IN PARTNERSHIP WITH:
CNRS

Institut polytechnique de Grenoble

Université Joseph Fourier (Grenoble)

Activity Report 2012

Team NANO-D

Algorithms for Modeling and Simulation of Nanosystems

IN COLLABORATION WITH: Laboratoire Jean Kuntzmann (LJK)

RESEARCH CENTER
Grenoble - Rhône-Alpes

THEME
Computational models and simulation

Table of contents

Team NANO-D

Keywords: Modeling, Simulation, Nanosystems, Adaptive Algorithm, Knowledge-based Algorithms

Creation of the Team: January 01, 2008 .

1. Members

Research Scientists

Sergei Grudinin [CNRS Junior Researcher]
Stephane Redon [Team leader, Inria Junior Researcher]

Engineers

Svetlana Artemova [Research Engineer]
Mael Bosson [Research Engineer]
Jocelyn Gate [Software Engineer]

PhD Student

Petr Popov [Inria PhD Student]

Administrative Assistant

Françoise de Coninck [Assistant]

2. Overall Objectives

2.1. Overview

During the twentieth century, the development of macroscopic engineering has been largely stimulated by progress in numerical design and prototyping: cars, planes, boats, and many other manufactured objects are nowadays designed and tested on computers. Digital prototypes have progressively replaced actual ones, and effective computer-aided engineering tools have helped cut costs and reduce production cycles of these macroscopic systems.

The twenty-first century is most likely to see a similar development at the atomic scale. Indeed, the recent years have seen tremendous progress in nanotechnology - in particular in the ability to control matter at the atomic scale. The nanoscience revolution is already impacting numerous fields, including electronics and semiconductors, textiles, energy, food, drug delivery, chemicals, materials, the automotive industry, aerospace and defense, medical devices and therapeutics, medical diagnostics, etc. According to some estimates, the world market for nanotechnology-related products and services will reach one trillion dollars by 2015. Nano-engineering groups are multiplying throughout the world, both in academia and in the industry: in the USA, the MIT has a “NanoEngineering” research group, Sandia National Laboratories created a “National Institute for Nano Engineering”, to name a few; China founded a “National Center for Nano Engineering” in 2003, etc. Europe is also a significant force in public funding of nanoscience and nanotechnology.

Similar to what has happened with macroscopic engineering, powerful and generic computational tools will be employed to engineer complex nanosystems, through modeling and simulation.

Modeling and simulation of natural or artificial nanosystems is still a challenging problem, however, for at least three reasons: (a) the number of involved atoms may be extremely large (liposomes, proteins, viruses, DNA, cell membrane, etc.); (b) some chemical, physical or biological phenomena have large durations (e.g., the folding of some proteins); and (c) the underlying physico-chemistry of some phenomena can only be described by quantum chemistry (local chemical reactions, isomerizations, metallic atoms, etc.). The large cost of modeling and simulation constitutes a major impediment to the development of nanotechnology.

The NANO-D team aims at developing efficient computational methods for modeling and simulation of complex nanosystems, both natural (e.g., the ATPase engine and other complex molecular mechanisms found in biology) and artificial (e.g., NEMS - Nano Electro-Mechanical Systems).

In particular, the group develops novel multiscale, adaptive modeling and simulation methods, which automatically focus computational resources on the most relevant parts of the nanosystems under study.

2.2. Research axes

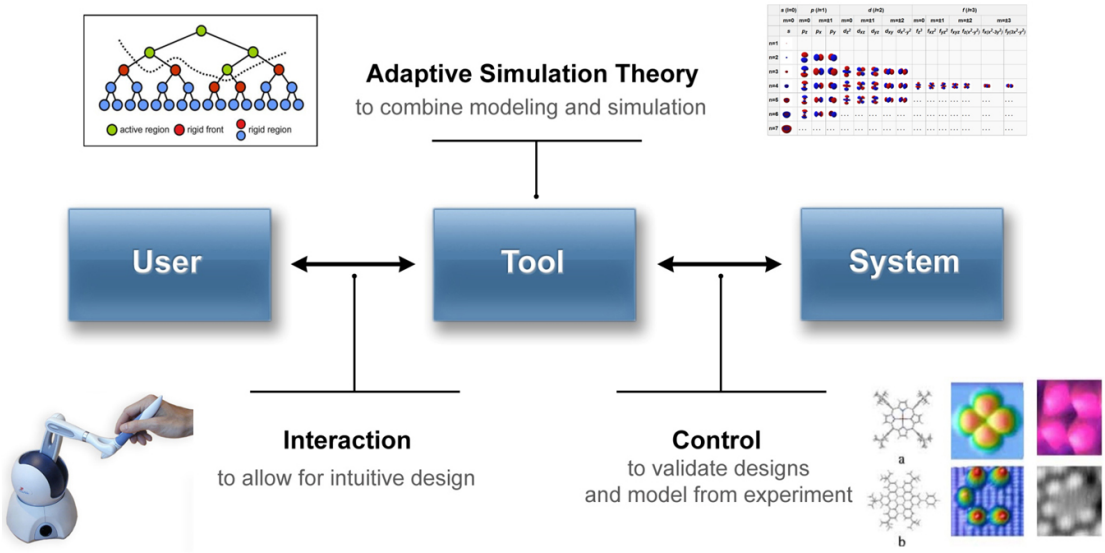

Figure 1. NANO-D's research axes.

The goal of NANO-D is to help current and future designers of nanosystems by developing some of the mathematical and computational foundations of a software application which will run on a desktop computer, and will allow for efficient analysis, design, modeling and simulation of complex nanosystems, whether they are artificial or natural, or a combination of both. For clarity, the research program of the NANO-D group is best introduced by referring to Figure 1. There:

- **User** is any person who wants to study, analyze, design, model, simulate, and control a nanosystem.
- **Tool** is the software application being built on the research performed within NANO-D. This tool will have functionalities that will be similar to those used to design macrosystems (e.g., CATIA, SolidWorks, etc.), but also some others which will be specific to nanoscience.
- **System** is the nanosystem being designed, simulated, controlled, etc., and potentially its environment. The environment can be any system in interaction with the given nanosystem: two electrodes between which a nanotube has been placed, the atomic force microscope which interacts with a nano-wheel, a protein interacting mechanically with a nano-drug or an engineered protein, etc.

This simple diagram makes it clear that a complete, coherent effort towards practical design of nanosystems should be organized in three research axes:

- **Adaptive Simulation Theory**. This first research axis deals with the core algorithms for modeling and simulation, which constitute the heart of the design functionalities of the tool. The main paradigm in this axis is to rely on divide-and-conquer, hierarchical representations to design to adaptive algorithms for modeling and simulating nanosystems.

- **Interaction**. The second group of tasks deals with the interaction of the user with the tool. This is essential, because nanosystems may have complex topologies, kinematics and dynamics, which may make it difficult to edit and model.
- **Control**. The last group of tasks deals with the bidirectional relationships between the simulated nanosystems and the actual ones. Indeed, nanosystems may be characterized through a variety of experimental techniques (e.g., Atomic Force Microscopy, etc.). In order to help verify designs, the tool must thus be able to simulate characterization techniques. Conversely, the tool must be able to take advantage of existing knowledge (both experimental and computational) to help the user design nanosystems.

We believe that each of these three groups of tasks is essential. Fast modeling and simulation algorithms allow for relevant, efficient design ("Adaptive Simulation Theory"). However, software tools with powerful functionalities but which are difficult to use ("Adaptive Simulation Theory" without "Interaction") end up not being used at all. Finally, a tool for modeling and simulating nanosystems which would be "disconnected" from the reality of experimentation ("Adaptive Simulation Theory" without "Control") could only be used for theoretical designs with little assurance of their practicality.

2.3. Highlights of the Year

Stephane Redon has received an ERC Starting Grant in 2012 for his ADAPT project (ADAPT: Theory and algorithms for Adaptive Particle Simulation). The grant is about 1.5 million euros over 5 years.

3. Scientific Foundations

3.1. Overview

The adaptive simulation algorithms we develop typically consist in two main components. The first one determines *which degrees of freedom are simulated* at a give time step, based on the current system's state, as well as user-defined precision or cost thresholds. The second component *incrementally updates the system's state* based on the set of active degrees of freedom. In particular, incremental algorithms update the system's potential energy and forces. This allows the user to smoothly trade between precision and cost.

We detail this approach in two important types of simulations: Cartesian quasi-statics and torsion-angle dynamics. A novel, very general approach for adaptive dynamics simulations of particles — that has a number of important benefits over previous approaches — is mentioned in more detail in Section 6.1.

3.2. Adaptive Cartesian mechanics

In order to focus computations on a specific set of atoms, when performing quasi-static simulations (minimizations), we have developed an adaptive Cartesian mechanics algorithm, which decides which atoms should move at each time step.

In the simplest approach, we simply examine the force applied on each atom. When the norm of the force is above a user-defined threshold, the atom is active. Else the atom position is frozen. A slightly more elaborate version consists in defining the threshold automatically based on the system state (it might be e.g., the average applied force, a percentage of the maximum norm, etc.).

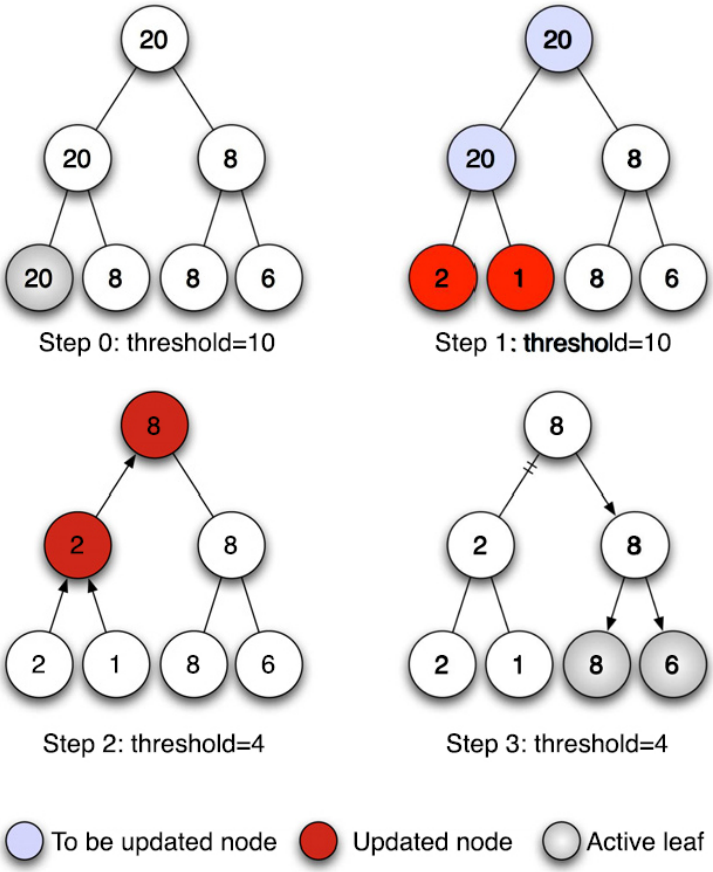

Figure 2. Adaptive Cartesian mechanics.

In order to avoid the linear cost of determining the set of active atoms at each time step, a binary tree is used to represent the system. Each leaf node represents an individual atom, while each internal node represents a set of atoms. Each leaf node stores the norm of the force applied to the corresponding atom. Each non-leaf node stores the maximum of the two force norms of its children, as illustrated in Figure 2. We use two tree passes in order to update tree nodes' values and to determine the new active atoms. In the first, bottom-up pass, force norms are updated in a sub-tree of the binary tree (only some atoms have moved since the previous time step, so only some forces have been updated), starting from the leaves with modified norms, in $O(k^{old}(log(\frac{n}{k^{old}})+1))$ times where k^{old} is the number of active atoms and n the total number of atoms. In the second, top-down pass, the new active atoms (i.e., the atoms with the force norms which are now the largest), are determined in $O(k^{new}(log(\frac{n}{k^{new}})+1))$ times where k^{new} is the new number of active atoms. This process is illustrated in Figure 2 as well.

Precisely, Figure 2 illustrates the procedure to determine the active zone, when the threshold is automatically set to half the largest atomic force norm. In this example, the four leaves correspond to atoms 1 to 4. The value indicated in each leaf node is the norm of the force applied to its corresponding atom. For internal nodes, this value is the maximum of the norms of the forces applied to atoms in the corresponding group. In step 0, the threshold is automatically set to 10. As a result, only atom 1 moves. In step 1, the potential is incrementally updated, and the norms of the forces applied to atoms 1 and 2 are updated. In step 2, the values associated to the tree nodes are incrementally updated through a bottom-up pass that starts from the modified leaf nodes values. Because of this bottom-up update, the adaptive threshold becomes equal to 4. In step 3, the new active atoms are determined through a top-down pass, by visiting only the nodes that have a value larger than the adaptive threshold.

3.3. Adaptive torsion-angle mechanics

In many situations, it is preferable to represent molecular systems as articulated bodies, and perform so-called *torsion-angle* mechanics. This may be to allow for larger time step sizes in a simulation, or because the user wants to focus to e.g., protein backbone deformations.

We have also developed an adaptive mechanics algorithm in the case of torsion-angle representations. In this case, a molecular system is recursively defined as the assembly of *two* molecular systems connected by a *joint* (when connecting two subassemblies which belong to the same molecule) or, more generally, by a *rigid body transform* (to assemble several molecules).

As in the Cartesian mechanics case, the complete molecular system is thus also represented by a binary tree, in which leaves are rigid bodies (a rigid body can be a single atom), internal nodes represent both sub-assemblies and connections between sub-assemblies, and the root represents the complete molecular system (see Figure 3 on the right, which shows an assembly tree associated to a short polyalanin). This hierarchical representation handles any branched molecule or groups of molecules, since the connections between two sub-molecular systems can be a rigid body transformation. In this representation, the positions of atoms are thus represented as superimposed rigid transformations: the position of any atom is obtained from the position of the whole set, to which is "added" the transformation from the complete set to the sub-set the atom belongs to, and so on until we reach the leaf node representing the atom. Similarly, the atomic motions are superimposed rigid motions.

Our adaptive framework relies on two essential components. First, we associate a hierarchical set of reference frames to the assembly tree. Precisely, each node is associated to a local reference frame, in which all dynamical coefficients are expressed. This allows us to avoid updating these coefficients when a sub-assembly moves rigidly. Second, we have demonstrated that it is possible to determine a priori, at each time step, the set of joints which have the largest accelerations. Precisely, when going down the tree to compute joint accelerations, we are able to compute the weighted sum of the (squared) norms of joint accelerations in a sub-assembly C before computing joint accelerations themselves:

$$\mathcal{A}(C) = (\mathbf{f}^C)^T \Psi^C \mathbf{f}^C + (\mathbf{f}^C)^T \mathbf{p}^C + \eta^C, \tag{1}$$

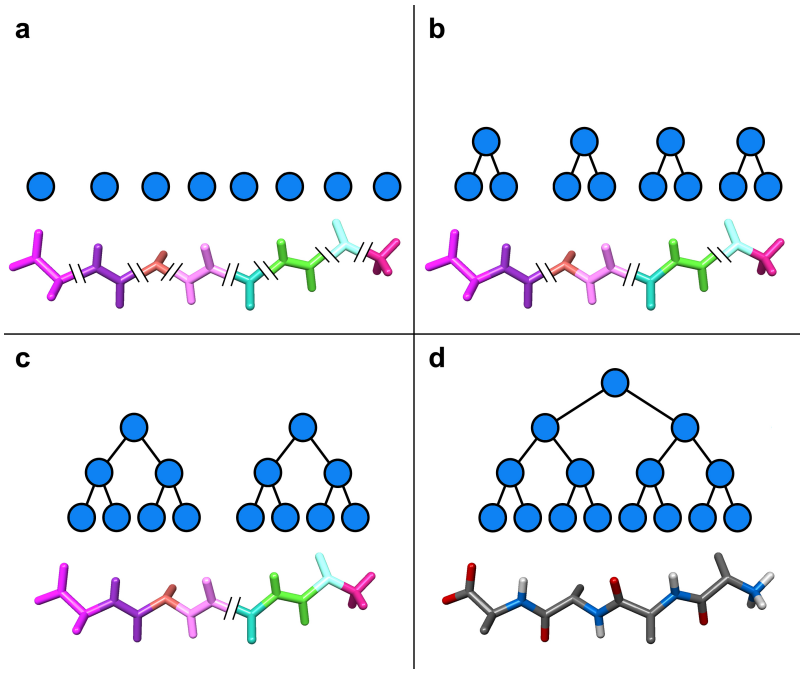

Figure 3. The assembly tree associated to a short polyalanin.

where the right part is a quadratic form of the spatial forces applied on the "handles" of node C. This allows us to cull away those sub-assemblies with (relatively) lower internal accelerations, and focus on the most mobile joints. Thus, at each time step, we can thus predict the set of joints with highest accelerations without computing all accelerations, and we simulate only a sub-tree of the assembly tree (the green nodes in the assembly tree, as in the figure above), based on an user-defined error threshold or computation time constraints. This sub-tree is called the active region, and may change at each time step.

We have exploited these two characteristics - hierarchical coordinate systems and adaptive motion refinement - to develop data structures and algorithms which enable adaptive molecular mechanics. The key observation in our approach is the following: all coefficients which only depend on relative atomic positions do not have to be updated when these relative positions do not change. We can thus store in each node of the assembly tree partial system states which hold information relative only to the node itself.

Precisely, each time step involves the following operations:

1. Adaptive acceleration update
    1. Determination of the acceleration update region: we determine the acceleration update region, i.e., the subset of nodes of the full articulated body which matter the most according to the acceleration metric, as indicated above. The union of the previous active region and the acceleration update region is the transient active region, i.e., the region temporarily considered as the active region.
    2. Joint accelerations projection: the acceleration is projected on the reduced motion space defined by the transient active region (to ensure that joint accelerations are consistent with both motion constraints and applied forces).
2. Adaptive velocity update

1. Determination of the new active region: we update the joint velocities and the velocity metric values of the nodes in the transient active region. We then determine the set of nodes which are considered to be important according to the velocity metric (which is similar to the acceleration metric). This set becomes the new active region.
2. Joint velocities projection: if one or more nodes become inactive due to the update of the active region, we determine a set of impulses that we must apply to the transient hybrid body to perform the rigidification of these nodes. This amounts to projecting joint velocities to the reduced motion space defined by the new active region.

3. Adaptive position update
   1. Position update: we update joint positions based on non-zero joint velocities in the active region.
   2. State update: once joint positions have been updated, we update the rest of the system's state: inverse inertias, acceleration metric coefficients, partial neighbor lists, partial force tables, etc.

Again, each of these steps involves a limited sub-tree of the assembly tree, which enables a fine control of the compromise between computation time and precision.

We have showed that our adaptive approach allows for a number of applications, some of which that were not possible for classical methods when using low-end desktop workstations. Indeed, by selecting a sufficiently small number of simultaneously active degrees of freedom, it becomes possible to perform interactive structural modifications of complex molecular systems.

4. Application Domains

4.1. Overview

NANO-D is *a priori* concerned with all applications domains involving atomistic representations, including chemistry, physics, electronics, material science, biology, etc.

Historically, though, our first applications have been in biology, as the next two sections detail. Thanks to the development of algorithms to efficiently simulate reactive force fields, as well as to perform interactive quantum mechanical calculations, however, we now have the possibility to address problems in chemistry, and physics.

4.2. Structural Biology

Structural biology is a branch of molecular biology, biochemistry, and biophysics concerned with the molecular structure of biological macromolecules, especially proteins and nucleic acids. Structural biology studies how these macromolecules acquire the structures they have, and how alterations in their structures affect their function. The methods that structural biologists use to determine the structure typically involve measurements on vast numbers of identical molecules at the same time, such as X-Ray crystallography, NMR, cryo-electron microscopy, etc. In many cases these methods do not directly provide the structural answer, therefore new combinations of methods and modeling techniques are often required to advance further.

We develop a set of tools that help biologists to model structural features and motifs not resolved experimentally and to understand the function of different structural fragments.

- Symmetry is a frequent structural trait in molecular systems. For example, most of the water-soluble and membrane proteins found in living cells are composed of symmetrical subunits, and nearly all structural proteins form long oligomeric chains of identical subunits. Only a limited number of symmetry groups is allowed in crystallography, and thus, in many cases the native macromolecular conformation is not present on high-resolution X-ray structures. Therefore, to understand the realistic macromolecular packing, modeling techniques are required.

- Many biological experiments are rather costly and time-demanding. For instance, the complexity of mutagenesis experiments grows exponentially with the number of mutations tried simultaneously. In other experiments, many candidates are tried to obtain a desired function. For example, about 250,000 candidates were tested for the recently discovered antibiotic Platensimycin. Therefore, there is a vast need in advance modeling techniques that can predict interactions and foresee the function of new structures.
- Structure of many macromolecules is still unknown. For other complexes, it is known only partially. Thus, software tools and new algorithms are needed by biologists to model missing structural fragments or predict the structure of those molecule, where there is no experimental structural information available.

4.3. Pharmaceutics and Drug Design

Drug design is the inventive process of finding new medications based on the knowledge of the biological target. The drug is most commonly an organic small molecule which activates or inhibits the function of a biomolecule such as a protein, which in turn results in a therapeutic benefit to the patient. In the most basic sense, drug design involves design of small molecules that are complementary in shape and charge to the biomolecular target to which they interact and therefore will bind to it. Drug design frequently relies on computer modeling techniques. This type of modeling is often referred to as computer-aided drug design.

Structure-based drug design attempts to use the structure of proteins as a basis for designing new ligands by applying accepted principles of molecular recognition. The basic assumption underlying structure-based drug design is that a good ligand molecule should bind tightly to its target. Thus, one of the most important principles for designing or obtaining potential new ligands is to predict the binding affinity of a certain ligand to its target and use it as a criterion for selection.

We develop new methods to estimate the binding affinity using an approximation to the binding free energy. This approximation is assumed to depend on various structural characteristics of a representative set of native complexes with their structure solved to a high resolution. We study and verify different structural characteristics, such as radial distribution functions, and their affect on the binding free energy approximation.

4.4. Nano-engineering

The magazine Science has recently featured a paper demonstrating an example of DNA nanotechnology, where DNA strands are stacked together through programmable self-assembly. In February 2007, the cover of Nature Nanotechnology showed a “nano-wheel” composed of a few atoms only. Several nanosystems have already been demonstrated, including a wheelbarrow molecule, a nano-car and a Morse molecule, etc. Typically, these nanosystems are designed in part *via* quantum mechanics calculations, such as the semi-empirical ASED+ calculation technique.

Of course, not all small systems that currently fall under the label “nano” have mechanical, electronic, optical properties similar to the examples given above. Furthermore, current construction capabilities lack behind some of the theoretical designs which have been proposed. However, the trend is clearly for adding more and more functionality to nanosystems. While designing nanosystems is still very much an art mostly performed by physicists, chemists and biologists in labs throughout the world, there is absolutely no doubt that fundamental engineering practices will progressively emerge, and that these practices will be turned into quantitative rules and methods. Similar to what has happened with macroscopic engineering, powerful and generic software will then be employed to engineer complex nanosystems.

We have recently shown that our incremental and adaptive algorithms allow us to easily edit and model complex shapes, such as a nanotube (Fig. 4) and the “nano-pillow” below (Fig. 5).

5. Software

5.1. SAMSON

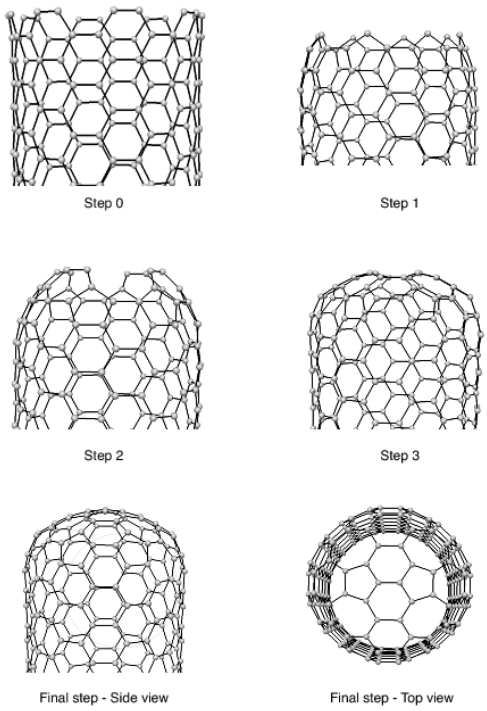

Figure 4. Snapshots of a nanotube capping process with the adaptive interactive modeler. Thanks to the adaptive methodology, this operation can be done in a few minutes.

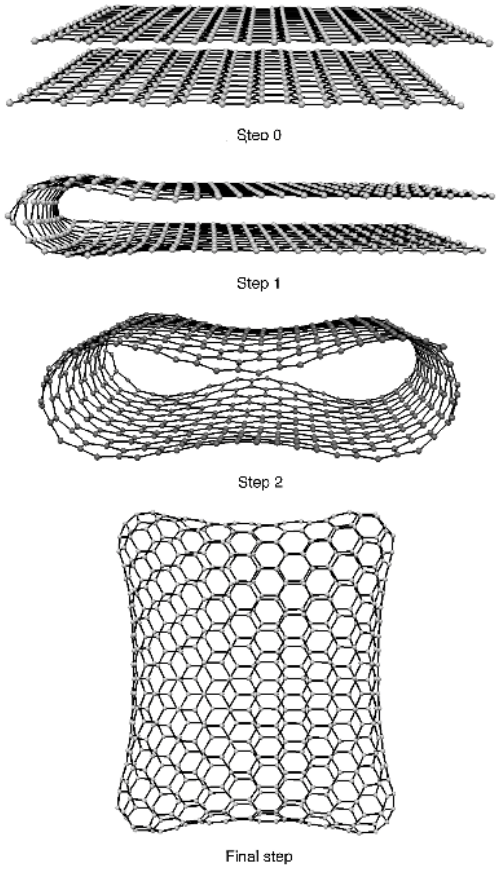

Figure 5. Different steps to prototype a "nano-pillow" with the adaptive interactive modeler.

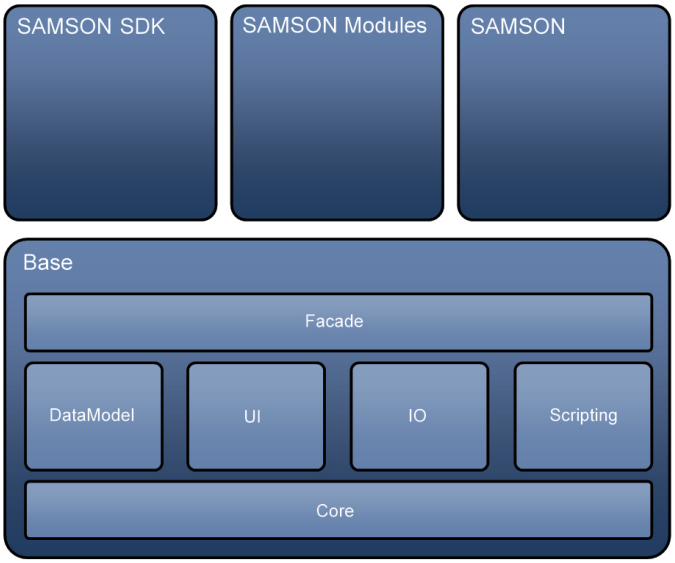

Figure 6. SAMSON's architecture.

A major objective of NANO-D is to try and integrate a variety of adaptive algorithms into a unified framework. As a result, NANO-D is developing SAMSON (Software for Adaptive Modeling and Simulation Of Nanosystems), a software platform aimed at including all developments from the group, in particular those described below.

The objective is to make SAMSON a generic application for computer-aided design of nanosystems, similar to existing applications for macrosystem prototyping (CATIA, SolidWorks, etc.).

The current architecture of SAMSON is visible in Figure 6. The code is organized into four main parts: a) the Base (in which "Core" contains, in particular, the heart of the adaptive algorithms: signaling mechanisms specifically designed for SAMSON), b) the Software Development Kit (SDK: a subset of the base that will be provided to module developers), c) Modules, and d) the SAMSON application itself.

Similar to the concept of Mathematica *toolboxes*, for example, the goal has been to make it possible to personalize the user interface of SAMSON for potentially many distinct applications. For example, we may want to personalize the interface of SAMSON for crystallography, drug design, protein folding, electronics, material science, nano-engineering, etc., by loading different modules at startup, depending on the user application domain.

6. New Results

6.1. Adaptively Restrained Particle Simulations

Participants: Svetlana Artemova, Stephane Redon.

Last year, we have introduced a novel, general approach to speed up particle simulations that we call Adaptively Restrained Particle Simulations (ARPS). This year we continued working on this approach. The obtained results have been published in Physical Review Letters [3], and the patent describing the theoretical basis and the algorithms for the numerical realization of ARPS has been deposited.

Particle simulations are widely used in physics, chemistry, biology [13], [14], and even computer graphics [9], and faster simulations (in particular ARPS) may result in progress on many challenging problems, e.g., protein folding, diffusion across bio-membranes, fracture in metals, ion implantation, etc.

ARPS rely on an adaptively restrained (AR) Hamiltonian used to describe a system of N particles:

$$H_{AR}(\mathbf{q},\mathbf{p}) = \frac{1}{2}\mathbf{p}^T\Phi(\mathbf{q},\mathbf{p})\mathbf{p} + V(\mathbf{q}).$$

This Hamiltonian has an unusual inverse inertia matrix $\Phi(\mathbf{q},\mathbf{p})$, which is made a general function of phase-space coordinates. The precise form of this matrix can be chosen according to the system under study and the problem stated.

We have proposed a particular (diagonal) form of the inverse inertia matrix for the simulations in Cartesian coordinates. In this case, Φ adaptively switches on and off positional degrees of freedom of individual particles while letting particle momenta evolve. The decision whether the particle is restrained or not depends on the particle's momentum, and, precisely, on it's kinetic energy. Two user-defined thresholds regulate the amount of simplification of the particle's motion. When a module of a particle's momentum becomes small enough (without necessarily becoming zero), the particle completely stops moving. Even when a particle is fully restrained, though, its momentum may continue to change, and its kinetic energy might become large enough again for the particle to resume moving. In general, ARPS restrain and release particles repeatedly over time.

This approach has numerous advantages: (a) it is mathematically grounded and is able to produce long, stable simulations; (b) it does not require modifications to the simulated interaction potential, so that any suitable existing force-field can be directly used with ARPS; (c) under frequently-used assumptions on the interaction potential, ARPS make it possible to reduce the number of forces that have to be updated at each time step, which may significantly speed up simulations; (d) when performing constant-energy simulations, ARPS allow users to finely and continuously trade between precision and computational cost, and rapidly obtain approximate trajectories; (e) the trade-off between precision and cost may be chosen for each particle independently, so that users may arbitrarily focus ARPS on specific regions of the simulated system (e.g., a polymer in a solvent); (f) most important, when performing Adaptively Restrained Molecular Dynamics (ARMD) in the canonical (NVT) ensemble, correct static equilibrium properties can be computed.

We have demonstrated the advantages of ARPS on several numerical experiments. For example, a planar collision cascade study in Fig. 7 shows how ARPS make it possible to smoothly trade between precision and speed of the simulation. Reference simulations were derived from the usual Hamiltonian $H(\mathbf{q},\mathbf{p}) = \frac{1}{2}\,\mathbf{p}^T\mathbf{M}^{-1}\mathbf{p} + V(\mathbf{q})$.

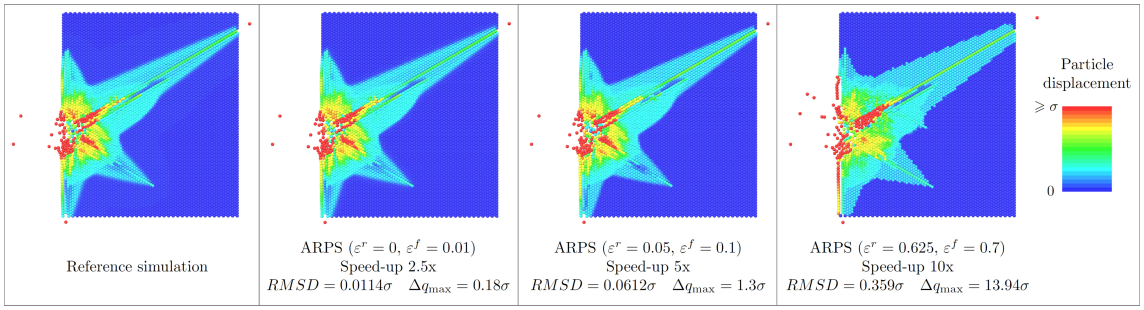

Figure 7. Simulating a collision cascade with controlled precision. Adaptively restrained simulations allow us to smoothly trade between precision and speed. Even for large speed-ups (up to 10x) the features of the shock are extremely well preserved.

6.2. Hierarchical Adaptively Restrained Particle Simulations

Participants: Svetlana Artemova, Stephane Redon.

It has been shown that algorithms relying on hierarchical representations of molecular systems may accelerate molecular simulations: for example, divide-and-conquer approach for simulations in internal coordinates [10], [11], adaptive algorithms for dynamics of articulated bodies [15], algorithms for neighbor search for system with symmetries [12] or for large rigid molecules [8].

Therefore, we were interested in combining hierarchically-based algorithms with Adaptively Restrained Particle Simulations (ARPS). Precisely, as with classical ARPS, we have considered the adaptively restrained (AR) Hamiltonian:

$$H_{AR}(\mathbf{q},\mathbf{p}) = \frac{1}{2}\mathbf{p}^T\Phi(\mathbf{q},\mathbf{p})\mathbf{p} + V(\mathbf{q}),$$

but we have introduced a different form of the inverse inertia matrix $\Phi(\mathbf{q},\mathbf{p})$. In this case, again, positional degrees of freedom are adaptively switched on and off during the simulation, but, these are *relative* positional degrees of freedom in the system, and not the positional degrees of freedom of individual particles. Precisely, particles are grouped together into rigid bodies according to the tree representation and released repeatedly during the simulation. We call this approach hierarchical Adaptively Restrained Particle Simulations (hierarchical ARPS).

We have performed several numerical experiments to illustrate this new approach. For example, in Fig. 8 we present the planar collision cascade study.

For hierarchical AR simulations, obtained results depend on the tree representation of the system: for the results demonstrated in Fig. 8 the tree was constructed in a top-down manner by recursive dividing of the system in halves and, therefore, the squares of different levels are being activated by the shock.

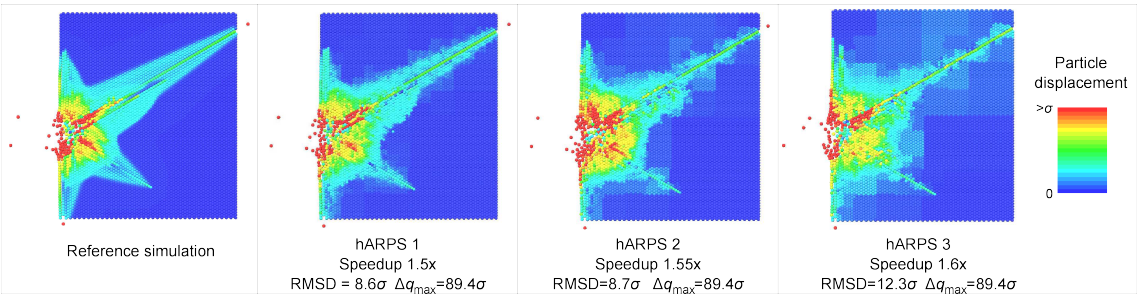

Figure 8. Simulating a collision cascade with controlled precision. Hierarchical adaptively restrained simulations allow us to smoothly trade between precision and speed. The main features of the shock are preserved. The binary tree representation was constructed top-down.

To clearly demonstrate the effect of the tree, we provide the results for the same four simulations with another tree built in a bottom-up manner by grouping the particles pairwise according to their sequence number (they were enumerated, first, along the y-axis, vertically, and then, along the x-axis, horizontally). These results are shown in Fig. 9, and are rather different from those in Fig. 8: vertical lines are being activated when the central part of the plane is reached by the shock.

The patent reporting the principles and the algorithms used to implement hierarchical ARPS has been deposited.

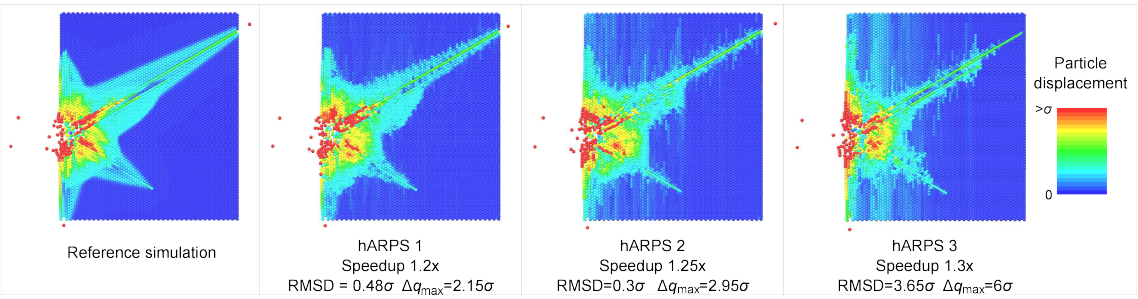

Figure 9. Simulating a collision cascade with controlled precision. Hierarchical adaptively restrained simulations allow us to smoothly trade between precision and speed. The main features of the shock are preserved. The binary tree representation was constructed bottom-up.

6.3. Interactive quantum chemistry

Participants: Mael Bosson, Caroline Richard, Antoine Plet, Sergei Grudinin, Stephane Redon.

Interactive simulation tools allow users to take advantage of their knowledge and intuition to understand physical properties and prototype new devices. To accurately describe bond breaking, bond formation, charge transfer or other electronic phenomena, interactive simulation should ideally be based on quantum mechanics. However, solving quantum chemistry models at interactive rates is a challenging task. Thanks to the algorithms developed in the group, SAMSON is the first software to propose interactive quantum chemistry.

A first contribution allows for interactive quantum chemistry with systems up to a few hundred atoms [6]. The method is based on a divide-and-conquer (D&C) approach. The D&C technique subdivides the system into many subsystems (a–h on the Figure 10). Each of them involves a diagonalisation at each time step. To treat larger systems, we introduce a new algorithm: Block-Adaptive Quantum Mechanics (BAQM) [5] from the combination of two new components.

- **Block-adaptive Cartesian mechanics**

  By freezing atomic positions in some subsystems (d–h on the Figure 10) (with atoms in blue), we may avoid updating some eigenproblems. The Block-adaptive Cartesian mechanics component takes advantage of this to control the simulation cost by adaptively adjusting the number of diagonalisations, based on the forces applied to the atoms. Only the subsystems with the largest applied forces are allowed to have mobile atoms.

- **Adaptive reduced-basis quantum mechanics**

  Solving even just one of the subsystem's eigenproblem may be too costly to achieve interactive rates. The Adaptive reduced-basis quantum mechanics component projects the equation in an adaptive reduced basis composed of low-energy eigenvectors that have been computed at a previous time step, to benefit from temporal coherence between successive eigenproblems (subsystems (b) and (c) with atoms in black and white on the Figure 10). We use a simple distance to decide on the fly when to automatically update the reduced basis during the simulation (subsystem (a) with atoms in red on the Figure 10).

We demonstrated that BAQM may accelerate geometry optimization for several atomic systems. Indeed, each step is solved significantly faster by constraining some nuclei and electrons, and, by focusing computational resources on the most active parts of the system, we obtain a faster potential energy descent. The proposed BAQM approach also allows for interactive rates with many atomic systems.

6.4. Molecular Docking

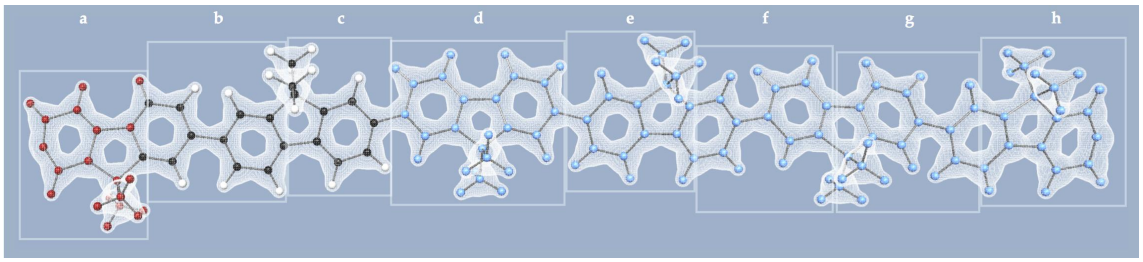

Figure 10. ***Interactive editing of a polyflurorene molecule with the BAQM algorithm***

6.4.1. Development of a new Knowledge-Based Potential for Protein-Ligand Interactions

Participants: Sergei Grudinin, Georgy Cheremovskiy.

Macromolecular complexes formed by proteins with small molecules (ligands) play an important role in many biological processes such as signal transduction, cell regulation, etc. Experimental methods for determining the structures of molecular complexes have a very high cost and still involve many difficulties. Therefore, computational methods, such as molecular docking, are typically used for predicting binding modes and affinities, which are essential to understand molecular interaction mechanisms and design new drugs.

Databases containing three-dimensional protein-ligand structures determined by experimental techniques grow very rapidly. In 2011, the PDB (Protein Data Bank) contained about 70,000 of protein structures, with almost 8,000 structures of protein-ligand complexes having refined binding affinity data. The CSD (Cambridge Structural Database), a database for small molecules, contained about 500,000 entries at the beginning of 2012. Thus, we believe that computational tools based on statistical information extracted from three-dimensional structures of protein-ligand complexes will play an ever more increasing role in the functional study of proteins as well as in structure-based drug design and other fields.

We proposed and validated a new statistical method that predicts binding modes and affinities of protein-ligand complexes. To do so, we have developed a novel machine-learning-based approach. Precisely, we have formulated a new optimization problem with 30,000 unknowns, whose solution is a scoring function. We trained the scoring function on 6,000 structures of protein-ligand complexes of high accuracy from the PDB database. Despite the very high dimensionality of the optimization problem, we manage to solve it on a desktop computer in just a few hours.

Our scoring function has three major applications in drug-design:

- Docking: determination of the binding site of a ligand bound to a protein.
- Ranking: identifying a set of ligands with the highest binding affinity for the given protein target by screening a large ligand database.
- Binding constants prediction: prediction of the absolute value of the binding constant of a protein-ligand complex.

The success rates of our method rank it among the top three methods currently available. Thus, we believe that our scoring function is the first one that performs well in all three major applications in drug-design.

6.4.2. DockTrina

Participants: Sergei Grudinin, Petr Popov.

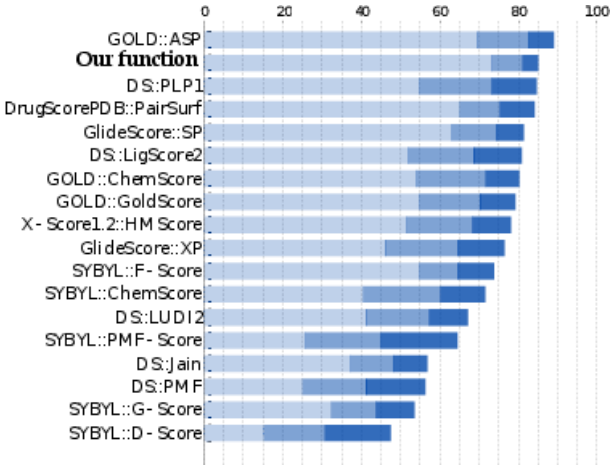

Figure 11. Comparison of the success rates of scoring functions when the best-scored binding pose differs from the true one by RMSD < 1.0 Å (light bars), < 2.0 Å (darker bars) or < 3.0 Å (the darkest bars), respectively. Scoring functions are ranked by success rates when the ligand binding pose is found within RMSD < 3.0 Å.

We derived analytical formulas for fast evaluation of the Root-Mean-Square-Deviation (RMSD) between rigid protein structures. This work resulted in a RMSD library containing algorithms to calculate the RMSD between two proteins in constant time. Based on this library we introduced an efficient algorithm to predict triangular protein structures and implemented it into the DockTrina software. We collected bound benchmarks of 220 protein trimers with and without symmetry properties from the Protein Data Bank and demonstrated the superiority of DockTrina over standard combinatorial algorithms aimed at predicting nonsymmetrical protein trimers.

6.4.3. Machine Learning for Structural Biology

Participants: Sergei Grudinin, Petr Popov, Mathias Louboutin.

We developed a new formulation of the machine learning optimization problem to predict protein–protein interactions. We implemented several optimization strategies, both in *d*ual and *p*rimal. We studied the effect of different types of loss-functions on the quality of the prediction. We also tested the efficiency of three descent algorithms, Nesterov descent, gradient descent, and stochastic descent. We demonstrated that generally, primal optimization is faster compared to dual optimization. In the primal, Nesterov descent has a better convergence compared to the gradient descent. Finally, stochastic algorithms often provide a better convergence compared to deterministic algorithms. All the studied algorithms were implemented as a stand-alone library.

6.5. Software Engineering

Participants: Jocelyn Gate, Stephane Redon.

We have continued the development of SAMSON, our open-architecture platform for modeling and simulation of nanosystems (SAMSON: Software for Adaptive Modeling and Simulation Of Nanosystems). The interface has been improved:

- The visualization of the data graph has been improved. Users may now drag and drop models and parts between layers, as well as directly drag and drop files into SAMSON.
- The undo/redo stack can now be visualized.
- We have begun to work on selection and highlighting.

The software engineering process has been improved as well, in particular to help base and modules developers:

- We have reorganized the file hierarchy so that modules can have associated data.
- We have developed a system to build SAMSON automatically on virtual machines (e.g., ubuntu 12.04 32bit, ubuntu 12.04 64 bit, fedora 17 32 bit, etc.).
- Tools have been created to let modules developers easily write new modules.
- We have begun to develop a mechanism to make it easy to install and update SAMSON automatically.

We have also developed several *SAMSON apps* to test various concepts, including scripting, manipulating molecules with haptic feedback, etc. Figure 12 shows the current user interface of SAMSON.

We have deposited the first version of SAMSON's code base at the APP ("Agence de Protection des Programmes").

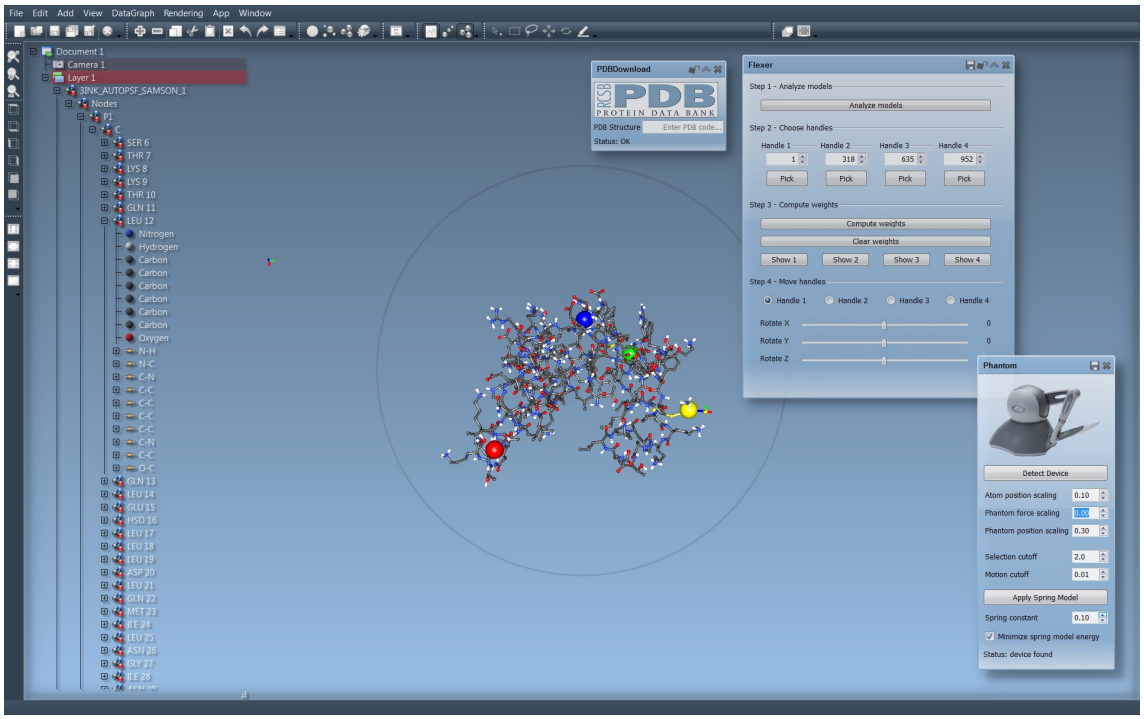

Figure 12. The current user interface of SAMSON, showing an app to download molecules directly from the Protein Data Bank, an app to deform molecules, and an app for haptic interaction. The data graph on the left shows the hierarchical structure of the data graph.

7. Partnerships and Cooperations

7.1. Regional Initiatives

We have obtained a regional grant for a PhD student (ARC 2012). The PhD student will be co-advised by Jean-François Mehaut (LIG, Grenoble) and Benjamin Bouvier (IBCP, Lyon), and will develop algorithms for parallel adaptive molecular dynamics simulations.

7.2. National Initiatives

7.2.1. ANR

In 2012, NANO-D received funding from four ANR programs:

- **ANR JCJC**: 340,000 Euros over three years (2011-2014). This grant has been provided to S. Redon by the French Research Agency for being a finalist in the ERC Starting Grant 2009 call, and is for two PhD students and an engineer.
- **ANR MN**: 180,000 Euros over four years (2011-2015). This project, coordinated by NANO-D (S. Grudinin), gathers biologists and computer scientists from three research groups: Dave Ritchie at LORIA, Valentin Gordeliy at IBS (total grant: 360,000 Euros).
- **ANR PIRIBio**: 25,000 Euros over four years (2010-2013). We are participating in this project coordinated by Michel Vivaudou at IBS, with Serge Crouzy at CEA/LCBM and Frank Fieschi at IBS.
- **ANR COSINUS**: 85,000 Euros over four years (2009-2012). This project, coordinated by NANO-D (S. Redon), gathers physicists, biologists and computer scientists from five research groups: Xavier Bouju and Christian Joachim at CEMES, Martin J. Field at IBS, Serge Crouzy at CEA/LCBM, Thierry Deutsch and Frederic Lancon at CEA/SP2M (total grant: 380,000 Euros).

7.2.2. PEPS

Sergei Grudinin participates in the Cryo-CA PEPS project. Cryo-CA (Computational algorithms for biomolecular structure determination by cryo-electron microscopy) is a 2-years project, supported by the Projets Exploratoires Pluridisciplinaires (PEPS) program in the panel Bio-Maths-Info provided by CNRS (French National Centre for Scientific Research). The project started on the 01/09/2012. Its main goal is to develop computational algorithms for cryo-electron microscopy (cryo-EM).

The partners of the Cryo-CA project are: Inria Nancy / Team Orpailleur (David Ritchie); Inria Grenoble / Team NANO-D (Sergei Grudinin); and INSERM IGBMC/ Team Integrated structural Biology (Annick Dejaegere, Patrick Schultz, and Benjamin Schwarz).

The main scientific aim of this cross-disciplinary project is to develop computational algorithms to help experimentalists and molecular modelers to solve more rapidly and accurately the structures of macromolecular complexes using cryo-electron microscopy (cryo- EM) and integrative structural biomolecular modeling techniques. More specifically, this PEPS initiative aims to address two important challenges in single particle cryo-EM, namely particle picking and multi-dimensional structure fitting. In the longer term, a further driving aim of this project is to develop strong collaborations amongst the participating teams to position ourselves for a larger project proposal to ANR or ERC.

7.3. European Initiatives

7.3.1. FP7 Projects

7.3.1.1. ADAPT

Title: Theory and algorithms for adaptive particle simulation

Type: IDEAS

Instrument: ERC Starting Grant

Duration: September 2012 - August 2017

Principal Investigator: Stephane Redon

Coordinator: Inria (France)

7.4. International Research Visitors

7.4.1. Internships

Georgy CHEREMOVSKIY (from Jul 2012 until Oct 2012)

Subject: Development of Orientation-Dependent Potential Function for Computational Drug Design

Institution: Moscow Institute for Physics and Technology (Russian Federation)

8. Dissemination

8.1. Teaching - Supervision

8.1.1. Teaching

Licence : Stephane Redon, “Introduction to computer science”, INF311 and INF321, 80h, Ecole Polytechnique, Paris, France

8.1.2. Supervision

PhD : Svetlana Artemova, Adaptive Algorithms for molecular simulation, Grenoble University, May 30, 2012, Stephane Redon

PhD : Mael Bosson, Adaptive algorithms for computational chemistry and interactive modeling, Grenoble University, October 19, 2012, Brigitte Bidegaray and Stephane Redon

PhD in progress : Petr Popov, Computational methods for protein structure prediction, November 2011, Sergei Grudinin

8.2. Participation to conferences, seminars

- S. Grudinin and P. Popov attended "Journees du GdR BiMGdR Bim", Paris (January 20 2012).
- S. Grudinin and P. Popov participated in a workshop "Exploring Protein Interactions through Theory and Experiments", Lausanne (September 24-26 2012).
- S. Grudinin gave a talk titled "Fast Fitting of Atomic Structures into Cryo-EM Density Maps Using Hermite Orthogonal Functions" in a workshop "Computational Challenges in Structural Biology", Strasbourg (November 14-15 2012).
- P. Popov participated in a workshop "Computational Challenges in Structural Biology", Strasbourg (November 14-15 2012).

9. Bibliography

Publications of the year

Doctoral Dissertations and Habilitation Theses

[1] S. ARTEMOVA. *Adaptive Algorithms for molecular simulation*, Grenoble University, 2012.

[2] M. BOSSON. *Adaptive algorithms for computational chemistry and interactive modeling*, Grenoble University, 2012.

Articles in International Peer-Reviewed Journals

[3] S. ARTEMOVA, S. REDON. *Adaptively Restrained Particle Simulations*, in "Physical Review Letters", November 2012, vol. 109, 190201 [*DOI :* 10.1103/PHYSREVLETT.109.190201], http://hal.inria.fr/hal-00756121.

[4] M. BOSSON, S. GRUDININ, X. BOUJU, S. REDON. *Interactive physically-based structural modeling of hydrocarbon systems*, in "Journal of Computational Physics", March 2012, vol. 231, n° 6, p. 2581-2598 [*DOI :* 10.1016/J.JCP.2011.12.006], http://hal.inria.fr/hal-00755542.

[5] M. BOSSON, S. GRUDININ, S. REDON. *Block-Adaptive Quantum Mechanics: An Adaptive Divide-and-Conquer Approach to Interactive Quantum Chemistry*, in "Journal of Computational Chemistry", October 2012 [*DOI :* 10.1002/JCC.23157], http://hal.inria.fr/hal-00755521.

[6] M. BOSSON, C. RICHARD, A. PLET, S. GRUDININ, S. REDON. *Interactive quantum chemistry: A divide-and-conquer ASED-MO method*, in "Journal of Computational Chemistry", March 2012, vol. 33, p. 779-790 [*DOI :* 10.1002/JCC.22905], http://hal.inria.fr/hal-00755498.

[7] I. Y. GUSHCHIN, V. I. GORDELIY, S. GRUDININ. *A Novel Dimerization Interface of Cyclic Nucleotide Binding Domain, which is Disrupted in Presence of cAMP: Implications for CNG Channels Gating*, in "Journal of Molecular Modeling", September 2012, vol. 18, n° 9, p. 4053-4060 [*DOI :* 10.1007/S00894-012-1404-5], http://hal.inria.fr/hal-00766015.

References in notes

[8] S. ARTEMOVA, S. GRUDININ, S. REDON. *A comparison of neighbor search algorithms for large rigid molecules*, in "Journal of Computational Chemistry", 2011, vol. 32, n° 13, p. 2865–2877.

[9] O. ETZMUSS, J. GROSS, W. STRASSER. *Deriving a particle system from continuum mechanics for the animation of deformable objects*, in "Visualization and Computer Graphics, IEEE Transactions on", 2003, vol. 9, n° 4, p. 538–550.

[10] R. FEATHERSTONE. *A divide-and-conquer articulated-body algorithm for parallel O(log (n)) calculation of rigid-body dynamics. Part 1: Basic algorithm*, in "The International Journal of Robotics Research", 1999, vol. 18, n° 9, p. 867–875.

[11] R. FEATHERSTONE. *A divide-and-conquer articulated-body algorithm for parallel O(log (n)) calculation of rigid-body dynamics. Part 2: Trees, loops, and accuracy*, in "The International Journal of Robotics Research", 1999, vol. 18, n° 9, p. 876–892.

[12] S. GRUDININ, S. REDON. *Practical modeling of molecular systems with symmetries*, in "Journal of Computational Chemistry", 2010, vol. 31, n° 9, p. 1799–1814.

[13] R. HOCKNEY, J. EASTWOOD. *Computer simulation using particles*, Institute of Physics, 1992.

[14] J. MONAGHAN. *Smoothed particle hydrodynamics*, in "Reports on Progress in Physics", 2005, vol. 68, 1703.

[15] S. REDON, N. GALOPPO, M. LIN. *Adaptive dynamics of articulated bodies*, in "ACM Transactions on Graphics (TOG)", ACM, 2005, vol. 24, n° 3, p. 936–945.