* calculation. The resulting potential energy surface (WSHDSP, from the initial of the authors) has a bent transition state with two almost equivalent N-N bonds sandwiched by two barriers connected by a shallow well. Unfortunately the WSHDSP PES is not available for distribution.

On the contrary, a PES available for distribution having a bent transition state (see Fig. 3.4) is the one based on the so called generalized rotating bond order (LAGROBO) functional [128–130] (LAG3) of Garcia and Laganà

described in ref. [131]. QCT and quantm RIOS calculations [131] performed on both the LEPS and the LAG3 PESs singled out the clearly different dynamical behaviour of the system on the two surfaces. However, to base the comparison of the dynamical properties of the LEPS and the LAG3 on a more (theoretically) robust ground, the authors carried out three dimensional (3D) time dependent quantum calculations [132, 133].

Experiments performed for the N + N_2 system

Experimental information about the N + N_2 interaction was derived in the early 1960s from the measurements of isotope-exchange thermal rate coefficients. In the temperature range falling between 300 and 1300 K the experiment was unable to detect isotope exchange between ^{14}N and $^{15}N_2$ [134]. The authors of ref. [134] indicated as an upper bound of 0.66×10^{-15} cm^3 $molecule^{-1}$ s for the rate constant of the global exchange process and suggested, as a resonable lower limit for the activation energy, a value falling in the range between 14 and 31 Kcal/mol (0.60 and 1.34 eV). Similar conclusions were reached also by Bauer and Tsang [135]. In addition, nitrogen molecules have been found to be quite stable in the presence of N atoms even when vibrationally excited [136].

A further upper limit to the rate constant at 1273 K was given by experimental results of Lyon in ref. [137], while an experimental esteem of the rate constant at 3400 K was given by Bar-Nun and Lifshitz in ref. [138] with a well defined error bar.

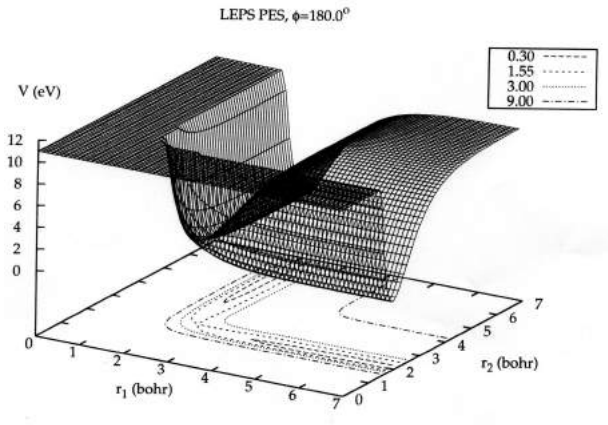

Figure 3.3: LEPS Potential Energy Surface plotted as a function of two internuclear distances at a fixed angle of 180.0 degrees for N + N_2 system

ISSN: 2279-8773

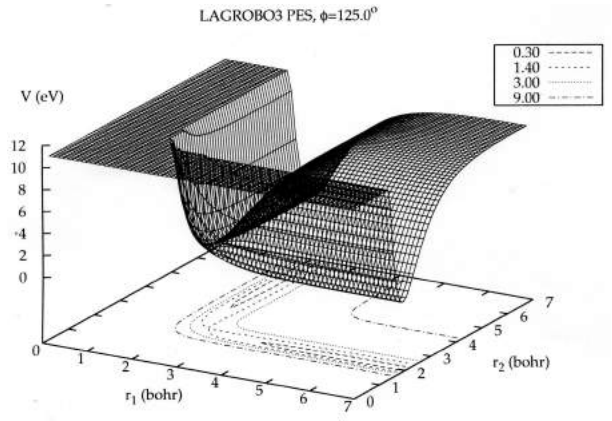

Figure 3.4: LAG3 Potential Energy Surface plotted as a function of two internuclear distances at a fixed angle of 125.0 degrees for N + N_2 system

3.2.4 Accurate *ab initio* N + N_2 calculations

A more rigorous quantum study of the N + N_2 system was carried out by calculating the detailed reactive probabilities of ^{14}N reacting with both $^{28}N_2$ and $^{30}N_2$ as specified by the reactions 3.8 and 3.9.

$$^{14}N(^4S) +^{28} N_2(^1\Sigma_g{}^+, \upsilon, j) \longrightarrow^{28} N_2(^1\Sigma_g{}^+, \upsilon', j') +^{14} N(^4S) \tag{3.8}$$

$$^{14}N(^4S) +^{30} N_2(^1\Sigma_g{}^+, \upsilon, j) \longrightarrow^{29} N_2(^1\Sigma_g{}^+, \upsilon', j') +^{15} N(^4S) \tag{3.9}$$

These calculations have been performed on both the LAG3 and the LAG4 PESs (two recently formulated PESs obtained by fitting a LAGROBO functional form to *a*b initio data) using the ABC program [141]. As summarized in table 3.2 the calculations were performed at zero total angular momentum (`jtot`) and diatomic parity (`jpar`) equal to 1 in order to include only odd reactant rotational levels. For LAG3 the investigated interval of total energy was varied from 1.2 eV (`enrg`), some tenths of eV below the barrier, up to 3.1 eV, about two times the energy of the barrier, using a grid (`nnrg`) of 153 total values having an energy step (`dnrg`) of 0.0125 eV. The maximum internal energy (`emax`) was set equal to 3.5 eV. To include all the asymptotically open channels a large maximum rotational quantum number (`jmax`) should have been considered. However, since the value of the computed probabilities does not show any significant change if channels with j larger than 75 are

neglected, we truncated accordingly the number of rotational functions used in the expansion. Convergence tests suggested to choose an upper value of the hyperradius (`rmax`) of 12.0 bohr. Due to the higher reaction barrier, for LAG4 the minimum value of the total energy was set at 2.0 eV (just below the barrier) and the maximum one at 4.1 eV (about two times the energy of the barrier), with a maximum internal energy set equal to 4.5 eV.

Table 3.2: Input parameters for the test calculation on the N + N_2 reactions using LAG3 and LAG4 PESs

Parameter	LAG3	LAG4
`mass =`	14.00674,15.00011,15.00011	14.00674,15.00011,15.00011
`jtot =`	0	0
`jpar =`	1	1
`emax =`	3.5	4.5
`jmax =`	75	75
`rmax =`	12.0	12.0
`mtr =`	150	150
`enrg =`	1.2	2.0
`dnrg =`	0.0125	0.0125
`nnrg =`	153	169

Zero total angular momentum and rotational quantum numbers product vibrational distributions (PVDs) were calculated at a relative kinetic energy of 1.65 eV and $\upsilon = 0$, 1, 2 for both reactions 3.8 and 3.9 on the LAG3 PES. The PVDs calculated on the LAG3 PES and normalized to the maximum are shown in the left hand side panels of Figure 3.5 as a function of the product vibrational quantum number υ'. The plots show that on LAG3 the reactive processes are mainly adiabatic with the exception of $\upsilon = 1$, for which a strong deexcitation to $\upsilon' = 0$ is observed (see Ref. [4] for its rationalization). No significant changes are seen in the product vibrational distribution of the isotopic variant. Similar calculations were performed on LAG4 at a relative kinetic energy of 2.40 eV. Relevant results are shown in the right hand side panels of Figure 3.5. The plots show that the reactive process for both isotopic variants is mostly adiabatic for $\upsilon = 0$ while it is strongly non adiabatic for $\upsilon = 1$ and 2, for which deexcitation to $\upsilon' = 0$ is the most likely transition. Moreover, a significant vibrational excitation of the products is also found. Calculations were repeated for both reactions 3.8 and 3.9 by selecting a value of the relative kinetic energy correponding to a peak and a slightly lower one. Related PVDs are shown in the right and

ISSN: 2279-8773

left hand side panels of Figure 3.6, respectively. In both cases no significant isotope effect was found. A more detailed discussion of these aspects of the resonance has been given in ref. [141].

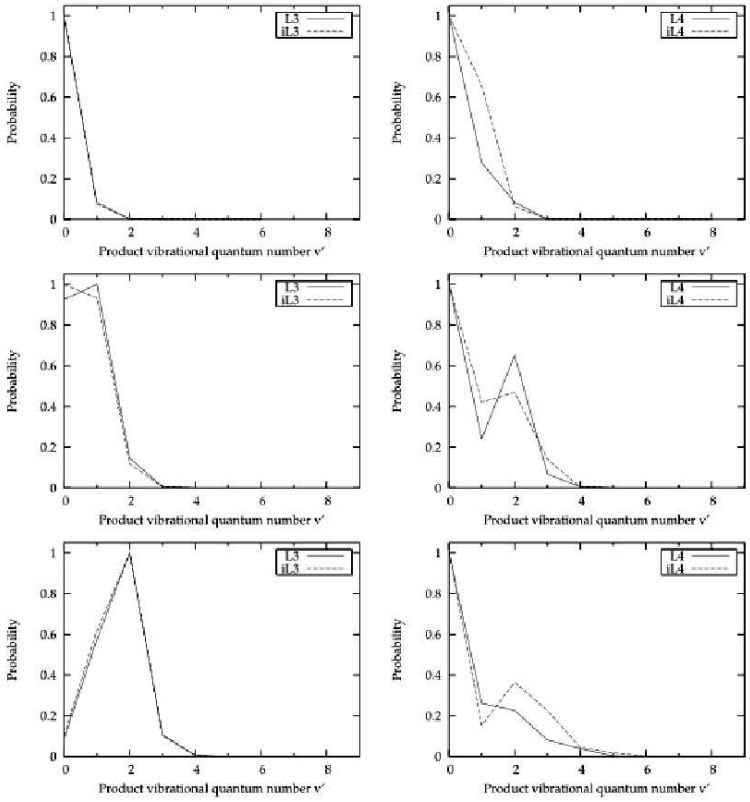

Figure 3.5: Product vibrational distributions calculated on the LAG3 PES at $E_{tr} = 1.65$ eV, $\upsilon = 0$ (upper panels), 1 (central panels), 2 (lower panels) and $j = J = 0$ plotted as a function of the product vibrational state υ' for reaction 1 (solid line) and 2 (dashed line) in the left hand side column. The corresponding plots calculated on the LAG4 PES at $E_{tr} = 2.40$ are given in the right hand side panels. Distributions are normalized to the maximum.

3.2.5 Gridification process of the ABC code

As already mentioned a prototype grid implementation of the code has been performed on the COMPCHEM User Interface (ui.grid.unipg.it) using the P-GRADE Grid Portal [5]. P-GRADE (release 2.7) provides graphical tools and services supporting grid application developers in porting legacy programs

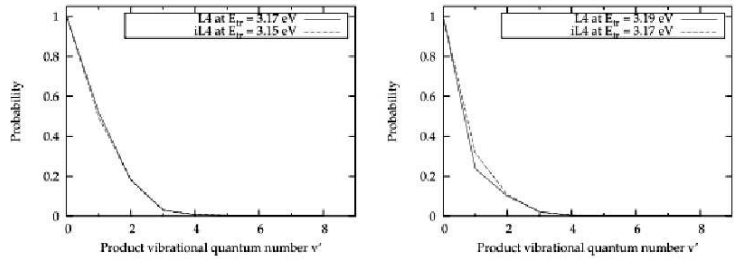

Figure 3.6: Product vibrational distributions calculated on the LAG4 PES at $v = 0$ and $j = J = 0$ for a non resonant (left hand side panel) and a resonant (right hand side panel) kinetic energy plotted as a function of the product vibrational state v' for reaction 1 (solid line) and 2 (dashed line). Distributions are normalized to the maximum.

onto grids without reengineering or modifying the code. It enables, in fact, application developers to define a parameter study application structure in a graphical environment and, based on this description, it generates the grid specific scripts and commands which actually carry out the execution on the distributed grid platform. The generic structure of P-GRADE Grid Portal application is a workflow. A workflow is a tool that integrates batch programs into a directed acyclic graph by connecting them together with file channels. A batch component can be any executable code which is binary compatible with the underlying grid resources (typically with Globus and EGEE clusters). A file channel defines directed data flows between two batch components and specifies that the output file of the source program must be used as the input file of the target program. The workflow manager subsystem of P-GRADE resolves such dependencies during the execution of the workflow by transferring and renaming files. A screen-shot of the P-GRADE Grid Portal implemented on COMPCHEM user interface and accessible to the user after the login procedure, is shown in Fig. 3.7 for four testing workflows.

The workflow developed for the ABC code with the help of the MTI-SZTAKI [110] team is shown in the left hand side panel of Fig. 3.8. As shown by the Figure the workflow is articulated in three different components made of large boxes called “Generator”, “Executor” and “Collector”. For the case study to be discussed here the convergence checks with the increase of the number of rotational skates considered and wiht the limit of propagation along the hyperradius are analyzed. In this study various copies of the ABC program needed to be executed with different “Maximum rotational quantum” and “Maximum hyperradius” values (see the right hand

ISSN: 2279-8773

side panel of Fig. 3.8) in a typical parameter study fashion for checking convergence. The role of the Generator is therefore that of producing all the necessary permutations of the jmax and rmax values and to store them in the input files. These files can then be used as input data during file stageing to grid resources and the Executor runs the sequential ABC code (implemented as a single Fortran 90 executable). In the left hand side of Figure 3.8 the small boxes placed by side to the components represent the input and output files which are used and produced by the Fotran code. These are prepared for the FORTRAN program by the workflow manager of the P-GRADE Portal and are then transferred to the EGEE Computing Element. This makes the executable need neither to know anything about the Grid configuration nor to be modified. The third component, the Collector, is responsible for collecting the results of the parameter study, analyzing them and creating a typical user friendly filtered result (to be considered as the final result of the simulation). In our case the Collector does not carry out any post-processing. It simply collects the results from the ABC jobs and compresses the files into a single archive file that can be downloaded to the user through the Portal web interface. The purpose of this step is, in fact, to make the access to results more convenient for the end users. After the definition of the workflow structure and the production of the executable and input components for the workflow nodes, the ABC code was run with multiple input data sets on the section of the production EGEE Grid infrastructure available to the COMPCHEM VO. Using the Workflow Manager window of P-GRADE (Fig. 3.9), the user is able to perform all the actions related to the chosen workflow (submission, abortion, resuming) and monitor the status of the job with the possibility of looking at the log file produced for every step when errors occur. From the same window the user is also able to directly download the results coming from the calculations by pressing the green button located under the “Output” field.

Performances

The gridified version of the ABC application executes in a sequence the input generator, the concurrent run of the ABC code on several grid resources, then the output collector. Since the execution time of both the generator and the collector stages are negligible compared with that of the ABC code, the latter is the dominant contribution to the overall execution duration. One execution of the ABC program on a single Intel Pentium 4 machine with 3.4 GHz CPU and 1 Gbyte memory takes from 3 to 6 hours, depending on the chosen values of jmax and rmax. On the other hand, using EGEE Grid Resources, every ABC job spends about as much time in the job queue of a

ISSN: 2279–8773

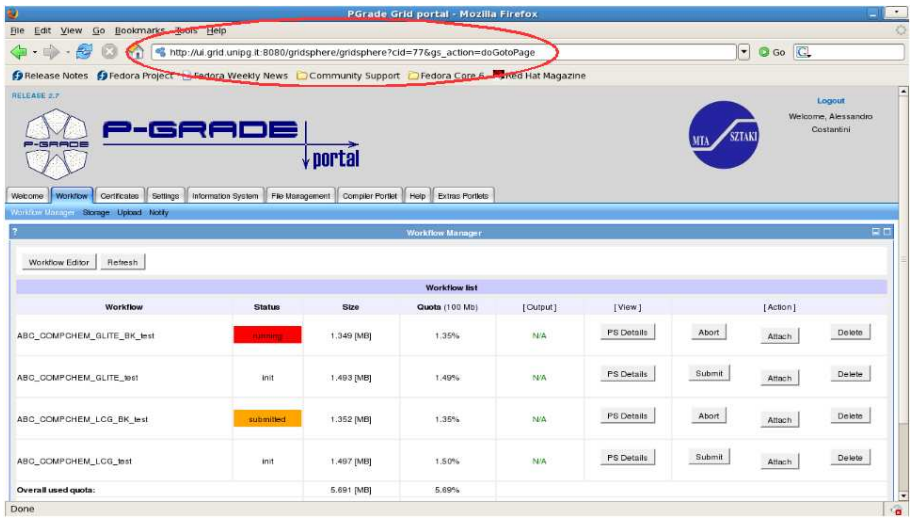

Figure 3.7: A screen-shot of the fully functional P-GRADE Grid Portal 2.7 installed on COMPCHEM user interface.

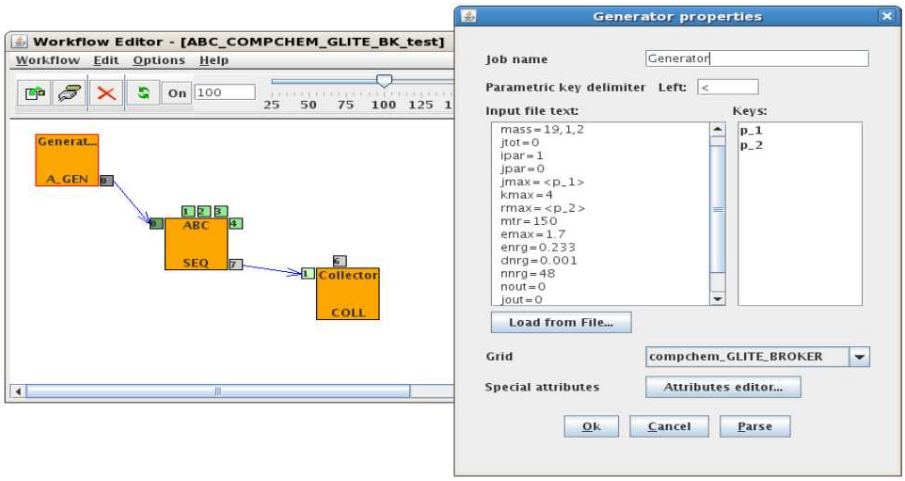

Figure 3.8: A screen-shot of the workflow prototype components generated by the automatic Generator of P-GRADE.

grid resource as on the CPU itself. This means that the average execution time of an ABC job on the grid is about twice as long as that on a dedicated local machine that is equivalent to say that the grid execution adds about 5 hours to each job. Accordingly, as soon as there are at least 4 ABC jobs in a simulation, the grid based execution is advantageous over the one on a single

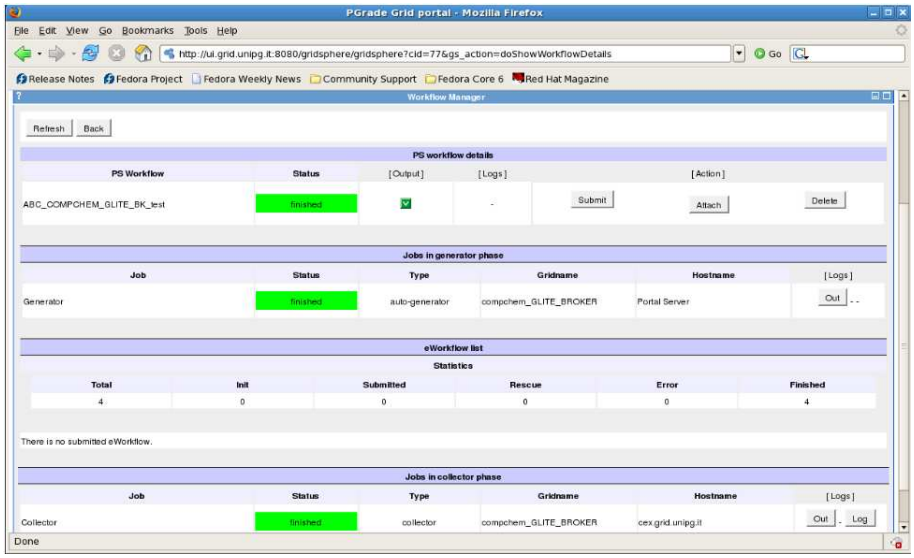

Figure 3.9: A screen-shot of the Workflow Manager window of the P-GRADE Grid Portal.

processor because 4 jobs running concurrently on 4 grid resources take an average of 10 hours. Meanwhile, the same simulation would take about 20 hours on a single processor local machine. The more parameter study jobs are executed within a single grid execution the lower elapsed time is obtained if compared with the one used by a local machine.

As a matter of fact in our bench runs we had between 2 to 4 times faster execution on the grid than on a local machine for a typical ABC parameter list (Table 3.1). A critical point instead is the choice of the DO LOOP on which the iteration is distributed. The choice made in our bench run does not fully exploit the potentialities of ABC for concurrent computing since it is clearly targeted to convergence studies (usually performed once for ever at the beginning of an investigation) and not to production runs and related massive computational campaigns. This means that it would be more appropriate to be able to intervene inside the code to exploit, for example, the concurrency at the energy iteration level. For this purpose a tool able to unpack the related DO LOOP would be truly useful.

P-GRADE empowered version: The ABC visualization tool

With the support of CESGA [142] some prototypical visualization tools (like a web portal able to analyze the ABC output files) have been implemented and are at present supported by on the P-GRADE system. For this pur-

pose a demo portlet prepared for Gridsphere (the same technology on which P-GRADE is based) was adapted in order to enable to draw 2D-Graph rendering of the Reaction Probabilities for a single output file and deploy it on P-GRADE. Portlets are pluggable user interface components which are managed and displayed on a web portal producing fragments of markup codes aggregated in a portal page. The implemented portlet has been written entirely in java and makes use of external bash scripts in order to:

- list the workflows present on the user's home
- unpack and list the packed file which contains all the output files for a single workflow
- extract all the needed data from the selected output file and process them in order to obtain a data format usable for the visualization.

The External Portlet is shown in both Fig. 3.10 and Fig 3.11. After a single workflow has been completed, the user can go to the "Externals Portlets", upload the workflow list with the related button and copy and paste the selected one on the first blank space before pressing the "Job list" button. After that the job list appears on the screen and the user needs to repeat the copy-and-paste procedure choosing a job output and put it on the second blank space. At this point the workflow and the related job have been selected and the 2D-Graph can be rendered. The final user can in this way compare the Reaction Probabilities for a selected atom-diatom reaction (with no need to download all the output files which remain on the server) and evaluate the possible strategies for a new calculation.

3.2.6 Conclusions

The study of the reactive properties of the F + HD and the N + N_2 reactions has given us the opportunity of analysing and performing the porting at the ABC application onto the EGEE grid environment and exploiting the collaboration with the Application Porting group of the EGEE project. Within this collaboration the P-GRADE portal was made to perform in a user friendly way the implementation of a distributed concurrent version of the code. This has allowed us to show that the competition for job slots on the EGEE grid can make the execution of a single ABC run twice slower than that on a local machine, the overall execution time of a simulation is far shorter than any local CPU run. This has been exploited to reexamine the case of the location of the low collision energy resonance of the F + HD system and to carry out an extended computational campaign of the reactive

ISSN: 2279–8773

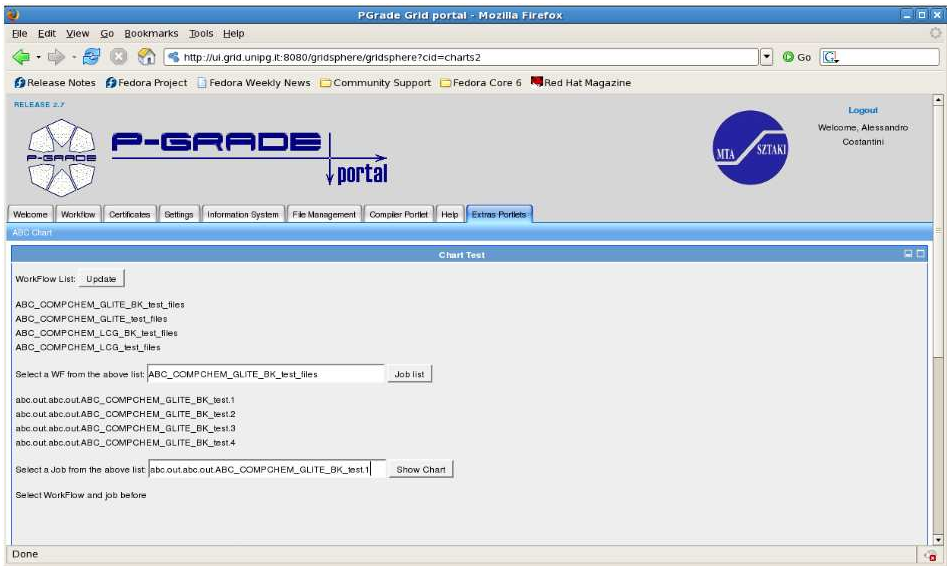

Figure 3.10: A screen-shot of the new Extra Portlet window of the P-GRADE Grid Portal in which the user can select the workflow from a workflow-list and a job from a job-list.

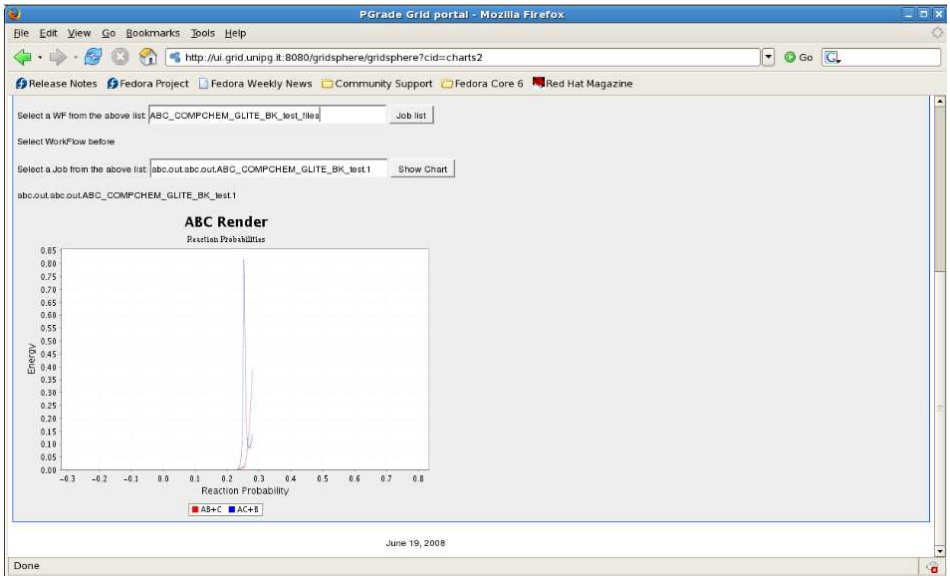

Figure 3.11: A screen-shot of the new Extra Portlet window of the P-GRADE Grid Portal in which the Reaction Probabilities 2D-Graph is rendered.

ISSN: 2279–8773

probability values needed to evaluate the thermal ratio coefficient of the N + N_2 reaction on different potential energy surfaces. As most of the computationally intensive atom-diatom quantum dynamics simulations fall into this category, the same approach used for the ABC code can be easily extended to other quantum reactive scattering codes.

ISSN: 2279-8773

ISSN: 2279-8773

3.3 The MD study of some Hydrocarbon Systems

Propane and methane are hydrocarbons commonly used as fuels. More recently, due to the Copenhagen amendments to the Montreal protocol [7] that has banned the use of CFCs and HCFCs before the year 2015, methane and propane are also being considered as long-term alternative refrigerants with no impact on global enviroment and in particular on stratospheric ozone depletion and global warming.

The possibility of carrying out massive computational campaigns on the EGEE production grid has motivated us to carry out extended calculations of the dynamics and thermodynamics properties of liquid propane and methane in order to perform a comparative analysis of some propane and methane macroscopic properties at some temperatures of experimental interest [8, 9] by adopting two different formulations of the force field to render theri interaction. The calculations were carried out using the DL_POLY [10] software package that is known for being native parallel, for having limited request of memory and offering room for several options in dealing with molecular process as already mentioned in Chapter 1. As will be discussed in detail in the following the gridification of DL_POLY has been carried out using the standard (and non-standard) tools of the EGEE environment and analysing in detail the performances achieved.

3.3.1 The formulations of the force field

The construction of the interaction represents, indeed, the most difficult task of any molecular dynamics simulation. The problem of giving a proper representation of the interaction of large systems is usually tackled using a force field which estimates the interaction by calculating the force acting on every particle of a given chemical system in an approximative way that take into account information coming from both experiments and theoretical calculations. In the present study use has been made of two different formulations of the force field named respectively

- OPLS/AMBER all-atoms potential [143]
- Dreiding potential [144]

The OPLS/AMBER force field

In the OPLS/AMBER [143] formulation of the force field the total potential energy V is expressed as a sum of V_{OPLS} (a model potential describing

ISSN: 2279–8773

the intermolecular non-bonded component of the interaction optimized for the simulation of liquids), V_{bond} (a model potential describing the stretching component of the interaction for the kth pair of bonded atoms), V_{bend} (a model potential describing the bending component of the interaction for the planar angle θ_l defined by the lth triplet of bonded atoms) and V_{tors} (a model potential describing the torsion component of the interaction for the dihedral angle ϕ_m defined by the mth quartet of bonded atoms) terms as follows:

$$V = V_{OPLS} + V_{bond} + V_{bend} + V_{tors} \tag{3.10}$$

V_{OPLS} is formulated as a sum of the Coulomb plus Lennard-Jones pairs of functionals as follows:

$$V_{OPLS} = \sum_{i}^{mol.a} \sum_{j}^{mol.b} \left[\frac{q_i q_j e^2}{r_{ij}} + 4\epsilon_{ij} \left(\frac{\sigma_{ij}^{12}}{r_{ij}^{12}} - \frac{\sigma_{ij}^{6}}{r_{ij}^{6}} \right) \right] f_{ij} \tag{3.11}$$

where q_i and q_j are the charges of atoms i and j respectively, e is the charge of the electron, ϵ_{ij}, σ_{ij} and r_{ij} are the depth, the equilibrium and the distance, respectively, between atoms i and j. In eq. 3.11 f_{ij} is always set equal to 1 for all ij pairs. Equation 3.11 is also used for intramolecular non-bonded interactions between all pairs of atoms separated by at least three bonds. The values adopted for the parameters of the non-bonded intermolecular interactions are those of the well known Jorgensen set of parameters that is widely used in Molecular Dynamics when simulating solvated or not solvated organic molecules. For our work these values have been modified with respect to those published in ref. [143] (see Table 3.3), after running extensive simulations of methane and propane bulks to the end of reproducing the related experimental values of volume and density. Standard combination rules specific of the OPLS model [143] (like the $\sigma_{ij} = (\sigma_i \sigma_j)^{1/2}$ and the $\epsilon_{ij} = (\epsilon_i \epsilon_j)^{1/2}$ ones) are adopted. The stretching (eq. 3.12), bending (eq. 3.13) and torsional (eq. 3.14) components of the intramolecular interactions are formulated as

$$V_{bond} = \sum_{k} K_k \left(r_k - r_{keq} \right)^2 \tag{3.12}$$

where K_k is the stretching force constant and r_{keq} is the equilibrium value of the bond for the atomic pair k;

$$V_{bend} = \sum_{l} K_l \left(\theta_l - \theta_{leq} \right)^2 \tag{3.13}$$

where K_l is the bending force constant and θ_{leq} is the equilibrium value of

Table 3.3: Values of the parameters of the nonbonded component of OPLS/AMBER force field for the propane and methane system

propane			
Atom	q/e^-	$\sigma/\dot{A}$	$\epsilon/kcalmol^{-1}$
$C-CH_3$	-0.180	3.620	0.064
C, R_2CH_2	-0.120	3.620	0.064
H	0.060	2.550	0.029
methane			
Atom	q/e^-	$\sigma/\dot{A}$	$\epsilon/kcalmol^{-1}$
C, RCH_3	-0.240	3.740	0.061
H	0.060	2.670	0.028

the angle bending for the atomic triplet l and

$$V_{tors} = \sum_m \frac{V_1^m}{2}\left[1+\cos\left(\phi_m\right)\right] + \frac{V_2^m}{2}\left[1+\cos\left(2\phi_m\right)\right] + \frac{V_3^m}{2}\left[1+\cos\left(3\phi_m\right)\right] \tag{3.14}$$

where ϕ_m is the dihedral angle and V_1, V_2 and V_3 are the coefficients of the Fourier series of the expansion of the interaction atomic quartet m. Values of the parameters of the stretching, bending and torsional components are given in Table 3.4 (as obvious, the torsional term does not apply to methane).

The Dreiding force field

In the Dreiding formulation [144] of the force field the total potential energy V is expressed again, as in the OPLS/AMBER one, as a sum of four components. They are namely:

$$V = V_{non-bonded} + V_{bond} + V_{bend} + V_{tors} \tag{3.15}$$

The van der Waals non-bonded intermolecular component is given as a sum of Lennard-Jones (12-6) model potentials of the following form:

$$V_{non-bonded} = \sum_i^{mol.a}\sum_j^{mol.b}\left[\left(\frac{A_{ij}}{{r_{ij}}^{12}}\right) - \left(\frac{B_{ij}}{{r_{ij}}^{6}}\right)\right] \tag{3.16}$$

where A_{ij} and B_{ij} are the two van der Waals parameters of the i j atom pair. The values adopted for parameters of the non-bonded intermolecular

ISSN: 2279–8773

Table 3.4: Values of the parameters of the stretching, bending and torsion components of the OPLS/AMBER force field for propane and methane (the torsion parameters do not apply to CH_4) systems

Stretching		
Type	r_{eq} / ∘	K_r / $kcal\ mol^{-1}$
$C-H$	1.090	662
$C-C$	1.526	620

Bending		
Group	θ_{eq} / ∘	K_θ / $kcal\ mol^{-1}$
$H-C-H$	109.5	70.00
$H-C-C$	109.5	70.00
$C-C-C$	109.5	80.00

Torsion			
Group	V_1 /$kcal\ mol^{-1}$	V_2 /$kcal\ mol^{-1}$	V_3 /$kcal\ mol^{-1}$
$H-C-C-H$	0.000	0.000	0.318
$H-C-C-C$	0.000	0.000	0.366

component are given in Table 3.5. They have been modified, with respect to those published in the literature [144] in order to reproduce the experimental volume and density of the two systems. The formulation of the intramolecular interaction component for the stretching is the same as the one given in eq. 3.12. For the bending (eq. 3.17) and torsional (eq. 3.18) components of the intermolecular interaction the following formulations have been adopted:

$$V_{bend} = \sum_l A_l \left(\cos\theta_l - \cos\theta_{leq}\right)^2 \tag{3.17}$$

and

$$V_{tors} = \sum_m A_m \left[1 + \cos\left(n_m\phi_m - \delta_0\right)\right] \tag{3.18}$$

where A_l and A_m are the bending and torsional force constants, δ_0 is the value of ϕ_m at which the torsional interaction has a minimum, and n_m is the

ISSN: 2279–8773

multiplicity (i.e. the number of minima as the bond undergoes a full rotation of 360°). Values of the parameters of the stretching, bending and torsional components are given in Table 3.6.

Table 3.5: Values of the parameters of the nonbonded component of the Dreiding force field for the propane and methane system

propane		
Atom	A $/kcalmol^{-1}\dot{A}$	B $/kcalmol^{-1}\dot{A}$
$C-C$	1042.494E3	740.5926
$C-H$	134.0361E3	167.98674
$H-H$	15306.78	35.89398
methane		
Atom	A $/kcalmol^{-1}\dot{A}$	B $/kcalmol^{-1}\dot{A}$
$C-C$	1112.774E3	700.8924
$C-H$	143.0723E3	158.9063
$H-H$	16338.70	33.95377

Isoenergetic countours of the associated potentials for the investigated systems

To illustrate the key features of the two force field adopted in our work the isoenergetic contours of the potential associated with a system of two molecules of the same hydrocarbon have been plotted in Fig. 3.12 and in Fig. 3.13.

For illustrative purposes we examine the propane case. To draw the isoenergetic contours both the origin of the reference system was set on the center of mass of one propane molecule (say molecule A), the xy plane was chosen to coincide[1] with the one of its three carbon atoms and the y axis was made solid with its central carbon atom (see Fig. 3.14). The isoenergetic contour plots refer to the case in which (for the sake of reducing the dimensionality of the plot) both molecules are frozen but the second molecule (molecule B) is allowed to rotate of an angle ϕ around the line connecting the center of mass of the two propane molecules R_{cm} (which lie both on the xy plane) and of an angle θ around the center of mass of A. The value of the potential taken for

[1]strictly speaking the two planes may not coincide and be only parallel and slightly

ISSN: 2279–8773

Table 3.6: Values of the parameters of the stretching, bending and torsion components of the Dreiding force field for the propane and methane systems

Stretching		
Type	r_{eq} / ∘	K_r / $kcal\ mol^{-1}$
$C-H$	1.090	700
$C-C$	1.530	700

Bending		
Group	θ_{eq} / ∘	K_θ / $kcal\ mol^{-1}$
$H-C-H$	109.47	112.49
$H-C-C$	109.47	112.49
$C-C-C$	109.47	112.49

Torsion			
Group	K_{tors} /$kcal\ mol^{-1}$	δ_0 ∘	n
$H-C-C-H$	0.111	0.000	3
$H-C-C-C$	0.111	0.000	3

the isoenergetic countours is the R_{cm} relaxed (i.e. its minimum along R_{cm} shown in the top panels of Fig. 3.12 left hand side for OPLS/AMBER; right hand side for Dreiding). The representation given in Fig. 3.12 is of the polar type with θ being the radius of the plot and ϕ the angle around its center (0,0). As shown by the two isoenergetic plots, there is a shallow minimum energy path that indicates the system can evolve following two alternative approaches in which the orientation of molecule B has opposite simmetric perpendicular positions with respect to molecule A. At the same time the Dreiding force field (right hand side) shows a more attractive long-range potential and deeper wells. The plots of the values of R_{cm} associated to the potential energy minimum for the two force fields show a similar shape indicating that the most energetically favoured part of the process occurs when the planes of molecule B is perpendicular to that of molecule A.

Corresponding easier to read cartesian plots are given in Fig. 3.13 for the same system by putting the angle of approach θ on the x axis and the angle of rotation ϕ on the y axis. As apparent from the figure the two dimensional (2D) contours of the model for OPLS force field (left hand side) single out

ISSN: 2279–8773

two deep wells associated with the most stable (with the second molecule (molecule B) perpendicular with respect to molecule A) adduct geometry and others two deep wells respectively for $\theta = 0$ and $\theta = \pi$ as reported in Table 3.7. At the same time the Dreiding force field (right hand side) shows a more attractive nature of long-range and a more repulsive one at short range. As for the polar type graph the plots of the values of R_{cm} shown a similar shape for the two models. The plots show also an insertion channel at $\theta = \pi/2$

In order to better single out the differences between the two formulations of the interaction for the propane molecule, the Lennard-Jones potential for the C-C interaction has been plotted in Fig. 3.15. As shown by the figure the Dreiding force field has deeper wells and a greater interaction sphere with respect to that of the OPLS one due to the more attractive nature of the long-range interaction.

More quantitative information on the stationary points of the OPLS term of the potential is given in Table 3.7.

Table 3.7: Characteristics of the main stationary points of the 2D cross section of the OPLS force field (upper panel) and Dreiding force field (lower panel) of the potential taken at the minimum along R_{cm}

E/$Kcal\ mol^{-1}$	$R_{cm}/\dot{A}$	θ/deg	ϕ/deg
-1.51	4.53	0	0/180
-1.46	4.36	90	90/270
-1.47	4.45	180	90/270
-0.78	5.47	50	0/180
-1.37	4.55	180	0/180
E/$Kcal\ mol^{-1}$	$R_{cm}/\dot{A}$	θ/deg	ϕ/deg
-1.75	4.54	0	0/180
-1.66	4.40	90	90/270
-1.02	5.50	180	90/270
-1.52	4.67	50	0/180
-1.75	4.54	180	0/180

3.3.2 Calculations and results

Several extended molecular dynamics calculations have been carried out using both the OPLS/AMBER and the Dreiding formulations of the force field.

ISSN: 2279–8773

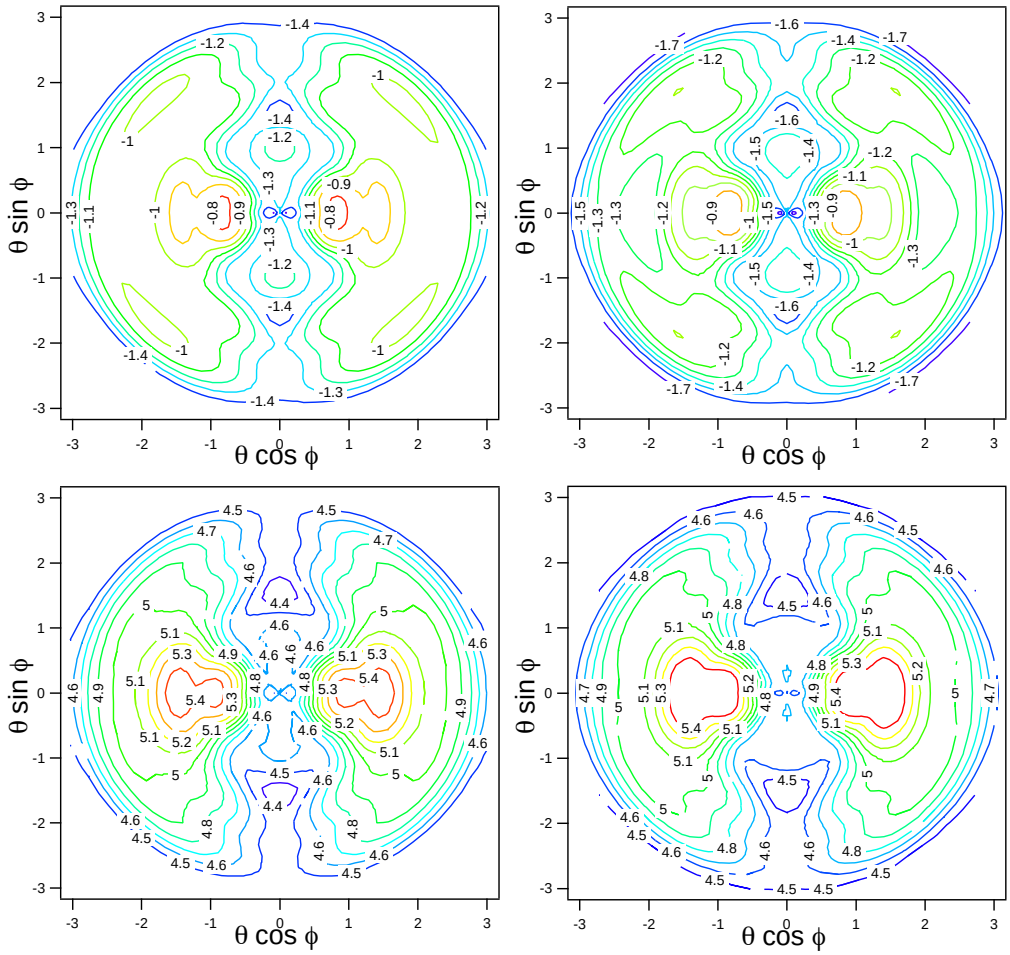

Figure 3.12: Polar isometric contours of the OPLS (left hand side panel) and of the Dreiding (right hand side panel) terms of the potential of two propane system given in $Kcal\ mol^{-1}$ and taken at the minimum along R_{cm} plotted as function of $\theta \sin\phi$ and $\theta \cos\phi$. Lower panels: the corresponding isometric contours of R_{cm} in $\dot{A}$.

All the calculations were performed using version 2.15 of DL_POLY (after implementing it on the EGEE-Grid production platform) and an *npt* statistical ensemble of 288 molecules for a time duration of 1 *ns* for each system. In order to mimic the experimental conditions of the system near the liquid-

ISSN: 2279–8773

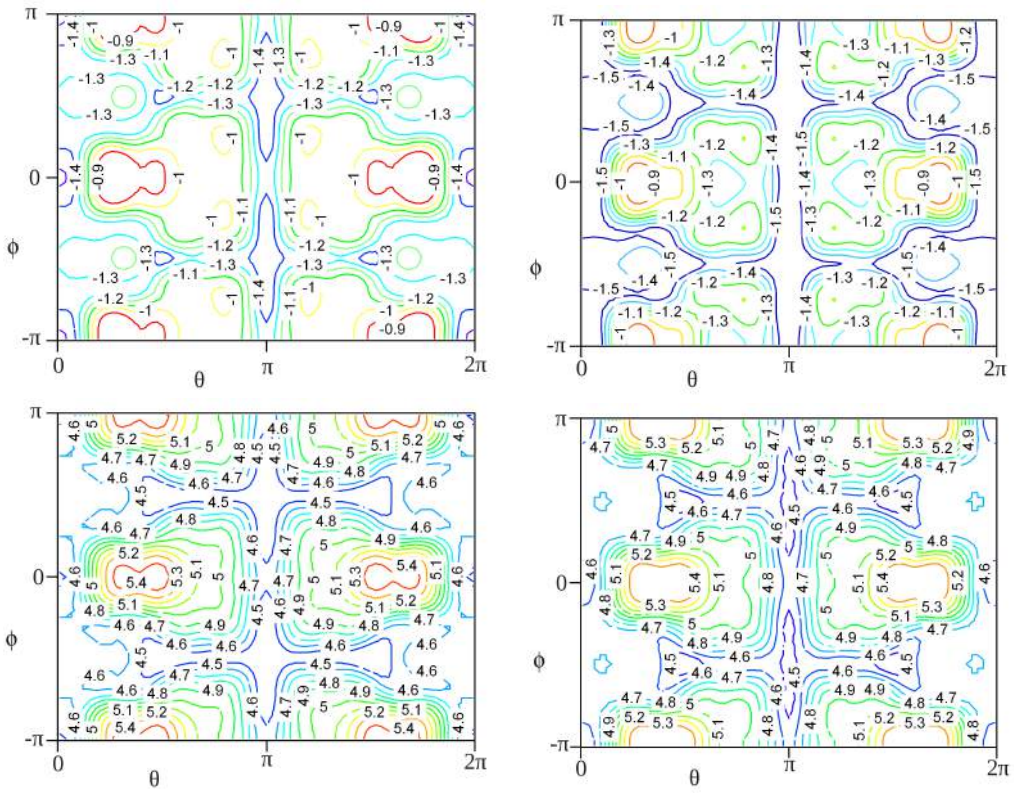

Figure 3.13: Cartesian isometric contours of the OPLS (left hand side panel) and of the Dreiding (right hand side panel) terms of the potential of two propane systemgiven in $Kcal\ mol^{-1}$ and taken at the minimum along R_{cm} plotted as function of ϕ and θ. Lower panels: the corresponding isometric contours of R_{cm} in $\dot{A}$.

gas equilibrium point, the calculations for the propane bulk were performed at a temperature T of 230 K and a pressure P of 1.013 bar [145, 146] while those for the methane bulk were performed at a temperature T of 110 K and a pressure P of 1.013 bar [147]. In all the calculations carried out so far, the computed values of the pressure of the system fluctuate as typical of the npt statistical ensemble. Calculated values of some observable properties of

ISSN: 2279-8773

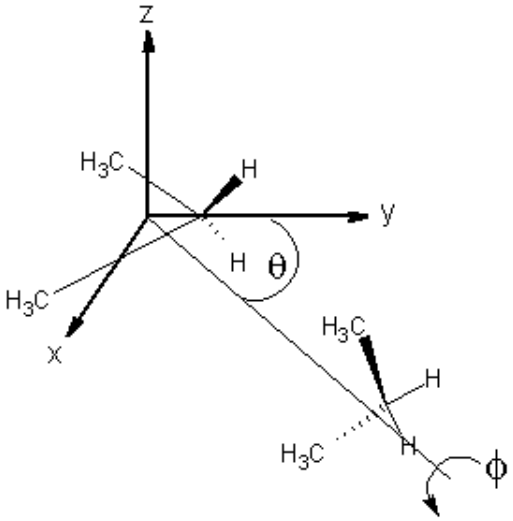

Figure 3.14: A sketch of the propane-propane adduct geometry

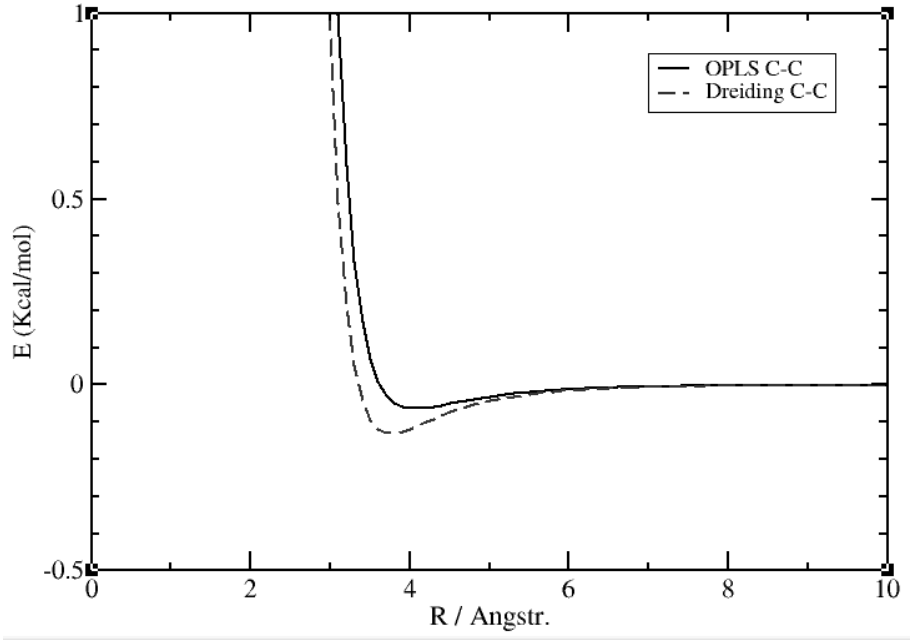

Figure 3.15: Plot of the C-C interaction for OPLS (solid line) and Dreiding (dashed line).

ISSN: 2279-8773

the bulk propane and methane systems are compared with the corresponding experimental information in Table 3.8.

As shown by the table, the values calculated using the two force fields almost coincide and both satisfactorily reproduce experimental data [146, 147]. In order to obtain more detailed information on the nature of the solvation processes of a hydrocarbon molecule by an hydrocarbon bulk we calculated the C-C, C-H and H-H pair distribution functions $g(r)$ which are be defined as follows [148]:

$$g(r) = \frac{1}{N\rho}\langle\sum_{i\neq j}\delta\left(\mathbf{r}+\mathbf{r}_i-\mathbf{r}_j\right)\rangle \tag{3.19}$$

where $\mathbf{r}_i$, $\mathbf{r}_j$ are the positions in the ith and jth molecule, respectively, $\langle\ldots\rangle$ means a thermal average, N is the number of atoms and ρ is the average density of the system. In other words $g(r)$ describes the radial density of the C and H atoms surrounding (solvent) the C and H atoms of a given (solute) molecule. This radial density however is not related to a stable structure but to a dynamical one in which the molecules wander around and interchaing the positions. Still the net results is that on the average there is a molecule in the related position. A comparison of the $g(r)$ values calculated at the temperature of 110 K on the two force fields is given in the upper panels of Figures 3.16 and 3.17 for methane and propane, respectively. The lower panels of the two figures show instead the value of $g(r)$ derived at the same temperature from the experiment for the same hydrocarbons.

We examine first the methane case. The upper panel of Fig. 3.16 shows as a solid black line values of $g(r)$ calculated at the temperature of $110K$ on the OPLS potential for the pairs C-C between the Carbon atom of a solute methane molecule and those of all the other (solvent) methane molecules. In the same figure the corresponding pair distribution function calculated on the Dreiding potential is given as a dotted black line. The two curves exhibit an almost coincident quite sharp peak having the same location and a height differing by only about 7%. An inspection of the spatial distribution of the molecules of the simulating ensemble shows that each methane molecule is surrounded by a quite well resolved spherical layer formed by other 3 methane molecules whose distance from the C atoms is on the average about 4.2 $\dot{A}$ away from it. This is, indeed, confirmed by the experimental study of Ref. [9] carried out at the same temperature of 110 K from which the plot of $g(r)$ given in the lower panel of the figure was reported. A sketch of the molecular arrangements in the first solvation sphere is given in Fig. 3.18 where the average distance and position of C and H atoms from the solute molecule are illustrated. The upper panel of Fig. 3.16 shows also at

ISSN: 2279–8773

Table 3.8: Macroscopic properties for the propane and methane bulk systems calculated after 1 ns of simulation time at the pressure of 1 *bar*

propane				
	Volume $/\dot{A}^3$	Temp. K	Pressure bar	Density kg/m^3
Exper.	3.6E4	230	1.013	582
OPLS	3.6E4	230	1.002	581
Dreiding	3.6E4	230	1.013	581
methane				
	Volume $/\dot{A}^3$	Temp. K	Pressure bar	Density kg/m^3
Exper.	1.8E4	110	1.013	422
OPLS	1.8E4	110	1.071	421
Dreiding	1.8E4	110	1.040	422

about 8 $\dot{A}$ a second broader maximum implicating the formation of a larger and more diffuse spherical layer of solvent methane molecules accomodating about 33 molecules. A second maximum located at about the same distance is shown also in the lower panel of the same Figure by the experimental data. The structure sketched in Fig. 3.18 is confirmed by the other two pairs of curves (blue and red) of Fig 3.16 related to C-H and H-H pair distribution functions. The almost coincidence of the C-H pair of curves (solid for the OPLS and dotted for the Dreiding potentials) indicate, in fact, that there are two spherical layers of H atoms surrounding the Carbon of the solute methane molecule located at an average distance of about 3.6 $\dot{A}$ and 4.9 $\dot{A}$. This difference reflects the average distance between the upper and the lower H atoms of the solvent methane molecule as shown in Figure 3.18. This is also in agreement with the structure of the H-H $g(r)$ plots which show a less resolved (yet three-modal) structure in which the three peaks located at about 3.0 $\dot{A}$, 4.3 $\dot{A}$ and 5.7 $\dot{A}$ respectively can be understood as the result of the coincidence between the first and the second H layers, belonging to the solute molecule, and the first and the second H layers belonging to the solvent molecule. Unfortunately, the poor resolution of the structure is insufficient to offer indications on the characteristic of the second $g(r)$ peak. A similar (yet less resolved) structure is shown by the plots of the propane molecules. These plots, given in the upper panel of Fig. 3.17, show that the propane pair distribution functions (calculated $g(r)$ are given in the upper

panel while the experimental ones [8] are given in the lower panel) are in agreement with the arrangements illustrated in Fig. 3.19 (though definitely smoother). The key difference is that now the C-C $g(r)$ plot has a bimodal first region (a first peak is located at about 4.5 $\dot{A}$ and the second about one 1 $\dot{A}$ above) associated with the fact that the central carbon is in the case of propane sandwiches by two other Carbon atoms (after all C_3H_8 can be seen of as a methane in which two Hydrogen atoms have been replaced by methyl groups). Again the agreement between the calculated and the measured pair distribution functions is satisfactory. As expected (see again Fig. 3.19) also the C-H and the H-H $g(r)$ have a more structured shape. The longer chain of the propane molecule explains also the displacement of the second maximum that is shifted by about 1.5 $\dot{A}$ with respect to the plots of Fig. 3.16, in agreement with the experimental data shown in the lower panel of Fig. 3.17. However, the smoothing down of the $g(r)$ structure would be probably better illustrated using different distribution functions more oriented towards larger and more structured molecules.

3.3.3 Gridification of DL_POLY on the EGEE Grid Platform

To investigate how Grid targeted modifications of the distribution strategies of the DL_POLY suite of codes would work, we implemented it on the EGEE production Grid platform within the activities carried out by the virtual organization COMPCHEM [2]. As a first step, to measure the performance of the code on the Grid and to single out the Grid features exploitable for the purpose of improving the statistics of the selected events, we ran the parallel version of the DL_POLY suite of codes, based on the Replicated Data parallelization strategy [149] on six different EGEE-Grid clusters of processors. In order to evaluate the elapsed time of each simulation and the related speed-up for each cluster, we ran the calculations sequentially on one node and in parallel on 2 and 4 nodes. As a second step, the parallelization of DL_POLY has been carried out at the coarsest granularity, by adopting a single program multiple data (SPMD) like model. In this simple model the program is distributed to all the available Grid nodes and executes on them concurrently in a parameter sweeping fashion. This model, described in the next section, has the advantage of being simple to implement and to be ideally suited for the present organization of the EGEE Grid.

Related measured elapsed times and calculated speedups are plotted in Fig. 3.20 and 3.21, respectively and the main features of the used clusters (cpu vendor, clock frequance and RAM size) are reported in Table 3.9. As

ISSN: 2279–8773

shown by the figures some clusters on the EGEE-Grid have a parallel performance very close to the ideal value because they allow a dedicated usage of the processors. Deviations from the ideal value not necessarily depend only from the time sharing regime adopted by some clusters. To investigate this aspect detailed evaluations of the parallel performances of the clusters and of the waiting time intercurring between the scheduling and the running of a process were obtained by restricting parallel calculations to two nodes. In Table 3.10 the the typical situation of our production grid runs in which 50 parallel jobs are executed is illustrated. As apparent from the table, more than 70% of the jobs ran properly and only 26% were aborted. The main reason for abortion was found in communication errors between the nodes of the same cluster (62%). About 15% of the failures were due to faults of the scheduler and another 23% to internal errors of DL_POLY at run time.

Contemporarily in order to measure the parallel performance of the code on a single node of the Grid, the parallel version of DL_POLY was run in parallel on the cex.grid.unipg.it node using simultaneously up to 16 nodes. The used cluster is made of Xeon cpus with a clock frequency of 2993 MHz and a RAM of 2048 MB. In order to evaluate the speed-up we ran the calculations sequentially on one node and in parallel on 2, 4, 8 and 16 nodes. Relevant measured elapsed times (upper panel) and calculated speedups (lower panel) are plotted in Fig. 3.22. Deviations from the ideal speedup (red dashed line) may depend not only on the time sharing regime adopted by the cluster but also on load imbalances generated by calculations run on different CPUs.

Table 3.9: Main features of the used clusters on EGEE-Grid platform.

Cluster	Location	CPU vendor	CPU clock (MHz)	RAM (MB)
atlas-na	Napoli (IT)	Opteron	2393	3943
isabella-gr	Crete (GR)	Xeon	2800	1024
grid-ct	Catania (IT)	Xeon	2800	2007
grid-pi	Pisa (IT)	Xeon	2800	1024
gridit-na	Napoli (IT)	Xeon	2400	2048
grid-ba	Bari (IT)	PentiumIV	2800	1024

ISSN: 2279–8773

Table 3.10: Statistics on EGEE-Grid submissions out of 50 jobs.

Job status	Number	%
Success	37	74
Aborted	13	26
Cause for abort	**Number**	**%**
Communic. error	8	62
DL_POLY error	3	23
Scheduler error	2	15

3.3.4 Alternative distribution schemes

The non negligible failure rate of the calculations due to internal DL_POLY reasons (and not to networking problems) adds further arguments to the need for developing alternative distribution strategies already pointed out in section 3. In fact, while failures due to either communication or scheduling reasons can be remedied by re-running the related computation, while waiting for the Grid to become more fault tolerant, internal DL_POLY failures imply that one or more starting conditions are inappropriate and make it impossible to properly sample initial conditions. To deal with this problem we discuss in the followings on how increasing the statistical significance of the sample after the thermalization process.
As is well known, the average value $< A >$ of the property A of a given system of N particles is defined as

$$< A >= \int \int dp^N dr^N A(p^N, r^N)\rho(p^N, r^N) \tag{3.20}$$

where $A(p^N, r^N)$ is the property A expressed as a function of the phase space variables p^N and r^N while $\rho(p^N, r^N)$ is the probability density of the ensemble. Usually, in Molecular Dynamics simulations $< A >$ is approximated by the time average $< A >_{time}$ defined as

$$< A >_{time}= \lim_{t\to\infty} \frac{1}{t} \int A(p^N(t), r^N(t))dt \simeq$$
$$N_\tau^{-1} \sum_{t=1}^{N} A(p^N(t), r^N(t)) \tag{3.21}$$

where N_τ is the number of sampling points considered. Accordingly, after partitioning the integral in N_k subsets we can also write

ISSN: 2279–8773

$$< A >= N_k^{-1} \sum_{k=1}^{N_k} < A >_k =$$

$$N_k^{-1} \sum_{k=1}^{N_k} N_{\tau k}^{-1} \sum_{t=1}^{N_{\tau k}} A(p^N(t), r^N(t)) \tag{3.22}$$

This leaves us with the choice of selecting subsets of the particles associated with appropriate configurations that can be obtained by randomly altering the after thermalization configurations having the same energy. This is obtained by randomly inverting the sign of a fixed number of momentum components of the particles of the system. As a result, the portion of the phase space sampled by the system changes without altering the global and individual total energy of the particles.

The distribution algorithm

To the end of incorporating the above described selection criterium, the section of the DL_POLY suite of programs taking care of starting the after thermalization evolution of the system was modified accordingly, in order to implement a coarse grained task farm model.

Therefore, in order to implement our selection algorithm, new MPI constructs have been added to the original code. Use has also been made of the preprocessor instructions to avoid the compilation of the original MPI implementation of the DL_POLY package (in particular we used the "SERIAL" macro instead of the "MPI" one commonly used for parallel compilations of the code). The related section of the code reads as sketched in Fig. 3.3.4. In the figure number_ra and therm_per are, respectively, a numerical variable indicating how many atoms are going to have a component of the momentum inverted and a logical variable becoming true only after completion of the thermalization process. The section of the code is articulated into two ifs the second of which discriminates the action to be taken by the master process from the one to be taken by the slave processes. The first if carries out a check for the completion of the thermalization and of the momenta inversions to be performed, while the second if allows the master process to create and distribute the array of the identities of the atoms whose momentum has to be modified and the slave process to perform the modification. It is important to point out here that the carrying out of the Molecular Dynamics simulation of the unperturbed system is left with the master process. The algorithm was tested on a DL_POLY demo application (using for a short number of steps, since this are the conditions of the standard DL_POLY benchmark used) on

ISSN: 2279–8773

a cluster of 8 biprocessor Pentium IV 2.8 GHz nodes.
The tests gave the following indications:

- the program scales very well; this is apparent from Fig. 3.24 where the scaled speedup is plotted against the number of processors used. The figure shows that the trend is almost linear up to 8 processors.
- the modification of the momenta of some atoms has a clear impact on the evolution of the simulation; the values of the observables calculated by the slaves processes show, in fact, some differences with respect to the ones calculated by the master process that takes care of carrying out the calculation for the unperturbed system. This means that the modifications introduced by our algorithm are effective in extending the region of the phase space covered by the simulation.
- the modification of the momenta of some atoms may lead to unstable configurations; this makes a certain number of runs to fail (this is usually termed as abnormal termination). The failure rate is, however, scarcely sensitive to the percentage of atoms whose momentum components were changed (see Table 3.11). Moreover, by performing the above mentioned variation, the net result is that in any case the statistics improves and the event that there are temperatures for which no results are available reduces.

Table 3.11: Failure rate

	300 steps		1000 steps	
% of changes	**Term.**	**Failed**	**Term.**	**Failed**
25 %	10	0	10	0
50 %	10	0	9	1
75 %	9	1	8	2
100 %	9	1	8	2

3.3.5 Conclusions

The study of hydrocarbon bulks (either pure or as solution) has carried out using the EGEE production computing Grid platform. In particular the

ISSN: 2279–8773

adoption of a double level parallel model for the DL_POLY suite of codes has allowed a massive run of molecular dynamics simulations.

In this way it has been possible to show that the gridification of the computational procedure can lead to a reduction of the elapsed time of the computation to about 1/5 on 8 CPUs. The possibility of distributing the jobs over the about 10.000 machines available to the COMPCHEM Virtual Organization has allowed a significant progress of our study and has opened interesting prespectives to massive molecular simulations.

More Specifically in our case this has allowed a rationalization of the behaviour of a methane molecule when treated as a solute in a propane bulk. The study has singled out the tendency of large propane molecules ensembles to cage solute molecules in statistically well defined (though liquid structures). This makes the present study of interest for the development of alternative ways of storing gaseous materials by trapping them inside specific solvent structures. Furthermore, the study of the phase diagram of this process may provide the ground for developing innovative refrigeration cycles.

ISSN: 2279-8773

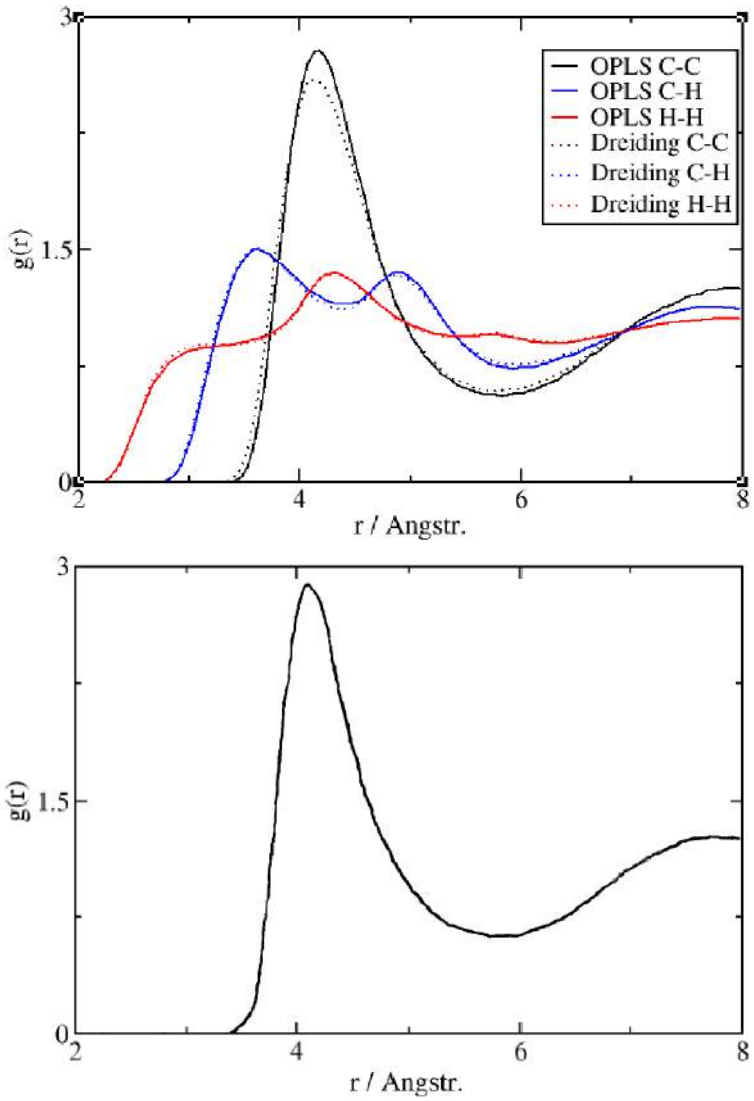

Figure 3.16: Upper panel: pair distribution functions plotted as a function of the radius for the methane system calculated for CC using the OPLS (black solid line) and Dreiding (black dotted line) formulation of the force field at the temperature of $110K$. Red and blue colors are used for H-H and C-H distributions. Lower panel: experimental C-C pair distribution function plotted as a function of the distance for the methane system obtained [9] at the same temperature.

ISSN: 2279-8773

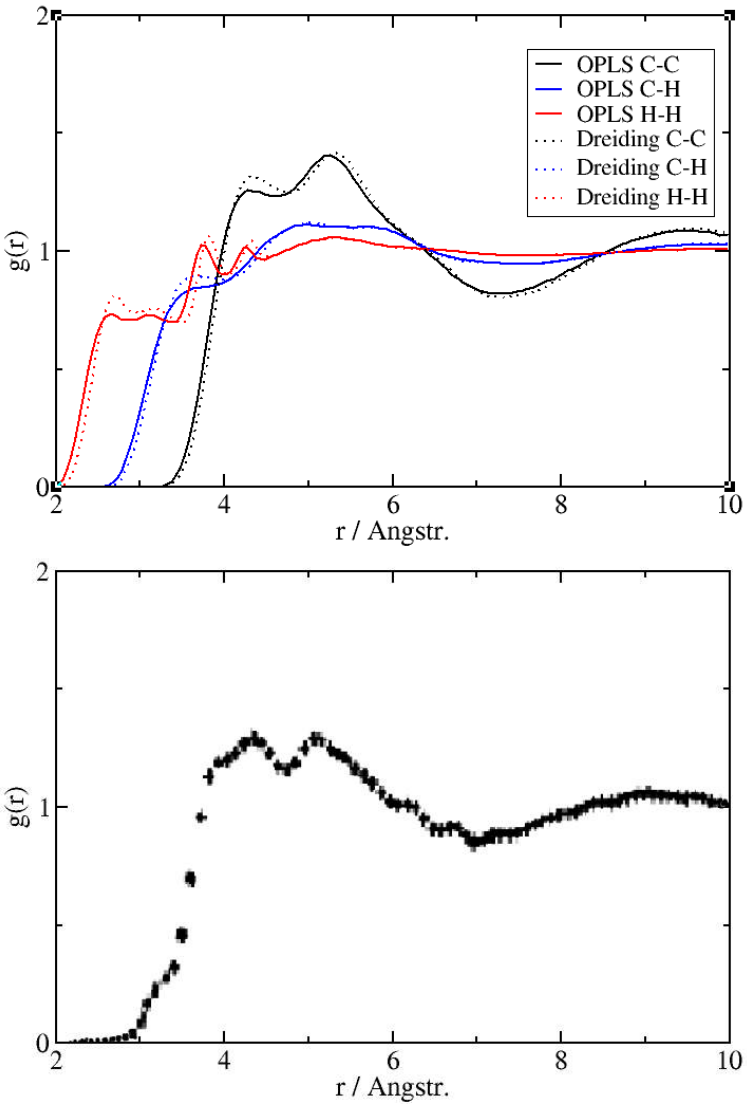

Figure 3.17: Upper panel: pair distribution functions plotted as a function of the radius for the propane system calculated for CC using the OPLS (black solid line) and the Dreiding (black dotted line) formulation of the force field at the temperature of $288K$. Red and blue colors are used for the H-H and C-H distributions. Lower panel: experimental C-C pair distribution function plotted as a function of the distance for the propane system obtained [8] at the same temperature.

ISSN: 2279-8773

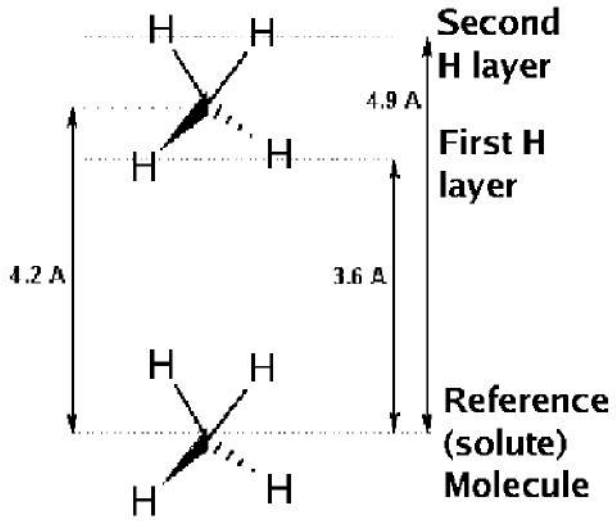

Figure 3.18: A pseudo three-dimensional view of methane molecules

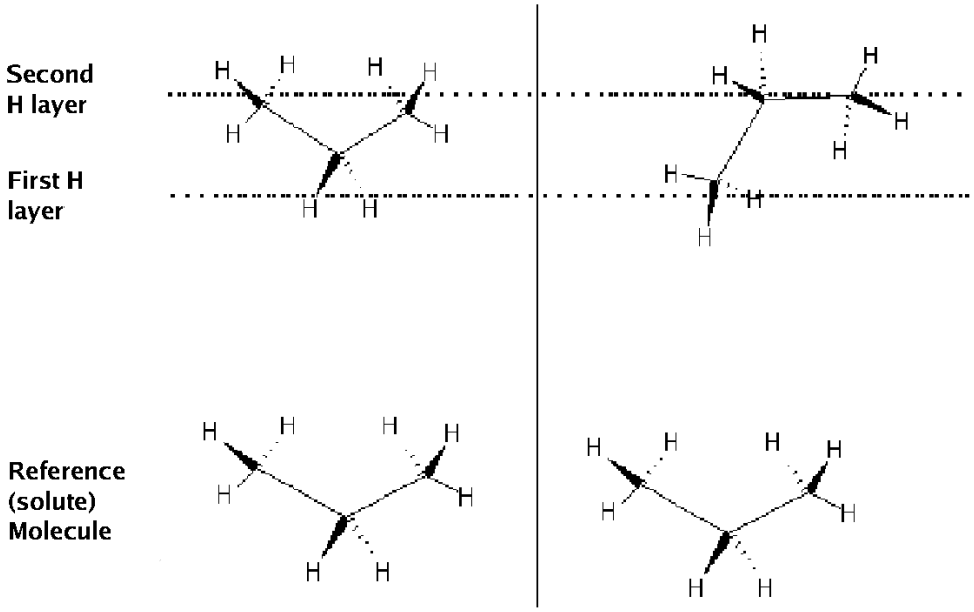

Figure 3.19: A pseudo three-dimensional view of propane molecules

ISSN: 2279-8773

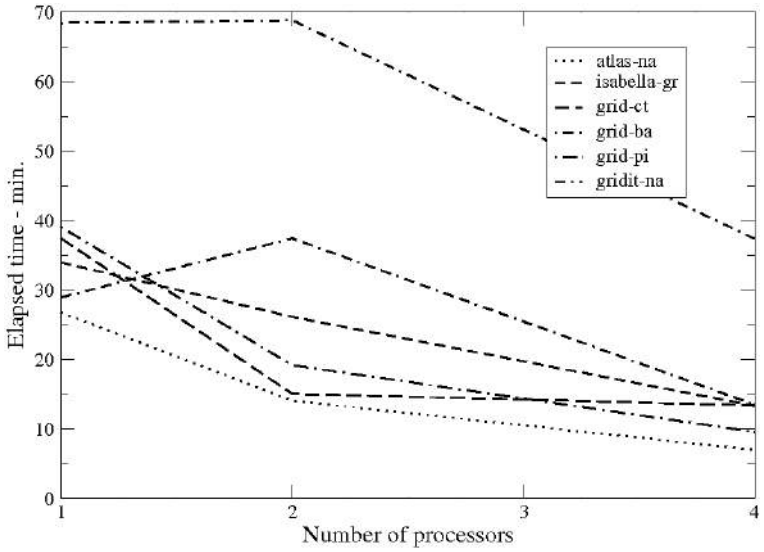

Figure 3.20: Elapsed time measured on six different EGEE-Grid clusters (listed in the right hand side of the graph) plotted as a function of the number of processors used.

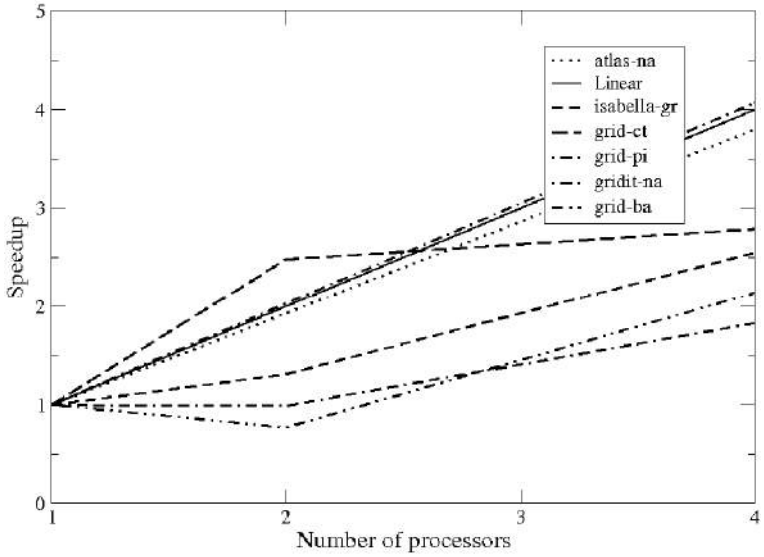

Figure 3.21: As in Figure 3.20 for speedups.

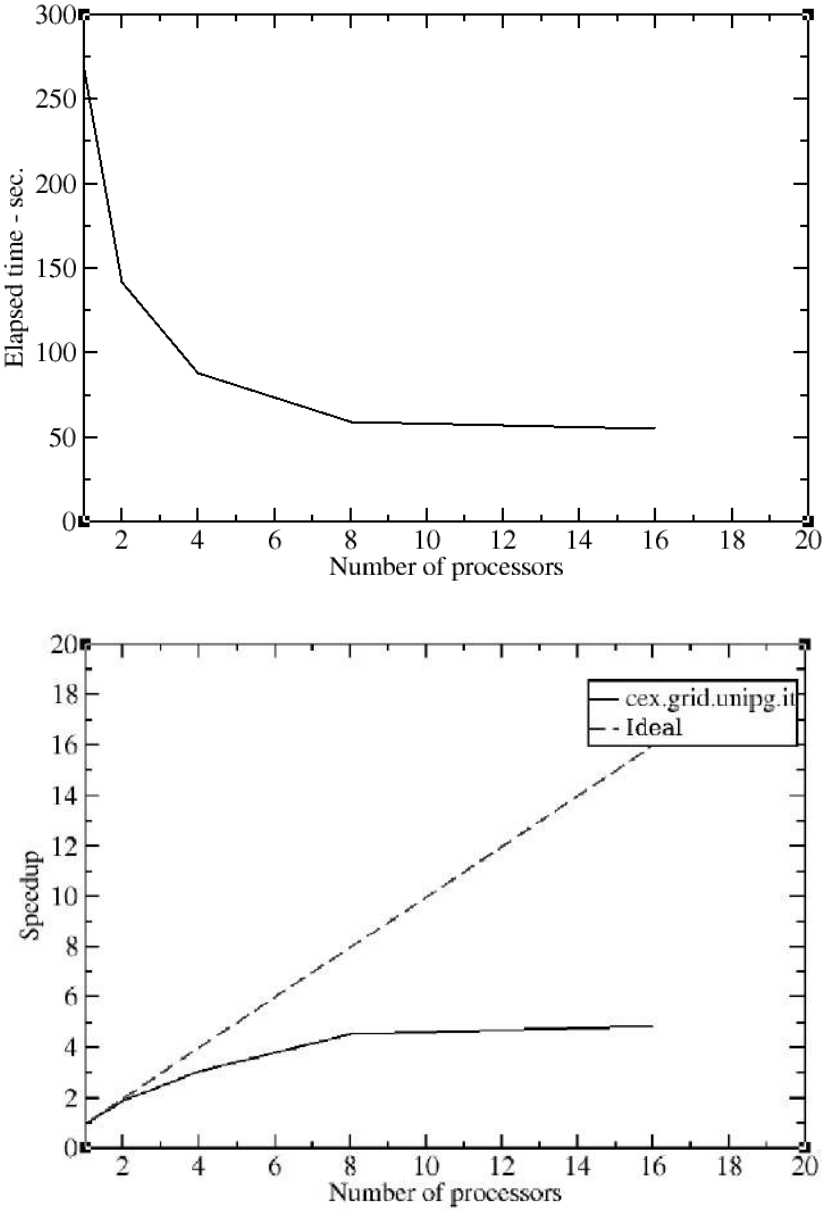

Figure 3.22: Elapsed time (upper panel) and speedup (lower panel) measured for cex.grid.unipg.it cluster plotted as solid line as a function of the number of processors used. For comparison also the ideal speedup (dashed line) is given in the lower panel.

ISSN: 2279-8773

```
IF (random_ra is zero or therm_per is not finished) goto "skip"
IF (myrank is zero) then
      LOOP on slave processes
            LOOP on random_ra
                  randomly generate atom_array(myrank)
            END loop on random_ra
            MPI_Send to the slave processes
            the atom_array(myrank)
      END loop on slave processes
ELSE IF (myrank not zero) then
      MPI_Receive from master process the atom_array(myrank)
      LOOP on random_ra
            invert the i-th momentum component of the atoms
            of atom_array(myrank)
      END loop on random_ra
END IF
"skip"
```

Figure 3.23: Pseudocode of the distribution algorithm

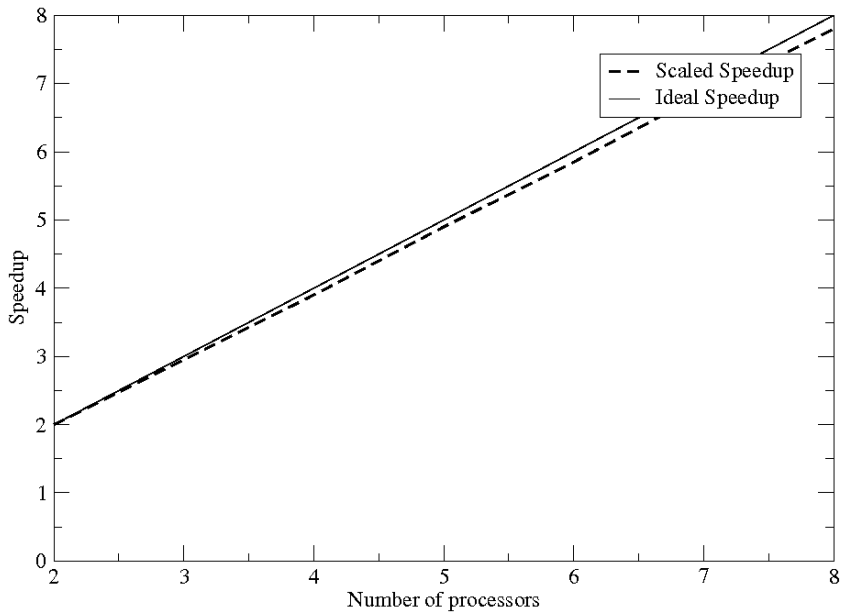

Figure 3.24: Scaled Speedup

ISSN: 2279–8773

3.4 The Multiscale study of O_3 Tropospheric

A third family of computational applications we have considered for implementation on the grid is the multiscale suite of codes modelling the production of secondary pollutants in the atmosphere. The atmosphere, in fact, represents an invaluable shared commodity and its quality is increasingly being controlled for preservation. The major threat to the air quality is usually represented by the pollutants released in the atmosphere by human activities. These emissions modify the atmosphere composition and worsen its quality with a consequent damage to human health and to the ecosystem not only because of their direct effect but also as a result of the substances produced by them by reacting with light and/or other species.

The European Community has issued a directive (96/62/CE) [11] which recommends the Member States to adopt specific tools for controlling air quality. The recommendations include, among other suggestion, the encouragement to implement modeling packages. The software used for this purpose integrates at the same time on different scales the fluid dynamics equations (for the transport of gaseous masses and dusts) and chemical equations (for the interactions between matter and light) by taking into account simulataneously the orography and weather conditions. Related computational codes take therefore as input data from emission inventories, weather conditions and geographical information to produce as output atmospheric concentrations of several kinds of pollutants including some secondary pollutants like ozone (O_3) and thin dust ($PM_{2.5}$ and PM_{10}).

This kind of computational simulations allow to simulate short term acute episodes as well as long-term sustained trends. They allow also a comparison of the calculated values of the pollutants with the ones measured locally, and operational forecast as well meant to support recovery strategies (reduction plans and pollution control).

In the present Section the steps for implementing on the EGEE grid the necessary computational tools are described, a study case concerning the modeling of air quality in the Umbria Region is considered and the comparison of the calculated results with the measured ones, are discussed. In particular the production of secondary pollutants during the summer 2004 and its impact on the Umbria territory has been studied. The work has been carried out in collaboration with ARPA (Regional Agency for Prevention and Environment) Umbria [13].

ISSN: 2279–8773

3.4.1 Multiscale approaches to atmospheric secondary pollution

Atmospheric dispersion modeling is the mathematical simulation of how air pollutants disperse in the ambient atmosphere. It is performed using computer programs which solve the mathematical equations governing the pollutant dispersion. The dispersion models are used to estimate or to predict the downwind concentration of air pollutants emitted from sources such as industrial plants and vehicular traffic. Such models are important to governmental agencies tasked with protecting and managing the ambient air quality. The models are typically employed to determine whether existing or proposed new industrial facilities are or will be in compliance with the National Ambient Air Quality Standards (NAAQS) in the United States and other nations. The models also serve to assist in the design of effective control strategies to reduce emissions of harmful air pollutants.

The dispersion models require the input of data which includes:

- Meteorological conditions such as wind speed and direction, the amount of atmospheric turbulence (as characterized by what is called the "stability class"), the ambient air temperature and the height to the bottom of any inversion aloft that may be present.

- Emissions parameters such as source location and height, source vent stack diameter and exit velocity, exit temperature and mass flow rate.

- Terrain elevations at the source location and at the receptor location.

- The location, height and width of any obstructions (such as buildings or other structures) in the path of the emitted gaseous plume.

Many of the modern, advanced dispersion modeling programs include a pre-processor module for the input of meteorological and other data, and many also include a graphic interface for graphing the output data and/or plotting the area impacted by the air pollutants on maps.

The atmospheric dispersion models are also known as atmospheric diffusion models, air dispersion models, air quality models, and air pollution dispersion models.

Pollutants

An air pollutant is known as a substance in the air that can cause harm to humans and the environment. Pollutants can be in the form of solid

ISSN: 2279–8773

particles, liquid droplets, or gases. In addition, they may be natural or man-made. Pollutants can be classified as either primary or secondary. Primary pollutants are substances directly emitted from a process, such as ash from a volcanic eruption, the carbon monoxide gas from a motor vehicle exhaust or sulfur dioxide released from factories. Secondary pollutants are not emitted directly. They rather form in the air when primary pollutants react or interact. An important example of a secondary pollutants is ground level ozone, one of the many secondary pollutants which make up photochemical smog. Note that some pollutants may be both primary and secondary: that is, they are both emitted directly and formed from other primary pollutants.

Major "primary pollutants" produced by human activity include:

- Sulfur oxides (SO_x), especially sulfur dioxide, a chemical compound with the formula SO_2. SO_2 is produced by volcanoes and in various industrial processes. Since coal and petroleum often contain sulfur compounds, their combustion generates sulfur dioxide. Further oxidation of SO_2, usually in the presence of a catalyst such as NO_2, forms $SO_e^=$ (and thus H_2SO_4 and acid rain). This is one of the causes for concern over the environmental impact of the use of these fuels as power sources.

- Nitrogen oxides (NO_x), especially nitrogen dioxide, are emitted from high temperature combustion. They can be seen as the brown haze dome above or plume downwind of cities. Nitrogen dioxide is the chemical compound with formula NO_2. It is one of the several nitrogen oxides. This reddish-brown toxic gas has a characteristic sharp, biting odor. NO_2 is one of the most prominent air pollutants.

- Carbon monoxide is colourless, odourless, non-irritating but very poisonous gas. It is a product by incomplete combustion of fuel such as natural gas, coal or wood. Vehicular exhaust is a major source of carbon monoxide.

- Carbon dioxide (CO_2), a greenhouse gas emitted from combustion when it take place in excess of O_2.

- Volatile organic compounds VOCs are an important outdoor air pollutant. In this field they are often divided into the separate categories of methane (CH_4) and non-methane (NMVOCs). Methane is an extremely efficient greenhouse gas which contributes to enhanced global warming. Other hydrocarbon VOCs (that are also significant greenhouse gases) have a role in creating ozone and in prolonging the life

ISSN: 2279-8773

of methane in the atmosphere, although the effect varies depending on local air quality. Within the NMVOCs, the aromatic compounds benzene, toluene and xylene are suspected carcinogens and may lead to leukemia through prolonged exposure. 1,3-butadiene is another dangerous compound which is often associated with industrial uses.

- Particulates, alternatively referred to as particulate matter (PM) or fine particles, are tiny particles of solid or liquid suspended in a gas. In contrast, aerosol refers to particles and the gas together. Sources of particulate matter can be man made or natural. Some particulates occur naturally, originating from volcanoes, dust storms, forest and grassland fires, living vegetation, and sea spray. Human activities, such as the burning of fossil fuels in vehicles, power plants and various industrial processes also generate significant amounts of aerosols. Averaged over the globe, anthropogenic aerosols-those made by human activities-currently account for about 10 percent of the total amount of aerosols in our atmosphere. Increased levels of fine particles in the air are linked to health hazards such as heart disease, altered lung function and lung cancer.

- Toxic metals, such as lead, cadmium and copper.

- Chlorofluorocarbons (CFCs), harmful to the ozone layer emitted from products currently banned from use.

- Ammonia (NH_3) emitted from agricultural processes. Ammonia is a compound with the formula NH_3. It is normally encountered as a gas with a characteristic pungent odor. Ammonia contributes significantly to the nutritional needs of terrestrial organisms by serving as a precursor to foodstuffs and fertilizers. Ammonia, either directly or indirectly, is also a building block for the synthesis of many pharmaceuticals. Although in wide use, ammonia is both caustic and hazardous.

- Odors, such as from garbage, sewage, and industrial processes

- Radioactive pollutants produced by nuclear explosions, war explosives, and natural processes such as the radioactive decay of radon.

Secondary pollutants include:

- Particulate matter formed from gaseous primary pollutants and compounds in photochemical smog. Smog is a kind of air pollution; the word “smog” is a portmanteau of smoke and fog. Classic smog results

ISSN: 2279-8773

from large amounts of coal burning in an area caused by a mixture of smoke and sulfur dioxide. Modern smog does not usually come from coal but from vehicular and industrial emissions. In the atmosphere the smog is acted on by sunlight to form secondary pollutants which also combine with the primary emissions to form photochemical smog.

- Ground level ozone (O_3) formed from NOx and VOCs. Ozone (O_3) is a constituent of the troposphere (it is also an important constituent of certain regions of the stratosphere commonly known as the Ozone layer). Photochemical and chemical reactions involving it drive some of the chemical processes that occur in the atmosphere by day and by night. At abnormally high concentrations brought about by human activities (largely the combustion of fossil fuel), it is a pollutant, and a constituent of smog.

- Peroxyacetyl nitrate (PAN) similarly formed from NOx and VOCs.

Minor air pollutants include:

- A large number of minor hazardous air pollutants. Some of these are regulated in USA under the Clean Air Act and in Europe under the Air Framework Directive.

- A variety of persistent organic pollutants, which can attach to particulate matter.

Persistent organic pollutants (POPs) are organic compounds which are resistant to environmental degradation through chemical, biological, and photolytic processes. Because of this, they have been observed to persist in the environment, to be capable of long-range transport, bioaccumulate in human and animal tissue, biomagnify in food chains, and to have potential significant impacts on human health and the environment.

Atmospheric layers

Discussion of the layers in the Earth's atmosphere is needed to understand where airborne pollutants disperse in the atmosphere. The layer closest to the earth's surface is known as the troposphere. It extends from sea-level to a height of about 18 km and contains about 80 percent of the mass of the overall atmosphere. The stratosphere is the next layer and extends from 18 km to about 50 km. The third layer is the mesosphere which extends from 50 km to about 80 km. There are other layers above 80 km, but they are insignificant with respect to atmospheric dispersion modeling.

ISSN: 2279-8773

The lowest part of the troposphere is called the atmospheric boundary layer (ABL) or the planetary boundary layer (PBL) and extends from the Earth's surface to about 1.5 to 2.0 km in height. The air temperature of the atmospheric boundary layer decreases with increasing altitude until it reaches what is called the inversion layer (where the temperature increases with increasing altitude) that caps the atmospheric boundary layer. The upper part of the troposphere (i.e., above the inversion layer) is called the free troposphere and it extends up to the 18 km height of the troposphere.

The ABL is of the most importance with respect to the emission, transport and dispersion of airborne pollutants. The part of the ABL between the Earth's surface and the bottom of the inversion layer is known as the mixing layer. Almost all of the airborne pollutants emitted into the ambient atmosphere are transported and dispersed within the mixing layer. Some of the emissions penetrate the inversion layer and enter the free troposphere above the ABL.

In summary, the layers of the Earth's atmosphere from the surface of the ground upwards are: the ABL made up of the mixing layer capped by the inversion layer; the free troposphere; the stratosphere; the mesosphere and others. Many atmospheric dispersion models are referred to as boundary layer models because they mainly model air pollutant dispersion within the ABL. To avoid confusion, it should be noted that models referred to as mesoscale models have dispersion modelling capabilities that extend horizontally up to a few hundred kilometres. It does not mean that they model dispersion in the mesosphere.

Air pollutant emission sources

Sources of air pollution refer to the various locations, activities or factors which are responsible for the releasing of pollutants in the atmosphere. These sources can be classified into two major categories which are:

Anthropogenic sources (human activity) mostly related to burning different kinds of fuel:

- Stationary Sources as smoke stacks of power plants, manufacturing facilities, municipal waste incinerators. Power plant is also used to refer to the engine in ships, aircraft and other large vehicles. Some prefer to use the term energy center because it more accurately describes what the plants do, which is the conversion of other forms of energy, like chemical energy, gravitational potential energy or heat energy into electrical energy.

ISSN: 2279-8773

- Mobile Sources as motor vehicles, aircraft etc. Exhaust gases of automobiles and air crafts play a very important role in polluting the atmosphere.
- Marine vessels, such as container ships or cruise ships, and related port air pollution.
- Burning wood, fireplaces, stoves, furnaces and incinerators.
- Oil refining, and industrial activity in general. The refining process releases numerous different chemicals into the atmosphere; consequently, there are substantial air pollution emissions and a notable odor normally accompanies the presence of a refinery. Aside from air pollution impacts there are also wastewater concerns, risks of industrial accidents such as fire and explosion, and noise health effects due to industrial noise.
- Chemicals, dust and controlled burn practices in agriculture and forestry management. Controlled or prescribed burning is a technique sometimes used in forest management, farming, prairie restoration or greenhouse gas abatement. Fire is a natural part of both forest and grassland ecology and controlled fire can be a tool for foresters. Controlled burning stimulates the germination of some desirable forest trees, thus renewing the forest.
- Fumes from paint, hair spray, varnish, aerosol sprays and other solvents.
- Waste deposition in landfills, which generate methane. Methane is not toxic; however, it is highly flammable and may form explosive mixtures with air. Methane is also an asphyxiant and may displace oxygen in an enclosed space. Asphyxia or suffocation may result if the oxygen concentration is reduced to below 19.5% by displacement.
- Military, such as nuclear weapons, toxic gases, germ warfare and rocketry.

Natural sources :

- Dust from natural sources, usually large areas of land with little or no vegetation.
- Methane, emitted by the digestion of food by animals, for example cattle.

ISSN: 2279-8773

- Radon gas from radioactive decay within the Earth's crust. Radon is a colorless, odorless, naturally occurring, radioactive noble gas that is formed from the decay of radium. It is considered to be a health hazard. Radon gas from natural sources can accumulate in buildings, especially in confined areas such as the basement and it is the second most frequent cause of lung cancer, after cigarette smoking.
- Smoke and carbon monoxide from wildfires.
- Volcanic activity, which produce sulfur, chlorine, and ash particulates.

The types of air pollutant emission sources can be also characterized as either point, line, area or volume sources:

Point source - A point source is a single, identifiable source of air pollutant emissions (for example, the emissions from a combustion furnace flue gas stack). Point sources are also characterized as being either elevated or at ground-level. A point source has no geometric dimensions.

Line sources - A line source is one-dimensional source of air pollutant emissions (for example, the vehicular traffic on a roadway).

Area source - An area source is a two-dimensional source of diffuse air pollutant emissions (for example, a forest fire, a landfill or the evaporated vapors from a large spill of volatile liquid).

Volume source - A volume source is a three-dimensional source of diffuse air pollutant emissions. Essentially, it is an area source with a third (height) dimension (for example, the fugitive gaseous emissions from piping flanges, valves and other equipment at various heights within industrial facilities such as oil refineries and petrochemical plants). Another example would be the emissions from an automobile paint shop with multiple roof vents or multiple open windows.

Other air pollutant emission source characterizations are:

- Sources may be characterized as either stationary or mobile. Flue gas stacks are examples of stationary sources and busses are examples of mobile sources.
- Sources may be characterized as either urban or rural because urban areas constitute a so-called heat island and the heat rising from an urban area causes the atmosphere above an urban area to be more turbulent than the atmosphere above a rural area.

ISSN: 2279–8773

- Sources may be characterized by their elevation relative to the ground as either surface or ground-level, near surface or elevated sources.
- Sources may also be characterized by their time duration:
  - puff or intermittent: short term sources (for example, many accidental emission releases are short term puffs)
  - continuous: a long term source (for example, most flue gas stack emissions are continuous)

Air pollution dispersion models

There are five types of air pollution dispersion models, as well as some hybrid ones. The five types of models are:

Box model - This is the simplest model [150]. It assumes the airshed (i.e., a given volume of atmospheric air in a geographical region) has a box shape. It also assumes that the air pollutants inside the box are homogeneously distributed and uses this assumption to estimate the average pollutant concentrations anywhere within the airshed. Although useful, this model is quite inaccurate in predicting the dispersion of air pollutants over an airshed due to the variety of the assumption of a homogeneous distribution of the pollutant.

Gaussian model - The Gaussian model is perhaps the oldest (it dates back to approximately 1936) [151] and perhaps the most commonly used model type. It assumes that the air pollutant dispersion has a Gaussian distribution, meaning that the pollutant distribution has a normal probability distribution. Gaussian models are most often used to predict the dispersion of continuous, buoyant air pollution plumes originating from ground-level or elevated sources. Gaussian models may also be used to predict the dispersion of non-continuous air pollution plumes (called puff models). The primary algorithm used in Gaussian modeling is the Generalized Dispersion Equation For A Continuous Point-Source [153].

Lagrangian model - a Lagrangian dispersion model mathematically follows pollution parcels (also called particles) as the parcels move in the atmosphere and they model the motion of the parcels as a random walk process. The Lagrangian model then calculates the air pollution dispersion by computing the statistics of the trajectories making use of small boxes as indicated in the "Box Model" discussed above. A

ISSN: 2279–8773

Lagrangian model uses a moving frame of reference [154] as the parcels move from their initial location.

Eulerian model - an Eulerian dispersions model is similar to a Lagrangian model in that it also tracks the movement of a large number of pollution parcels as they move from their initial location. The most important difference between the two models is that the Eulerian model uses a fixed three-dimensional Cartesian grid [154] as a frame of reference rather than a moving frame of reference.

Dense gas model - Dense gas models are models that simulate the dispersion of dense gas pollution plumes (i.e., pollution plumes that are heavier than air). The three most commonly used dense gas models are:

- The DEGADIS model developed by Dr. Jerry Havens and Dr. Tom Spicer at the University of Arkansas under commission by the US Coast Guard and US EPA [155].
- The SLAB model developed by the Lawrence Livermore National Laboratory funded by the US Department of Energy, the US Air Force and the American Petroleum Institute [156].
- The HEGADAS model developed by Shell Oil's research division [157].

Eulerian air pollutant dispersion equations

The technical literature on air pollution dispersion is quite extensive and dates back to the 1930's and earlier. One of the early air pollutant dispersion equations was derived by Bosanquet and Pearson [151]. Their equation did not assume a Gaussian or Eulerian distribution nor did include the effect of ground reflection of the pollutant.

The theoretical basis of each Eulerian model [152] type is the equation that expresses the instantaneous mass balance for the various species involved in the simulation (not taking into account the molecular diffusion). Considering N species, a generic equation which considers the flux species i through a box is the following:

$$F_i = (c_i \cdot u_1 - c_i \cdot u_2)\delta y \cdot \delta z \tag{3.23}$$

where c_i is the concentration of the species i, u_1 represents the velocity of the air through the left face and u_1 represents the velocity of the air of the outgoing flux (see Fig. 3.25 for details).

ISSN: 2279–8773

The instantaneous mass balance requires that the fluctuation in concentation of the i species in the box is equal to the algebraic sum of the flows, of the sources present in the box and of the transformation and removal processes taking place in the box. In fact, we have that:

$$\begin{aligned}(\delta x \cdot \delta y \cdot \delta z \cdot \frac{\partial c_i}{\partial t}) &= -\delta x \cdot \delta y \cdot \delta z \left\{ \frac{\partial c_i u}{\partial x} + \frac{\partial c_i v}{\partial y} + \frac{\partial c_i w}{\partial z} \right\} \\ &\quad +\delta x \cdot \delta y \cdot \delta z \cdot (S + R + T)\end{aligned} \tag{3.24}$$

where S is the source term, R is the removal term and T is the term that takes into account the chemical transformations and beyond. This equation is not depending from the volume of the box considered, therefore we have

$$\frac{\partial c_i}{\partial t} + \frac{\partial c_i u}{\partial x} + \frac{\partial c_i v}{\partial y} + \frac{\partial c_i w}{\partial z} = S + R + T \tag{3.25}$$

that is the instantaneous mass balance equation for a generic species i.

The resulting calculations for air pollutant concentrations are often expressed as an air pollutant concentration contour map in order to show the spatial variation in contaminant levels over a wide area. In this way the contour lines can overlay sensitive receptor locations and reveal the spatial relationship of air pollutants to the areas of interest.

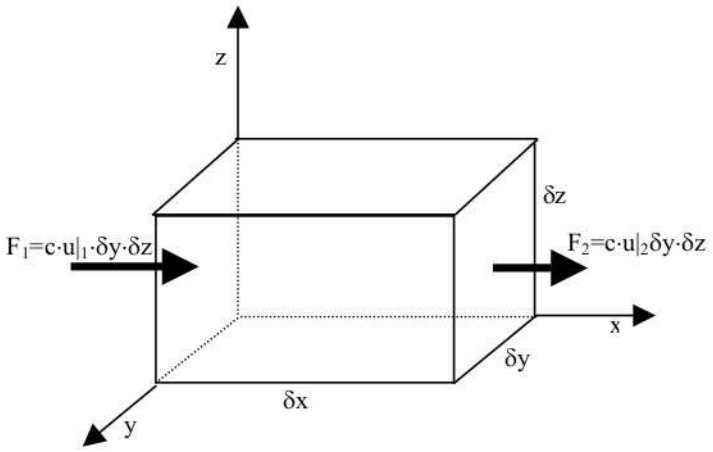

Figure 3.25: Representation of the box for the Eulerian model.

ISSN: 2279-8773

Chemistry in Air Quality Modelling

Chemical and Photochemical air quality models have become widely utilized tools for assessing the effectiveness of control strategies adopted by regulatory agencies. These models are large-scale air quality models which simulate the changes of pollutant concentrations in the atmosphere by characterizing the chemical and physical processes in the atmosphere. These models are applied at multiple geographical scales ranging from local and regional to national and global.

In order to synthesise and test theoretical understanding of atmospheric chemistry, computer models (such as chemical transport models) are used. Numerical models solve the differential equations governing the concentrations of chemicals in the atmosphere. They can vary from very simple to highly compliex. One common trade off in numerical models is between the number of chemical compounds and chemical reactions modelled versus the representation of transport and mixing in the atmosphere. For example, a box model might include hundreds or even thousands of chemical reactions but have only a very crude representation of mixing in the atmosphere. In contrast, 3D models represent many of the physical processes of the atmosphere because of the constraints set on the computer resources yet having far fewer chemical reactions and compounds than needed. Models can be used to interpret observations, to understand underlying chemical reactions and predict future concentrations of chemical compounds in the atmosphere. One important current trend is for atmospheric chemistry modules to become one part of earth system models in which the links between climate, atmospheric composition and biosphere activities can be studied.

Some models are constructed by automatic code generators. In this approach a set of constituents are chosen and the automatic code generator will then select the reactions involving those constituents from a set of reaction databases. Once the reactions have been chosen the ordinary differential equations (ODE) describing their time evolution can be automatically constructed.

Chemical models

Chemical air quality models have become widely utilized tools for assessing the effectiveness of control strategies adopted by regulatory agencies. These models are large-scale air quality models that simulate the changes of pollutant concentrations in the atmosphere by characterizing the chemical and physical processes in the atmosphere. These models are applied at multiple geographical scales ranging from local and regional to national and global.

ISSN: 2279-8773

The most known for them are:

Models-3/CMAQ - The latest version of the Community Multi-scale Air Quality (CMAQ) model has state-of-the-science capabilities for conducting urban to regional scale simulations of multiple air quality issues, including tropospheric ozone, fine particles, toxics, acid deposition, and visibility degradation.

CAMx - The Comprehensive Air quality Model with extensions (CAMx) simulates air quality over many geographic scales. It handles a variety of inert and chemically active pollutants, including ozone, particulate matter, inorganic and organic $PM_{2.5}$ and PM_{10}, and mercury and other toxics.

REMSAD - The Regional Modeling System for Aerosols and Deposition (REMSAD) calculates the concentrations of both inert and chemically reactive pollutants by simulating the atmospheric processes that affect pollutant concentrations over regional scales. It includes processes relevant to regional haze, particulate matter and other airborne pollutants, including soluble acidic components and mercury.

UAM-V - The Urban Airshed Model was a pioneering effort in photochemical air quality modelling in the early 1970s and has been used widely for air quality studies focusing on ozone.

CHIMERE - The CHIMERE multi-scale model is primarily designed to produce daily forecasts of ozone, aerosols and other pollutants and make long-term simulations for emission control scenarios. CHIMERE runs over a range of spatial scales from the regional scale (several thousand kilometers) to the urban scale (100-200 Km) with resolutions from 1-2 Km to 100 Km.

3.4.2 CHIMERE and developed interfaces

The computational tool chosen for our purpose is CHIMERE [12]. CHIMERE is a multi-scale package based on a chemistry and transport eulerian model that carries out air quality simulations. CHIMERE has been developed by the Institut Pierre-Simon Laplace (C.N.R.S.), INERIS and LISA (C.N.R.S.). It was primarily designed to carry out daily forecasts of Ozone, aerosols and other pollutants and make long-term simulations of pollutants concentrations for building emission control scenarios. CHIMERE spans spatial ranges going from the regional (several thousand kilometers) to the urban (100-200 Km)

ISSN: 2279–8773

scale with a resolution varying from 1-2 Km to 100 Km. At the same time it deals with the molecular aggregates scales ranging from small molecules (fractions of nanometer) to particulates (micron). CHIMERE is a portable model which has been adapted to various types of input data. It requires in fact meteorological data, boundary conditions, land-use information, anthropic and biogenic emissions which are usually downloaded from different sources. Moreover, CHIMERE can deal with the most typical chemical-physics molecular phenomena like transport, diffusion, deposition, chemical and photo chemical reactions involving gas, liquid and heterogeneous phases like gas-liquid interface (aereosol).

3.4.3 The CHIMERE workpackage

The version of CHIMERE chosen for implementation is the 200606A one (this was the most stable version of the code available when we started this investigation and the first validated parallel version). It is based on a MPMD (Multiple Program Multiple Data) model and makes use of the LAM/MPI parallel libraries [160].

The CHIMERE model consists of two separate modules:

- The meteorological interface which prepares the meteorological data in the netCDF [161] format needed by CHIMERE, diagnoses turbulent parameters and calculates biogenic emissions (meteorologically dependent);

- The CHIMERE code itself.

The general structure contains a central chemistry-transport calculation unit with interfaces to several databases (emissions, meteorology, boundary conditions, land use) as shown in Figure 3.26. Each interface transforms original or processed data to CHIMERE-input formatted data adapted to the time period and region to simulate.

CHIMERE offers the option to include different gas phase chemical mechanisms. The original, complete scheme [162], called MELCHIOR1, describes more than 300 reactions of 80 gaseous species. The hydrocarbon degradation is fairly similar to the EMEP gas phase mechanism [163]. Adaptations are made in particular for low NO_x conditions and NO_x -nitrate chemistry. All rate constants are updated according to [164] and [165]. Heterogeneous formation of HONO from deposition of NO2 on wet surfaces is also considered, using the formulation of [166]. In order to reduce the computing time a reduced mechanism with 44 species and about 120 reactions is derived from

ISSN: 2279–8773

MELCHIOR [167], following the concept of "chemical operators" [168]. This reduced mechanism is called MELCHIOR2.

The most complex interface of CHIMERE is the meteorological interface because it uses two steps. The first one, meteo, transforms original meteorological data (standard variables) given on the meteo model grid, at a frequency that is not necessarily one hour, to the same variables but on the CHIMERE grid (space interpolation) at the hourly sampling rate (time interpolation) in a specific file format. The second step for the meteorological interface is the diagnostic model used to take standard meteorological variables on the CHIMERE grid and transform them into input parameters. As an example, the boundary layer height can be diagnosed from the vertical profile of temperature, humidity and wind.

The boundary conditions interface takes either the outcome of a "coarse run" produced by CHIMERE and interpolates hourly pollutant concentrations at the domain boundaries, or provide climatological values at the domain boundaries from a global database consisting of simulation outputs.

The land-use interface is used to construct necessary terrain files and is used only once per domain construction.

The model core organization consists of a call to a main FORTRAN program which initializes and reads all data and runs an integration process in order to conver the time period specified. The output files are then available for postprocessing.

The whole system is pivoted by a top calling script which :

- reads parameters of the simulation
- links all necessary files into a temporary directory where all programs are executed
- compiles all necessary code
- runs the three "dynamic" interfaces (meteo, emission and boundary conditions)
- runs the model itself

The top calling script is executed to carry out a simulation. It must be edited and the options/directories must be chosen and defined by changing the values of parameters inside the script.

3.4.4 The CHIMERE output data

Four different output files are delivered. They are encoded in the netCDF format, which presents the following advantages :

ISSN: 2279-8773

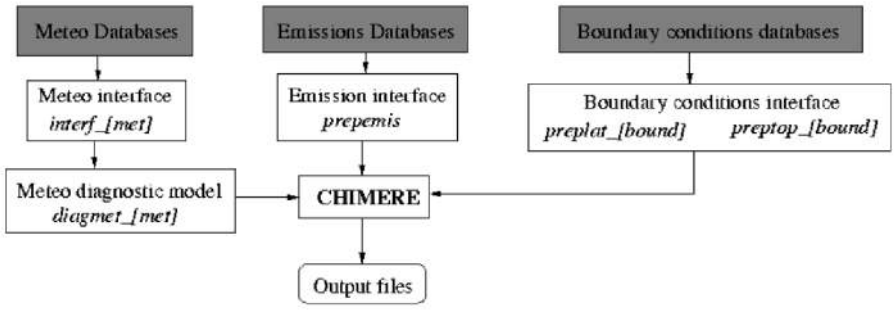

Figure 3.26: Representation of the structure of the CHIMERE input data

- portability among different architectures of computers
- self-documentation since a lot of meta-data are included in the file itself
- direct access
- compatibility with many free or commercial post-processing tools

Moreover, CHIMERE uses a netCDF convention derived from that used in NCAR'S WRF mesoscale modelling system which simplifies the toolset for post-processing. This contains:

chemical fields Required three-dimensional chemical fields are written in an output binary file called *out.*[*label*].*nc*. The selected species correspond to those listed in the OUTPUT_SPECIES parameter file. Default values are expressed in ppb. Output values may be transformed into $\mu g \cdot m^{-3}$ if the molar mass of each species is added as a second column in the parameter files.

deposition fields This file (*dep.*[*label*].*nc*) contains fluxes integrated over the chemical time-step duration. For the dry deposition flux, ony the first vertical cell is taken into account. On the other hand, for the wet deposition, fluxes are sumed up the whole column since, in the model, each vertical level may contribute to a net sink. Units of these two integrated fluxes are g/cm^2.

restart fields The main goal of this file called *end.*[*label*].*nc* is to save all concentrations fields every 24h. For example, for a second run used to give realistic initial conditions during the restart.

ISSN: 2279–8773

additional fields An additional output file is named *par.*[*label*].*nc*. Its goal is to provide meteorological outputs, corresponding to the values employed during the run to calculate the concentration fields written in *out.*[*label*].*nc*.

3.4.5 The CHIMERE pre and post processors

In order to use CHIMERE with the input data sets available for the period of interest, some pre-processor interfaces have been built. These interfaces are able to convert the original data format of some input files in a format readable from CHIMERE using the netCDF [161] libraries.

- Meteo data[2]: provided by ARPA (Regional Agency Prevention and Envitonment) Emilia-Romagna [169] using the LAMI (Local Area Model Italy) model [170].
- Biogenic emissions: hourly biogenic emissions depending of the vegetation and provided by ARPA (Regional Agency Prevention and Envitonment) Emilia-Romagna.
- Anthropic emissions: provided by the National Inventory of Emission [171] published in the year 2003 and disaggregated in space on the domain of interest.
- Boundary Conditions: define the concentration of the pollutants on the domain border. Provided by Prev'AIR [172] from INERIS.

Moreover, a post processor interfaces has been built in order to analyze the concentration of a selected pollutant present in the *out.*[*label*].*nc* file. In the present case the post processor calcualtes the overcoming events regarding the sum of the maxima of the daily 8-hours running average ozone concentration in excess of 120 $\mu g/m^3$ [173].

3.4.6 Grid implementation of CHIMERE

The porting of the CHIMERE multi-scale package onto the Grid environment was performed by making use of the User Intarface (UI) machine available in COMPCHEM. From the UI the user is able to compile and test the code, submit it on the grid environment for execution, control the status of the submitted work and, finally, retrieve the results of the performed calculations. The porting procedure was articulated in several steps.

[2]hourly conditions and emissions

ISSN: 2279–8773

In the first step the code was compiled using the Intel Fortran Compiler®, with the support of the Message Passing Interface (MPI) libraries in order to maximize the performance of the multi pocessor Working Nodes (WNs) present on the segment of the production EGEE grid available to the COMPCHEM VO, and using netCDF libraries.

In the second step the files necessary for the execution are uploaded to the Grid environment on one of the Storage Elements (a remote machine for the date storaging that support, via the gridftp protocol, the data transfer between the machines interconnected into the Grid) which supports the VO (in particular se.grid.unipg.it SE) before submitting the job.

In the third step the script is launched for the job execution. This is a simple bash script that is given below.

```
01 lcg-cp --vo compchem lfn:/grid/compchem/meteo \\
02      file:meteo
03 lcg-cp --vo compchem lfn:/grid/compchem/emi \\
04      file:emi
05 lcg-cp --vo compchem lfn:/grid/compchem/bound \\
06      file:bound
07 ./chimere > chimere-prod.log  2>&1
08 tar -cvzf data.tar.gz *.nc *.log
09 lcg-cr -d se.grid.unipg.it -l lfn:/grid/compchem/data.tar.gz \\
10      file:data.tar.gz
11 exit
```

In lines 01, 03 and 05 of the script the input files needed for the execution are retrieved from the SE and downloaded to the Computing Element (CE) where the calculation is performed; in line 07 the parallel version of CHIMERE is executed; in lines 08 and 09 the files produced by the execution are collected in a single tar file and uploaded to the SE.

In the fouth step, at the end of the simulation, the tar files created by the calculation is directly retrieved from the SE using the lcg-cp command and transferred into the UI machine.

3.4.7 Calculations and results

As already mentioned the calculations were performed using the V200606A version of CHIMERE. The V200606A version of CHIMERE implemented on the Pentium Xeon cluster of the University of Perugia that belongs to the production EGEE Grid infrastructure available to the COMCHEM VO. The calculation ran in parallel on 8 nodes for 124 hours and produced an amount

ISSN: 2279-8773

of 30 GB of output data. As illustrated in Fig. 3.27 the domain of interest is central Italy. Such domain consists of 8000 cells (5x5 Km) with an extention of 500x400 Km^2. The chemical mechanism used is MELCHIOR2 based on a reduced set of 44 chemical species, 120 chemical reaction in gas phase and the ISORROPIA [174] aereosol model. As already mentioned the period chosen for the simulation is the summer of the year 2004 (in particular the one going from the 1st of May 2004 to the 31st of August 2004). This period was chosen because the latest available input data was of the year 2004. The values adopted for the main parameters of the simulation are given in Table 3.12.

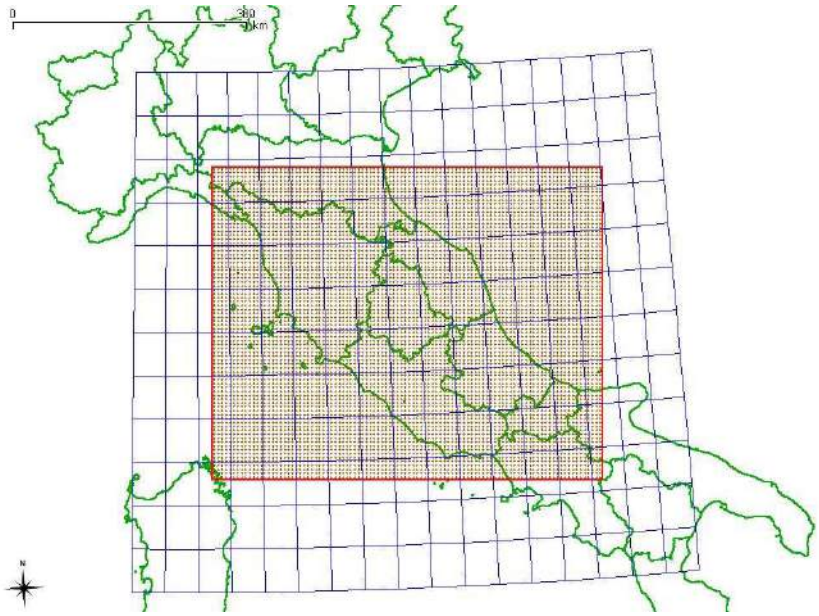

Figure 3.27: Representation of the simulated domain (red dotted grid) and the boundary conditions (blue grid).

In order to make it easier to understand the results of the calculations, an ad hoc visualization tool has been used and the produced images are reported here for discussion.

The Figures 3.28, 3.29, 3.30 and 3.31 show qualitative pictures of, respectively, the temperature at two meters from ground, the O_3 concentration and the biogenic and antropogenic emissions for NO at particular hours of the day on the 1st of July 2004. As shown by the Figures the distribution of temperature and of concentrations are the ones expected on the basis of the orography, the hour of the day and the level of urbanization.

ISSN: 2279-8773

Table 3.12: Main parameters of the CHIMERE middle-Italy simulation.

Version	V200606A
Horiz. Resolution	5x5 Km
Vertical Grid	Sigma P Hybrid
Verical Levels	8
Height cell	43m
Simulated Period	01 May 2004 - 31 August 2004
Elapsed Time	123.43 hours
Number of processors	8
Number of cells	8000
Input file space	10.6 GB
Output file space	30 GB
Gas Chemical Mechanism	MELCHIOR2
Aereosol Model	ISORROPIA
Aereosol species	7
Granulometric classes	6
Output species saved	13
Calculated species	190

Fig. 3.32 shows in a more quantitative fashion the hourly profiles. All the graphs start from the 1st of July and represent the distribution of the O_3 concentration on the 1st day (upper left hand side panel), in the week going from the 1st to the 7th of July (upper right hand side panel) and on the month of July (lower panel), respectively, as extracted from the results of the simulation for the cell of Perugia. The Figure shows a clear weekly ciclicity of the ozone concentration and its strong reduction during the week-end associated to traffic reduction.

The comparison of calculated (solid line) O_3 concentration values (given in μ / m^3) with those measured on-site (Perugia, via Cortonese) for the period ranging from the 1st to the 7th of July 2004 (given in μ / m^3) is given in Fig. 3.33. The two sets of data show an excellent accord. It is however important to notice that the measured results are those directly taken at the location of the monitoring station. They strictly depend on the antropic emissions localized in a well defined area and on the used instrumentation. On the contrary calculated values come from a properly disaggregated National database of the emissions which are mediated on an area of 25Km^2 and on 8 vertical levels within an height of 43m.

ISSN: 2279–8773

3.4.8 Conclusions and future work

The implementation of chemistry and transport models, able to reproduce and explain the complex mechanisms involved in the generation and diffusion of the pollutants in the atmosphere, is essential to master phenomena heavily affecting the quality of human life. A lot of work has been done in this direction on scalar, vector and parallel computers. The work ranges from the development of simple fluidodynamics mathematical models to the assemblage of modern multi-scale eulerian models able to describe the detailed evaluation of pollutants and some reactions in which they are envolved in.

The possibility of implementing one of these models, CHIMERE, on a computational distributed platform (cluster) is shown here not only to be possible and to lead to interesting speed-ups on the EGEE production Grid platform but also to open new perspectives for further evaluation of the code as sketched below:

- extend the simulations to long-term period (year) in order to better compare simulations and measurements;
- optimize the performance of concurrent executions on the Grid;
- improve of the adopted chemical mechanisms and implement new ones;
- use and implement different meteo models in order to compare and/or improve the results.

ISSN: 2279–8773

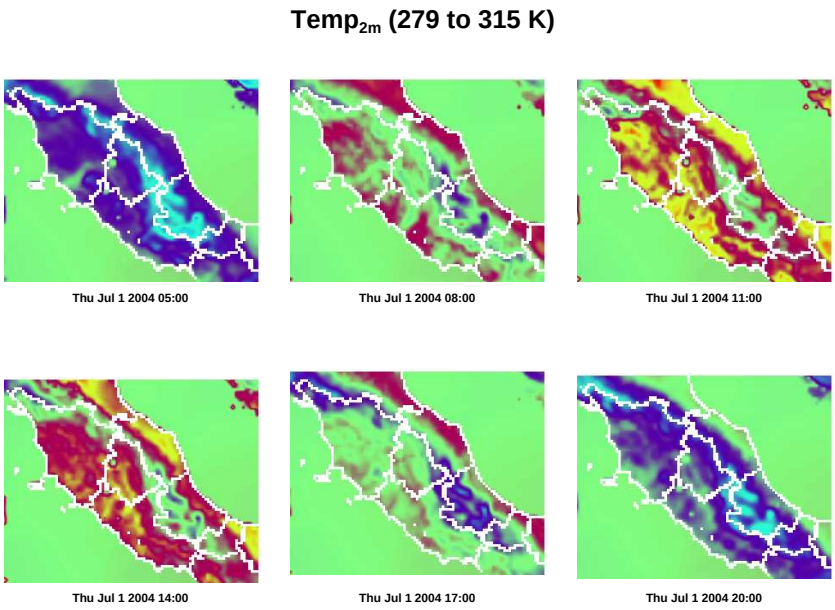

Figure 3.28: Representation of the temperature measured at 2 meters above ground and different hours on the 1st of July 2004. In blue the minimum ($6°C$) and in red the maximum ($42°C$) value.

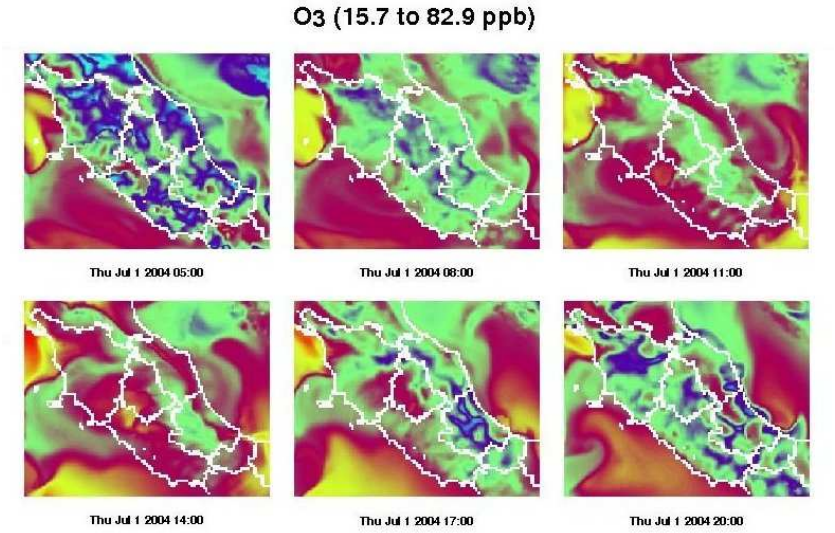

Figure 3.29: Representation of the simulated ozone (O_3) concentration calculated at different hours on the 1st of July 2004. In blue the minimum (15.7 ppb) and in red the maximum (82.9 ppb) values.

ISSN: 2279-8773

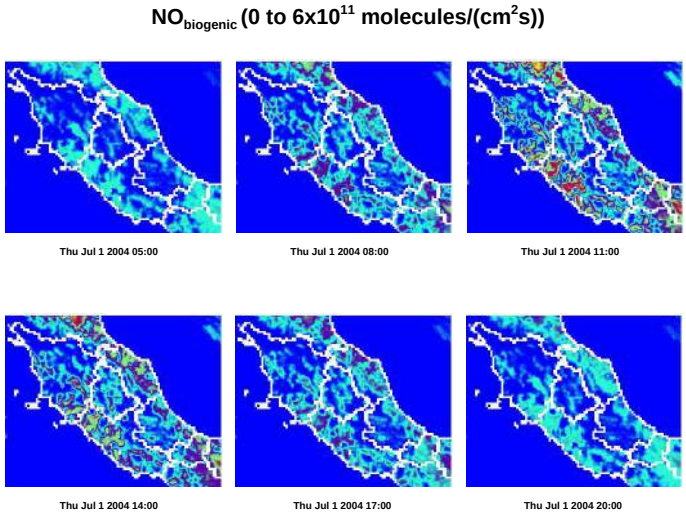

Figure 3.30: Representation of the simulated NO concentration calculated from the biogenic emissions at different hours on the 1st of July 2004. In blue the minimum (0 molecule/(cm^2 s)) and in red the maximum ($6{\cdot}10^{11}$ molecule/(cm^2 s)) value.

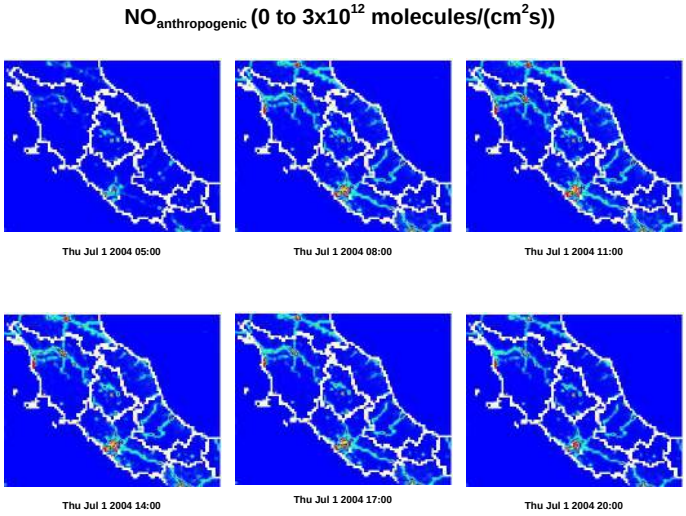

Figure 3.31: Representation of the simulated NO concentration calculated from the antropogenic emissions at different hours on the 1st of July 2004. In blue the minimum (0 molecule/(cm^2 s)) and in red the maximum ($3{\cdot}10^{12}$ molecule/(cm^2 s)) value.

ISSN: 2279-8773

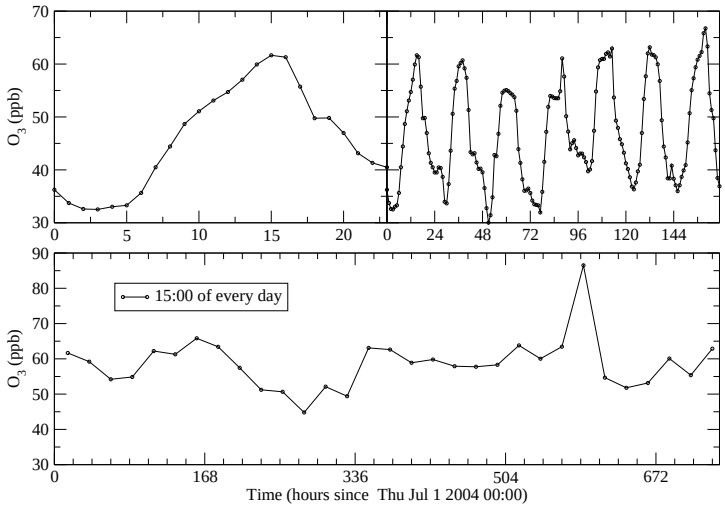

Figure 3.32: The hourly ozone (O_3) concentration (ppb) extracted from the results of the simulation for the cell of Perugia city (via Cortonese): hourly concentration of ozone referred to the 1st of July 2004 (upper left hand side panel); concentration of ozone in the week going from the 1st ot the 7th of July (upper right hand side panel); concentration of ozone in the month of July (lower panel).

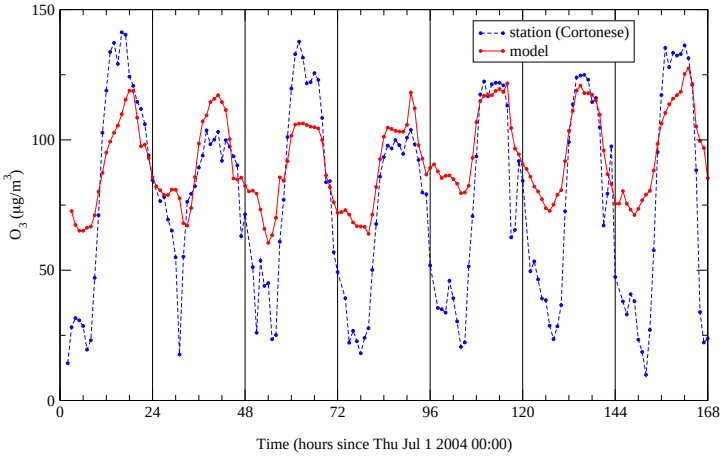

Figure 3.33: Comparison between the calculated (solid line) and measured (dashed line) concentration of O_3 (μg/m^3) from 1st to 7th July 2004 for a localized area in the center of Perugia.

Conclusions

The present thesis is based on the research work developed to investigate complex systems using quantum formalism, Molecular Dynamics and Multi-scale approach. This was tackled using the computational infrastructure of the EGEE Production Grid that support the activities of the Virtual Organization COMPCHEM devoted to deploy and support Computational Chemistry applications.

The research activity involved three different fields as follows:

- Study of the reactive properties of the F + HD and the N + N_2 reactions;
- Study of hydrocarbon bulks of methane and propane using Molecular Dynamics techniques;
- Implementation of multi-scale chemistry and transport models able to reproduce and explain the complex mechanisms involved in the generation and diffusion of the pollutants in the atmosphere.

In the first study the ABC quantum reactive scattering code was ported into the EGEE Grid environment performing, in a portal supported way, the implementation of a distributed concurrent version of the code. In this way it has been easy to reexamine the case of the location of the low collision energy resonance for the F + HD system and to carry out an extended computational campain to evaluate the thermal ratio coefficient of the N + N_2 reaction on different potential energy surfaces.

In the second study the thermodynamic and structural properties of propane and methane bulk system in liquid phase have been estimated using Molecular Dynamics calculations. To this end the parallel version of DL_POLY software package has been ported into the EGEE Grid environment adopting a double level parallel model which allowed a massive run of molecular dynamics simulations.

In the third study the CHIMERE multi-scale chemistry and transport model has been implemented in the portion of the EGEE Grid devoted to

ISSN: 2279-8773

COMPCHEM VO leading to interesting speed-ups but also to open new perspectives for further evaluation of the code. The model, based on a multi-scale eulerian model, is able to describe the detailed evaluation of pollutants and some reactions in which they are involved in to attempt to reproduce the phenomena heavily affecting the quality of uman life.

Prospects

The evolution of computer technologies is at present going beyond the policy of individual large computer centers and is increasingly fostering the networking of distributed computer facilities able to support the development of the so called Grand Challenges in computational science.

In this computational Chemistry plays a key role because of its ability to carry out realistic *a priori* simulations of the multiscale type from the molecular level up. The complexity of this type of multiscale simulations based on an *a priori* description of the reality going from the molecular to the human level requires the exploitation of large computing resources that can only be supplied by the Grid.

The work done in this thesis has evidenced the basic role of the EGEE Production Grid in providing computational infrastructures that enables the scientist to carry out massive computational campaigns in order to exploit the computational features of the implemented codes and to investigate, in a reasonable time, the properties of simple and complex systems. This is going to be in the near future a driving force of innovation and development.

ISSN: 2279-8773

Bibliography

[1] EGEE website: http://public.eu-egee.org

[2] COMPCHEM website: http://compchem.unipg.it

[3] Gervasi, O., Laganà, A.: Simbex a Portal for the a priori simulation of crossed beam experiments. Future Generation Computer Systems 20 (2004) 703-715

[4] Skouteris, D., Castillo, J.F., Manolopulos, D.E.: ABC: a quantum reactive scattering program. Comp. Phys. Comm. 133 (2000) 128-135

[5] P-GRADE Grid Portal: http://portal.p-grade.hu

[6] Sipos, G., Kacsuk, P.: Multi-Grid, Multi-User Workflows in the P-GRADE Portal. Journal

[7] The amendment to the Montreal Protocol (Copenhagen, November 1992): United Nations Environment Programme (UNEP), Ozone Secretariat, http://ozone.unep.org/Ratification_status/copenhagen_amendment.shtml

[8] Habenschuss, A., Narten, A. H.: X-ray diffraction study of liquid propane at 92 and 228K. J. Chem. Phys. 85 (1986) 6022-6026

[9] Jorgensen, W.L., Madura, J.D. and Swenson C.J.: Optimized Intermolecular Potential Functions for Liquid Hydrocarbons. J. Am. Chem. Soc. 106 (1984) 6638-6646

[10] Smith, W., Forester, T.R.: DL_POLY2: a general-purpose parallel molecular dynamics simulation package. J. Mol. Graph. 14 (3) (1996) 136-141

[11] http://www.ambientediritto.it/Legislazione/Mobilita/1996-2000/dir%2096%2062%20CE.htm

ISSN: 2279-8773

[12] The Chimere chemistry-transport model. A multi-scale model for air quality forecasting and simulation. Institute Pierre-Simon Laplace, INERIS, LISA, C.N.R.S. (2004) http://euler.lmd.polytechnique.fr/chimere

[13] http://www.arpa.umbria.it

[14] M. Born and J. R. Oppenheimer, *A*nn. Phys. 85, 457 (1927).

[15] D. R. Hartree, *P*roc. Cam. Phil. Soc., 24, 89 (1928).

[16] J. C. Slater, *P*hys. Rev., 34, 1293 (1929).

[17] http://www.cse.clrc.ac.uk/qcg/gamess-uk/.

[18] G. G. Balint-Kurti, R. N. Dixon and C. C. Marston, *I*nt. Rev. Phys. Chem., 11, 317 (1992).

[19] Á. Vibók and G. G. Balint-Kurti, *J.* Chem. Phys., 96, 7615 (1992).

[20] Á. Vibók and G. G. Balint-Kurti, *J.* Chem. Phys., 96, 8712 (1992).

[21] J. C. Light, I. P. Hamilton, and V. J. Lill, *J.* Chem. Phys., 82, 1400 (1985).

[22] J. V. Lill, G. A. Parker and J. C. Light, *J.* Chem. Phys., 89, 483 (1982).

[23] F. L. Querè and C. Leforestier, *J.* Chem. Phys., 94, 1118 (1991).

[24] F. J. Lin and J. T. Muckerman, *C*omp. Phys. Commun., 63, 538 (1991)

[25] O. Sharaffeddin and J. Z. H. Zhang, *C*hem. Phys. Lett., 204, 190 (1993).

[26] G. C. Corey, J. W. Tromp and D. Lemoine, Numerical Grid Methods and their Applications to Schrödinger Equation (Kluwer Academic, Dordrecht, 1993).

[27] J. T. Muckerman, R. V. Weaver, T. A. B. Kennedy and T. Uzer, Numerical Grid Methods and their Applications to Schrödinger Equation (Kluwer Academic, Dordrecht, 1993).

[28] S. K. Gray and G. G. Balint-Kurti, *J.* Chem. Phys., 108, 950 (1998).

[29] H. Tal-Ezer and R. Kosloff, *J.* Chem. Phys., 81, 3987 (1984).

[30] R. T. Pack and G. A. Parker, *J.* Chem. Phys., 87, 3888 (1987).

ISSN: 2279–8773

[31] J. P. Hansen and I. R. McDonald, *T*heory of simple liquids, 2nd Ed., Academic (1986).

[32] H. Goldstein, *C*lassical Mechanics, Addison-Wesley, Massachusetts (1950).

[33] L. Landau and E. Lifshitz, *M*echanics, Pergamon Press (1961).

[34] B.J. Alder and T.E. Wainwright. *J.* Chem. Phys., 27, 1208 (1957).

[35] L. Verlet, *P*hys. Rev., 165, 201 (1967).

[36] L. Verlet, *P*hys. Rev., 159, 98 (1967).

[37] D. Shepard, A two dimensional interpolation function for irregularly spaced data, *P*roc. 23rd National Conference of the ACM, 517-523 (1968).

[38] J. Ischtwan and M. A. Collins, *J.* Chem. Phys., 100, 8080 (1994).

[39] P. Lancaster and K. Salkauskas, Curve and Surface Fitting, An Introduction (Academic, London, 1986).

[40] F. O. Ellison, *J.* Am. Chem. Soc., 85, 3540 (1963).

[41] P. J. Kuntz, Ber. Bunsenges. *P*hys. Chem., 86, 367 (1982).

[42] A. Laganà, G. O. de Aspuru, and E. Garcia, *J.* Chem. Phys., 108, 3886 (1998).

[43] G. O. de Aspuru and D. C. Clary, *J.* Phys. Chem. A, 102, 9631 (1998).

[44] K. S. Sorbie, J. N. Murrell, *M*ol. Phys., 29, 1387 (1975).

[45] A. Laganà and E. Garcìa *M*ol. Phys., 3, 621 (1985).

[46] W. H. Press, W. T. Vetterling, S. A. Tenkolsky and B. P. Flannery, Numerical Recipes in Fortran, the Art of Scientific Computing (Cambridge, University Press, New York, 1992).

[47] A. Laganà, G. Lendvay: Theory of Chemical Reaction Dynamics. Nato Science Series, Kluwer Academic Publishers, Series II, Vol. 145

[48] S. Rampino, D. Skouteris, A. Laganà and E.Garcia, *L*ecture Notes in Computer Science, 5072 (2008)

[49] S. J. Weiner, P. A. Kollman, D. T. Nguyen and D. A. Case, *J.* Comp. Chem., 7, 230 (1986).

ISSN: 2279–8773

[50] S. J. Weiner, P. A. Kollman, D. A. Case, U. C.Singh, C. Ghio, G. Alagona, S. Jr. Profeta and P. Weiner, *J.* Amer. Chem. Soc., 106, 765 (1984).

[51] S. J. Weiner, P. A. Kollman, D. T. Nguyen and D. A. Case, *J.* Comp. Chem., 7, 230 (1986).

[52] N. L. Allinger, *A*dv. Phys. Org. Chem, 13, 1 (1976).

[53] N. L. Allinger, *J.* Amer. Chem. Soc., 99, 25 (1977).

[54] S. L. Mayo, B. D. Olafson and W. A. Goddard III, *J.* Phys. Chem., 94, 8897 (1990).

[55] K. Rasmussen, S. Balling Engelsen, J. Fabricius and B. Rasmussen, NATO ASI Series C: Mathematical and Physical Sciences, Vol. 406, Kluwer Academic Publishers, Dordrecht, 381-419 (1993).

[56] M. Albertì, A. Aguilar, M. Bartolomei, D. Cappelletti, A. Laganà, J.M. Lucas and F. Pirani, *P*hys. Scr. 78, 058108 (2008).

[57] L. V. Woodcock, *C*hem. Phys. Lett., 10, 257 (1971).

[58] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. Di Nola and J. R. Haak. *J.* Chem. Phys., 81, 3684 (1984).

[59] S. Nosé *J.* Chem. Phys., 81, 511 (1984).

[60] S. Melchionna, G. Ciccotti and B. L. Holian, *M*olec. Phys., 78, 533 (1993).

[61] A. J. Majda: Atmospheric and ocean science provides a rich source of multiscale problems, *A*mer. Math. Soc., (2000).

[62] A. Bensoussan, J.L. Lions, and G. Papanicolaou, Asymptotic Analysis of Periodic Structures, North-Holland, Amsterd-New York, (1978)

[63] J. Kevorkian and J. D. Cole, Multiple Scale and Singular Perturbation Methods, Springer, New York, (1996).

[64] V. P. Malsov and M. V. Fedoriuk, Semiclassical Approximation in Quantum Mechanics, D. Reidel, Dordrecht-Boston, (1981).

[65] J. B. Keller, Geometrical theory of diffraction, *J.* Opt. Soc. Amer. 52 (1962).

ISSN: 2279-8773

[66] A. J. Majda, I. Timofeyev, and E. Vanden-Eijnden, Systematic strategies for stochastic mode reduction in climate, *J.* Atmos. Sci., (2003)

[67] A. J. Chorin, Conditional expectations and renormalization, *M*ultiscale Model. Simul. 1 (2003).

[68] K. Wilson, *P*hys. Rev. B4 (1971).

[69] A. Brandt, Multigrid methods in lattice field computations, *N*uclear Phys. B Proc. Suppl. 26 (1992).

[70] E. B. Tadmor, M. Ortiz, and R. Phillips, Quasicontinuum analysis of defects in crystals, *P*hil. Mag. A73 (1996).

[71] R. B. Bird, C. F. Curtiss, R. C. Armstrong, and O. HASSAGER, Dynamics of Polymeric Liquids, vol. 2: Kinetic Theory, Wiley, New York, (1987).

[72] F. F. Abraham, J. Q. Broughton, N. Berdstein, and E. Kaxiras: Spanning the continuum to quantum length scales in a dynamic simulation of brittle fracture, *E*urophys. Lett. 44 (6) (1998).

[73] E. Vanden-Eijnden: Numerical techniques for multiscale dynamical systems with stochastic effects, *C*omm. Math. Sci. 1 (2003).

[74] C. Theodoropoulos, Y.H. Qian, and I. G. Kevrekidis *P*roc. Nat. Acad. Sci. U.S.A. 97 (18) (2000).

[75] K. Kempf: ENIAC. Electronic Computers Within The Ordnance Corps, The U.S. Army Research Lab. (1961)

[76] F. Baiardi, M. Vanneschi, A. Tomasi, *A*rchitettura dei sistemi di elaborazione, Vol. 1, Franco Angeli (1985).

[77] Gurindar S. Sohi, Scott E. Breach, and T.N. Vijaykumar. Multiscalar processors. *P*roceedings of the 22nd Annual International Symposium on Computer Architecture, pages 414-425 (June 1995).

[78] Salvatore Orlando. Parallelizing and optimizing compiler, 22 August 1994.

[79] Francisca Quintana, Jesus Corbal, Roger Espasa, and Mateo Valero. Adding a vector unit to a superscalar processor. *I*nternational Conference on Supercomputing, pages 1-10, (1999).

ISSN: 2279-8773

[80] C.J.C. Schauble. *V*ector Computing : An Introduction. High Performance Scientific Computing University of Colorado at Boulder (June 1995).

[81] M.J. Flynn, Very high-speed computing systems, *P*roc. of the IEEE, vol. 54, 1901 (1966).

[82] W. Handler, in: Parallel Processing Systems, An Advanced Course, ed. C. U. Press,p. 41 (Evans DJ ed., Cambridge, 1982).

[83] http://www.gigaflop.demon.co.uk/comp/chap7.html.

[84] R. W. Hockney and C. R. Jesshope, Parallel Computers 2 (Adam Hilger/IOP Publishing, Bristol, 1988).

[85] Centro di supercalcolo, Consorzio di 35 UniversitÃ italiane, http://www.cineca.it

[86] I. Foster, *D*esigning and Building Parallel Programs, Addison-Wesley Inc., Argonne National Laboratory (1995).

[87] Sunderam, V.S.: PVM: A framework for parallel distributed computing. Concurrency: practice and experience. 2(4) (1990) 315-339; Geist, G.A., Sunderam, V.S.: Network based concurrent computing on the PVM system. Concurrency: practice and experience. 4(4) (1992) 293-311; Beguelin, A., Dongarra, J., Geist, G.A., Manchek, R., Sunderam, V.S.: A user's guide to PVM Parallel Virtual Machine. Oak Ridge National Laboratory, Tennessee (1992).

[88] Message Passing Interface Forum, *I*nt. J. of Supercomputer Applications 8(3/4) (1994); Smir, M., Otto, S., Huss-Ledermam, S., Walker, D., Dongarra, J.: MPI: The complete reference. MIT Press (1996).

[89] *T*he Grid: Blueprint for a Future Computing Infrastructure, I. Foster, and C. Kesselman Eds., Morgan Kaufmann Publishers, USA (1999).

[90] gLite website: http://glite.web.cern.ch/glite

[91] The Globus Project: http://www.globus.org.

[92] http://firb.miur.it/ .

[93] http://www.grid.it .

[94] COST in Chemistry Action http://www.unil.ch/cost/chem.

[95] O. Gervasi, A. Laganà and F. Sportolari, *J.* Computational Meth. Science and Eng., 2, 377 (2002).

[96] O. Gervasi and A. Laganà, Future Generations of Computer

[97] S. Sato, *B*ull.Chem. SOCJ., 28, 450 (1955).

[98] S. Sato, *J.* Chem. Phys., 23, 592 (1955).

[99] S. Sato, *J.* Chem. Phys., 23, 2465 (1955).

[100] http://egeena4.lal.in2p3.fr/ .

[101] A. Laganà, A. Riganelli and O. Gervasi *L*ecture Notes in Computer Science, 3980, 665-674 (2006).

[102] W. L. Hase, R. J. Duchovic, X. Hu, A. Komornicki, K. F. Lim, D. -h. Lu, G. H. Peslherbe, K. N. Swamy, S. R. Vande Linde, A. Varandas, H. Wang, and R. J. Wolf VENUS96: A General Chemical Dynamics Computer Program, *Q*uantum Chemistry Program Exchange 16, 671 (1996)

[103] D. Skouteris, A. Laganà, G. Capecchi and H.J. Werner *I*nt. J. Quantum Chem., 96, 562-567 (2004).

[104] http://www.univie.ac.at/columbus/

[105] http://www.msg.ameslab.gov/GAMESS/

[106] H. Meyer, U. Manthe, L. Cederbaum, *C*hem. Phys. Lett. 165 (1990) 73-78.

[107] U. Manthe, H. D. Meyer, L. S. Cederbaum, *J.* Chem. Phys. 97 (1992) 3199-3213.

[108] D. M. Ceperley, *R*ev. Mod. Phys. 67 (1995) 279-355.

[109] QDYN is the working group n. 2 of the CMST COST Action D37: http://www.cost.esf.org/index.php?id=189&action _number=D37

[110] GASuC (Grid Application Support Centre): http://www.lpds.sztaki.hu/gasuc of Grid Computing, Vol. 3, No. 3-4, Kluwer Academic Publishers, pp. 221-238, 2006. /pgportal/

[111] Schatz, G.C.: Quantum reactive scattering using hyperspherical coordinates: results for H + H_2 and Cl + HCl. *C*hem. Phys. Lett. 150 (1998) 92-98

ISSN: 2279-8773

[112] R. T. Pack, G. A. Parker, *J. Chem. Phys.*, **87**, 3888 (1987)

[113] Manolopulos, D.E.: An improved log derivative method for inelastic scattering. *J.* Chem. Phys. 85 (1986) 6425-6429

[114] Pack, R.T., Parker, G.A.: Quantum scattering in the three dimentions using hyperspherical (APH) coordinates. Theory. *J.* Chem. Phys. 87 (1987) 3888-3921

[115] Skodje, R.T., Skouteris, D., Manolopulos, D.E., Lee, S.-H, Dong, F., Liu, K.: Resonance-mediated chemical reaction: F + HD to HF + D. *J.* Chem. Phys. 112 (2000) 4536-4552

[116] A. Laganà, E. Garcia, L. Ciccarelli, *J. Phys. Chem.*, **91**, 312 (1987)

[117] A. Laganà, E. Garcia, *J. Chem. Phys.*, **98**, 502 (1994)

[118] A. Laganà, G. Ochoa de Aspuru, E. Garica, *AIAA Journal*, **94**, 1986 (1994)

[119] A. Laganà, G. Ochoa de Aspuru, E. Garica, *Temperature dependence of quasiclassical and quantum rate coefficients for* $N + N_2$, Centro Stampa, Università di Perugia: Perugia (Italy), 1996

[120] I. Armenise, M. Capitelli, R. Celiberto, G. Colonna, C. Gorse, A. Laganà, *Chem. Phys. Lett.*, **227**, 157 (1994)

[121] F. Esposito, M. Capitelli, *Chem. Phys. Lett.*, **302**, 49 (1999)

[122] F. Esposito, I. Armenise, M. Capitelli, *Chem. Phys.*, **331**, 1 (2006)

[123] F. Esposito, M. Capitelli, *Chem. Phys. Lett.*, **418**, 581 (2006)

[124] N. Faginas Lago, A. Laganà, R. Gargano, P. R. P. Barreto, *J. Chem. Phys.*, **125**, 114311 (2006)

[125] C. Petrongolo, *J. Mol. Struct.*, **175**, 215 (1988)

[126] C. Petrongolo, *J. Mol. Struct. (Teochem)*, **202**, 135 (1989)

[127] D. Wang, J. R. Stallcop, W. M. Huo, C. E. Dateo, D. W. Schwenke, H. Partridge, *J. Chem. Phys.*, **118**, 2186 (2003)

[128] E. Garcia, A. Laganà, *J. Chem. Phys.*, **103**, 5410 (1995)

[129] A. Laganà, G. Ochoa de Aspuru, E. Garcia, *J. Phys. Chem.*, **99**, 17139 (1995)

ISSN: 2279-8773

[130] A. Laganà, G. Ochoa de Aspuru, E. Garcia, *J. Chem. Phys.*, **108**, 3886 (1998)

[131] E. Garcia, A. Laganà, *J. Phys. Chem. A*, **101**, 4734 (1997)

[132] E. Garcia, A. Saracibar, A. Laganà, D. Skouteris, *J. Phys. Chem.*, in press

[133] A. Saracibar, PhD Thesis, Sept. 2007

[134] R. A. Back, J. Y. P. Mui, *J. Phys. Chem.*, **66**, 1362 (1962)

[135] S. H. Bauer, S. C. Tsang, *Phys. Fluids*, **6**, 182 (1963)

[136] J. E. Morgan, H. I. Schiff, *Can. J. Chem.*, **41**, 903 (1963)

[137] R. Lyon, *Can. J. Chem.*, **50**, 1437 (1972)

[138] A. Bar-Nun, A. Lifshitz, *J. Chem. Phys.*, **47**, 2878 (1967)

[139] Stark, K., Werner, H.-J.: An accurate multireference configuration interaction (MRCI) calculation of the potential energy surface for the F + H_2 - HF + H reaction. *J.* Chem. Phys. 104 (1996) 6515-

[140] Castillo, J.F., Manolopulos, D.E.: Quantum mechanical angular distributions for the F+HD reaction. *F*araday Discuss. Chem. Soc. 110 (1998) 119-138

[141] Rampino, S., Skouteris, D., Lagana, A., and Garcia, E.: A comparison of the Isotope Effect for the N + N_2 Reaction Calculated on Two Potential Energy Surfaces. *L*ecture Notes in Computer Science 5072 (2008) 1081-1093

[142] http://www.cesga.es

[143] Jorgensen, W.L., Maxwell, D.S., Tirado-Rives, J.: Optimized potential for liquid simulations. J. Am. Chem. Soc. 118 (1996) 11225-11236

[144] Mayo, S.L., Olafson, B.D. and Goddard III, W.A.: DREIDING: A generic force field for molecular simulation. J. Phys. Chem. 94 (1990) 8897-8909

[145] Costantini, A., Laganà, A., Pacifici, L., Gervasi, O.: An alternative distribution model for the Molecular Dynamics study of liquid Propane on a grid platform. Computational Science and Applications (2007) 433-440

ISSN: 2279-8773

[146] Air liquid group website: http://encyclopedia.airliquide.com/encyclopedia.asp?languageid=11&GasID=53

[147] Air liquid group website: http://encyclopedia.airliquide.com/encyclopedia.asp?languageid=11&GasID=41

[148] Egelstaff, P.A.: An Introduction to the Liquid State. Oxford University Press, New York, (1992)

[149] Pinches, M.R.S., Tildesley, D., Smith, W.: The DL_POLY molecular dynamicspackage. Comp. Phys. Comm. 62 (1991) 229

[150] N. Atkins, *A*ir Pollution Dispersion: Ventilation Factor, Lyndon State College.

[151] C.H Bosanquet, J.L. Pearson. The spread of smoke and gases from chimney, *T*rans. Faraday Soc., 32, 1249 (1936).

[152] R. Stozzi: LA MICROMETEOROLOGIA E LA DISPERSIONE DEGLI INQUINANTI IN ARIA. Agenzia per la Protezione dell'Ambiente e per i servizi Tecnici (APAT) (2003)

[153] M.R. Beychok. Fundamentals Of Stack Gas Dispersion (4th Edition ed.), 8, 124 (2005)

[154] Features of Dispersion Models, European Union Joint Research Centre (JRC)

[155] DEGADIS Technical Manual and User's Guide (US EPA's download website)

[156] SLAB User's Manual

[157] HEGADIS Technical Reference Manual

[158] O.G. Sutton. The problem of diffusion in the lower atmosphere, *Q*JRMS, 73, 257 (1947)

[159] D.B. Turner. Workbook of atmospheric dispersion estimates: an introduction to dispersion modeling (2nd Edition ed.). *C*RC Press. (1994)

[160] http://www.lam-mpi.org

[161] http://www.unidata.ucar.edu/software/netcdf

ISSN: 2279-8773

[162] M. Lattuati: (1997). Contribution à l'étude du bilan de l'ozone troposphérique à l'interface de l'Europe et de l'Atlantique Nord: modèlisation lagrangienne et mesures en altitude. Thèse de sciences, Université Paris 6, France (1997).

[163] D. Simpson: Long period modeling of photochemical oxidants in Europe, Calculations for July 1985. *A*tmos. Environ., 26, (1992) 1609-1634.

[164] R. Atkinson, D. L. Baulsch, R. A. Cox, R. F. Hampton, J. A. Kerr, M. J. Rossi and J. Troe, *E*valuated kinetics, photochemical and heterogeneous data. J. Phys. Chem., 26(3), (1997) 521-1012.

[165] W. B. De Moore, S. P. Sandetr, D. M. Golden,R. F. Hampton, M. J. Kurylo, C. J. Howard, A. R. Ravishankara, C. E. Kolb, and M. J. Molina, Chemical kinetics and photochimical data for use in stratospheric modelling evaluation. JPL publication, 94, 26, JPL, Pasadena, US (1994).

[166] B. Aumont, F. Chervier, and S. Laval, Contribution of HONO to the NOx/HOx/O_3 chemistry in the polluted boundary layer. Atmos. Environ., 37 (2003) 487-498.

[167] C. Derognat, M. Beekmann, M. Baeumle, D. Martin, and H. Schmidt, Effect of biogenic volatile organic compound emissions on tropospheric chemistry during the Atmospheric Pollution Over the Paris Area(ESQUIF) campaign in the Ile-de-France region. *J.* of Geophys. Res., 108(D17), (2003) 8560.

[168] W. P. L. Carter, A detail mechanism for the gas-phase atmospheric reactions of organic compounds. *A*tmos. Environ., 24 (1990) 481-518.

[169] http://www.arpa.emr.it

[170] http://nuke.lismec-mci.marche.it/ModelliMeteorologiciLISMEC/LAMIwf/LAMIwfItalia/tabid/122/Default.aspx

[171] http://www.apat.gov.it/site/it-IT/Servizi_per_l'Ambiente/Inventario_delle_Emissioni_in_Atmosfera_(CORINAIR-IPCC)

[172] http://www.prevair.org

[173] Decreto Legislativo 21 maggio 2004, n. 183, "Attuazione della direttiva 2002/3/CE relativa all'ozono nell'aria"

ISSN: 2279-8773

[174] Nenes, A., Pandis, S.N., Pilinis, C.: ISORROPIA: A New Thermodynamic Equilibrium Model for Multiphase Multicomponent Inorganic Aerosols. Aquatic Geochemistry 4 (1998) 123-152

ISSN: 2279-8773