INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team ScAlApplix

High Performance Schemes and Algorithms for Complex Scientific Applications

Futurs

THEME NUM

Activity Report

2006

Table of contents

1. Team

ScAlApplix is a joint project of INRIA Futurs, LaBRI (Laboratoire Bordelais de Recherche en Informatique – CNRS UMR 5800, University of Bordeaux 1 and ENSEIRB) and MAB (Laboratoire de Mathématiques Appliquées de Bordeaux – CNRS UMR 5466, Universities of Bordeaux 1 and Bordeaux 2). ScAlApplix has been created on the first of november 2002 (http://www.labri.fr/scalapplix).

Team Leader

Jean Roman [INRIA Research scientist (on secondment), Professor at ENSEIRB (École Nationale Supérieure d'Électronique, d'Informatique, et de Radiocommunications de Bordeaux), LaBRI, HdR]

Staff members from INRIA

Josy Baron [Administrative assistant]
Olivier Coulaud [Research director]
Pascal Hénon [Research scientist]
Mario Ricchiuto [Research scientist]

Staff member from University of Bordeaux 1

Rémi Abgrall [Professor at MatMéca, Institut Universitaire de France (until september 2006), MAB, HdR]
Christophe Berthon [Senior assistant professor at MatMéca, MAB, HdR]
Cécile Dobrzynski-Proust [Assistant professor at MatMéca (since september 2006), MAB]
Aurélien Esnard [Assistant professor (since september 2006), LaBRI]
Abdou Guermouche [Assistant professor, LaBRI]
Boniface Nkonga [Senior assistant professor at MatMéca, MAB, HdR]
Pierre Ramet [Assistant professor at IUT, LaBRI]

Staff member from University Louis Pasteur of Strasbourg

Guillaume Latu [INRIA research scientist (on partial secondment since september 2006), Assistant professor, LSIIT]

Staff member from ENSEIRB

Olivier Beaumont [INRIA research scientist (on partial secondment), Senior assistant professor, LaBRI, HdR]
François Pellegrini [Assistant professor, LaBRI]

Post-doctoral fellow

Miroslaw Korzeniowski [funding from INRIA since october 2006, LaBRI]
Mikaël Papin [funding from INRIA until may 31th 2006, MAB]

Ph. D. students

Guillaume Anciaux [funding from INRIA and CEA, LaBRI]
Aubin Bellue [funding from CEA, since september 2006, CELIA]
Marie Billaud [funding from CEA, since september 2006, MAB]
Benjamin Braconnier [funding from CEA, MAB]
Cédric Chevalier [funding from CNRS and Conseil Régional d'Aquitaine, LaBRI]
Mathieu Faverge [funding from ANR CIGC NUMASIS, since october 2006, LaBRI]
Pierre Fortin [funding from CNRS and Conseil Régional d'Aquitaine, LaBRI]
Jérémie Gaidamour [funding from CNRS and Conseil Régional d'Aquitaine, since october 2006, LaBRI]
Adam Larat [funding from ADIGMA (since october 2006), MAB]
Bruno Lathuilère [funding from EDF, since november 2006, LaBRI]
Vincent Perrier [funding from Ministry of Research and Technology, MAB]
Nicolas Richart [funding from INRIA and Conseil Régional d'Aquitaine, LaBRI]
Cédric Tavé [funding from CNES, MAB]
Gaël Tessier [funding from Conseil Régional d'Aquitaine, LaBRI]

Technical staff

Guilhem Caramel [since january 1st until december 31th 2006, funding from EDF]
Mikaël Lutz [since september 1st until decembre 31th 2006, funding from CEA]

Mathieu Souchaud [since november 1st 2006, funding from INRIA]

Research scientist (partners)

Patrick Amestoy [Professor, ENSEEIHT (École Nationale Supérieure d'Électrotechnique, d'Électronique, d'Informatique, et d'Hydraulique de Toulouse), IRIT (Institut de Recherche en Informatique de Toulouse, UMR 5505), HdR]

Research scientist (visitors)

Chi-Wang Shu [Professor, Brown University, USA, 1 month in may 2006, HdR]
Jiri Dobes [Professor, CTU, Tcheck Republic, 1 month in september 2006, HdR]

Student interns

Jérémie Gaidamour [since february 1st until september 15th 2006, funding from INRIA, LaBRI]
Adam Larat [since february 1st until june 30th 2006, MatMéca]
Bruno Lathuilière [since march 1st until september 15th 2006, funding from EDF, LaBRI]
Andreas Bollerman [since january 1st until june 30th 2006, MatMéca]

2. Overall Objectives

2.1. Overall Objectives

The purpose of the `ScAlApplix` project is to analyze and solve scientific computation problems arising from complex research and industrial applications and involving scaling. These applications are characterized by the fact that they require enormous computing power, on the order of tens or hundreds of teraflops, and that they handle huge amounts of data. Solving these kinds of problems requires a multidisciplinary approach concerning both applied mathematics and computer science. In applied mathematics, it is essentially the field of numerical schemes that is concerned. In computer science, parallel computing and the design of high-performance codes to be executed on today's major computing platforms are concerned (parallel computers organized as a large network of SMP nodes, GRID platforms).

Through this approach, `ScAlApplix` intends to contribute to all steps in the line that goes from the design of new high-performance, more robust and more precise, numerical schemes to the optimized implementation of algorithms and codes for the simulation of *physical* (fluid mechanics, inert and reactive flows, multimaterial and multiphase flows), *biological* (molecular dynamics simulations) and *environmental* (host-parasite systems in population dynamics) phenomena that are by nature multiscale and multiphysics.

Another domain we are currently investigating is the development of distributed environments for coupling numerical codes and for steering interactively numerical simulations. The computational steering is an effort to make the typical simulation work-flow (modeling, computing, analyzing) more efficient, by providing on-line visualization and interactive steering over the on-going computational processes. On-line visualization appears very useful to monitor and detect possible errors in long-running applications, and interactive steering allows the researcher to alter simulation parameters on-the-fly and immediately receive feedback on their effects. Thus, scientists gain a better insight in the simulation regarding to the cause-effect relationship and can better grasp the complexity of the underlying models.

3. Scientific Foundations

3.1. Numerical schemes for fluid mechanics

Keywords: *complex physical models*, *unstructured meshes*, *upwind schemes*.

A large number of industrial problems can be translated into fluid mechanics ones. They may be coupled with one or more physical models. An example is provided by aeroelastic problems, which have been studied in details by other INRIA teams. Another example is given by flows in pipelines where the fluid (a mixture of air–water–gas) has no very well-understood physical properties. One may also consider problems in aeroacoustics, which become more and more important in everyday life. In some occasions, one needs specific numerical tools because fluids have exotic equation of states, or because the amount of computation becomes huge, as for unsteady flows. Another situation where specific tools are needed is when one is interested in very specific quantities, such as the lift and drag of an airfoil, a situation where commercial tools can only provide a very crude answer.

It is a fact that there are many commercial codes. They allow users to consider some of these examples, but the quality of the solutions is far from being optimal. Moreover, the numerical tools of these codes are often not the most recent ones. An example is the noise generated by vortices crossing through a shock wave. It is, up to our knowledge, even out of reach of the most recent technologies because the numerical resources that would necessitate such simulations are tremendous ! In the same spirit, the simulation of a 3D compressible mixing layer in a complex geometry is also out of reach because very different temporal and physical scales need to be captured, thus we need to invents specific algorithms for that purpose.

In order to reach efficient simulation of complex physical problems, we are working on some fundamental aspects of the numerical analysis of non linear hyperbolic problems. Our goal is to develop schemes that can adapt to the modern computer architectures. More precisely, we are working on a class of numerical schemes specifically tuned for unstructured and hybrid meshes. They have the most possible compact stencil that is compatible with the expected order of accuracy. The order of accuracy typically ranges from two to four. Since the stencil is compact, the implementation on parallel machines becomes simple. The price to pay is that the scheme is necessarily implicit. However, the compactness of the scheme enables to use the high performance parallel linear algebra tools developed by the team for the lowest order version of these schemes. The high order versions of these schemes, that are still under development, will lead to new scientific problems at the border between numerical analysis and computer science. In parallel to these fundemental aspects, we also work on adapting more classical numerical tools to complex physical problems such as those encountered in interface flows, turbulent or multiphase flows.

Within a few years, we expect to be able to consider the physical problems that are now difficult to compute thanks to the know–how coming from our research on compact distribution schemes and the daily discussions with specialists of computer science and scientific computing. These problems range from aeroacoustic to multiphysics problems, such as the one mentionned above.

3.2. Schemes and algorithms for computational chemistry

Keywords: *molecular dynamics*, *multiscale schemes*, *parallelism and distributed computing*.

Due to the increase of available computer power, new applications such as reaction paths, free energy computations, biomolecular dynamics simulations or failure material simulations are now commonly performed by chemists. These computations simulate systems up to several thousands of atoms, for large time scales up to several nanoseconds. The larger the simulation is, the smaller the computational cost of the potential driving the phenomena is, resulting in low precision results. To achieve realistic results, simulations need to include the environment surrounding the molecules, such as water and membranes, resulting in system sizes up to about several hundred thousands of atoms. Furthermore, simulating the aggregation of proteins, which is critical for biologists studying viruses, requires models of up to one million atoms, with a simulation time up to one millisecond. This implies that atomistic simulations must be speeded up by several orders of magnitude. To obtain this speed, numerical and parallel algorithms must be improved, as well as their implementations on distributed or parallel architectures.

We are currently focusing on several aspects of these problems.

First, we try to improve models and algorithms. To do this, we decrease the complexity of classical algorithms by introducing new approximations in the algorithms, in the model (this is the trick of linear scaling methods like the divide-and-conquer method), and by proposing new algorithms.

Second, we apply multiscale methods to decrease the number of atoms that are considered at the finest level (electronic or atomistic). To do this, we introduce a coarser model like continuum media to take into account the electrostatic effect of the environment, or an elasticity model for crystals. The difficulty here is to build an efficient scheme which couples the two different scales without any loss of precision.

Finally, an efficient implementation is necessary to reach the desired level of performance. For instance, we can rewrite our algorithms in the form of block computations, in order to use efficient computational routines such as BLAS vector-matrix operations, or to implement accurate load balancing strategies.

Another domain we are currently investigating is the development of parallel and distributed environments for coupling numerical codes, and for interactively steering numerical simulations in particular in the context of molecular dynamics.

3.3. Algorithms and high-performance solvers

Keywords: *fast multipole methods*, *high-performance computing*, *parallel sparse linear algebra*.

3.3.1. High-performance direct solvers for distributed clusters

Solving large sparse systems $Ax = b$ of linear equations is a crucial and time-consuming step, arising in many scientific and engineering applications. Consequently, many parallel techniques for sparse matrix factorization have been studied and implemented.

We have started this research by working on the parallelization of an industrial code for structural mechanics, which was a 2D and 3D finite element code and non linear in time. This computational finite element code solves plasticity problems (or thermo-plasticity problems, possibly coupled with large displacements). Since the matrices of these systems are very ill-conditioned, classical iterative methods are not an issue. Therefore, to obtain an industrial software tool that must be robust and versatile, high-performance sparse direct solvers are mandatory, and parallelism is then necessary for reasons of memory capabilities and acceptable solving time. Moreover, in order to solve efficiently 3D problems with more than 10 millions of unkowns, which is now a reachable challenge with new SMP supercomputers, we must achieve a good time scalability and control memory overhead.

In the ScAlApplix project, we focused first on the block partitioning and scheduling problem for high performance sparse LDL^T or LL^T parallel factorization without dynamic pivoting for large sparse symmetric positive definite systems. Our strategy is suitable for non-symmetric sparse matrices with symmetric pattern, and for general distributed heterogeneous architectures whose computation and communication performances are predictable in advance.

Research about high performance sparse direct solvers is carried on in collaboration with P. Amestoy (ENSEEIHT – IRIT) and J.-Y. L'Excellent (INRIA Rhône-Alpes), and has led to software developments (see section 5.4, 5.5, 5.8) and to industrial contracts with CEA (Commissariat à l'Energie Atomique).

3.3.2. High-performance iterative solvers

In addition to the project activities on direct solvers, we also study some robust preconditioning algorithms for iterative methods. The goal of these studies is to overcome the huge memory consumption inherent to the direct solvers in order to solve 3D dimensional problems of huge size (several millions of unknowns). Our studies focus on the building of generic parallel preconditioners based on ILU factorizations. The classical ILU preconditioners use scalar algorithms that do not exploit well the CPU power and are difficult to parallelize. Our work aims at finding some unknown orderings and partitionings that lead to a dense block structure of the incomplete factors. Then, based on the block pattern, some efficient parallel blockwise algorithms can be devised to build robust preconditioners that are also able to exploit the full capabilities of the modern high-performance computers.

We study two approaches:

- the first approach consists in building block ILU(k) preconditioners. The main idea is to adapt the classical ILU(k) factorization in order to reuse the algorithmic components that have been developed for direct methods. In this case, the ordering we use is the same than in the direct factorization and a dense block pattern (i.e. a partition of the unknowns) is obtained using an algorithm that lumps columns having few differences in their non zeros pattern. We have adapted the parallel direct solver chain in order to deal with the incomplete block factors defined by this process. Thus the preconditioner computation benefits from the breakthroughts made by the direct solver techniques studied in `PaStiX` (sections 5.5 and 6.4).
- the second approach we recently developed is based on the Schur complement approach. In this case, we use a partition of the adjacency graph of the system matrix into a set of small subdomains with overlap. The interior of these subdomains are treated by a direct method. Solving the whole system is then equivalent to solve the Schur complement system on the interface between the subdomains (this system has a much smaller dimension). We use the hierarchical interface decomposition (HID) that has been developed in `PHIDAL` to reorder and partition this system. Indeed, the HID gives a natural dense block structure of the Schur complement. Based on this partition, we define some efficient block preconditioners that allow the use of BLAS routines and a high degree of parallelism thanks to the HID properties. All these algorithms are implemented in a new library named `HIPS`. `HIPS` contains the `PHIDAL` library (HID ordering and partitioning) and proposes some extensions and new algorithms (multilevel functionalities, hybrid direct-iterative approach) to the former `PHIDAL` library. Details can be found in sections 5.6 and 6.4.

3.3.3. Fast Multipole Methods

In most of scientific computing applications considered nowadays as computational challenges like biological systems, astrophysic or electromagnetism, the introduction of hierarchical methods based on an octree structure has dramatically reduced the amount of computation needed to simulate those systems for a given error tolerance.

Among these methods, the Fast Multipole Method (FMM) allows the computation of the interactions in, for example, a molecular dynamics system of N particles in $O(N)$ time, against $O(N^2)$ with a direct approach. The extension of these methods and their efficient implementation on current parallel architectures is still a critical issue. Moreover the use of periodic boundary conditions, or of duplications of the system in 2 out of 3 space dimensions, just as well as the use of higher approximations for integral equations are also still relevant.

In order to treat biological systems of up to several millions of atoms, these methods must be integrated in the `QC++` platform (see section 5.7). They can be used in the three (quantum, molecular and continuum) models for atom-atom interactions in quantum or molecular mechanics, atom-surface interactions for the coupling between continuum and the other models, and also for fast matrix-vector products in the iterative solving of the linear system given by the integral formulation of the continuum method. Moreover, the significant experience achieved by the `Scotch` and `PaStiX` projects (see section 5.8 and 5.5) will be useful in order to develop efficient implementations of the FMM methods on parallel clusters of SMP nodes.

3.4. Parallel algorithms for heterogeneous platforms

Keywords: *computational grids*, *parallel algorithms*, *scheduling*.

Recently, a lot of work has been devoted to computational grids. Such computing architectures differ from usual parallel platforms in terms of heterogeneity (of both processing and communication resources) and scale (use of large distance network links with high latencies). Such platforms are usually not dedicated to one application, and therefore, security and fault tolerance problems also arise. In our works, we do not consider the security and fault tolerance problems, but rather concentrate on additional difficulties arising from the heterogeneity and the dynamicity (in terms of resource performances rather than topology) of such platforms.

Our goal is to design efficient scheduling algorithms for heterogeneous and non-dedicated platforms. Scheduling computational tasks or collective communications on a given set of processors is a key issue for high-performance computing. The traditional objective of scheduling algorithms is makespan minimization: given a task graph and a set of computing resources, find a mapping of the tasks onto the processors, and order the execution of the tasks so that (i) task precedence constraints are satisfied; (ii) resource constraints are satisfied; and (iii) a minimum schedule length is provided. However, makespan minimization turned out to be NP-complete problem in most practical situations and the advent of more heterogeneous architectural platforms is likely to even increase the computational complexity of the process of mapping applications to machines.

Many of the works presented in section 6.5 have been done in collaboration with the GRAAL project (INRIA Rhône-Alpes) during the PhD thesis of Loris Marchal (defended 10/06) which is co-directed by Olivier Beaumont and Yves Robert (GRAAL project).

3.5. Computational steering for parallel and distributed numerical simulations

Keywords: *computational steering*, *coupling*, *interaction*, *numerical simulation*, *scientific visualization*.

Thanks to the constant evolution of computational capacity, numerical simulations are becoming more and more complex; it is not uncommon to couple different models in different distributed codes running on supercomputer or heterogeneous grids. Nowadays, such simulations are typically running in batch mode, and the analysis of the results is then performed on a local workstation as a post-processing step, which implies to preliminary collect all the simulation output files. In most research fields, the post-processing step is realized thanks to 3D scientific visualization techniques. In the batch approach, there is a lack of control over the in-progress computations, and it might drastically decrease the profitability of the computational resources (repeated tests with different input files separated by excessively long waiting periods).

For years, the scientific computing community has expressed the need of new computational steering tools. The computational steering is an alternative approach to the typical simulation work-flow of performing computation and visualization sequentially. It mainly consists in coupling a remote simulation with a graphics system through the network in order to provide scientists with on-line visualization and interactive steering. On-line visualization appears very useful to monitor the evolution of the simulation by rendering the current results. It also allows us to validate the simulation codes and to detect conceptual or programming errors before the completion of a long-running application. Interactive steering allows the researcher to change the parameters of the simulation without stopping it. As the on-line visualization provides an immediate visual feedback on the effect of a parameter change, the scientist gains additional insight in the simulation, regarding to the cause-effect relationship. Such a tool might help the scientist to better grasp the complexity of the underlying models and to drive more rapidly the simulation *into the right direction*. Basically, a computational steering environment can be defined as a communication infrastructure, coupling a simulation with a remote user interface, called steering system. This interface usually provides on-line visualization and user interaction. Over the last decade, many steering environments have been developed. They distinguish themselves by their simulation integration process. A first solution is the problem solving environment (PSE) approach, like in SCIRun. This approach allows the scientist to construct a steering application according to a visual programming model. As an opposite, the majority of the steering environments, such as the well-known CUMULVS, are based on the annotation of the application source-code. This latter approach allows fine grain steering functionalities and can achieve better run-time performances.

Even though most of existing computational steering environments, such as CUMULVS, DAQV or gViz support parallel simulations, they are limited to sequential visualization systems. This leads to an important bottleneck and increased rendering time. In the gViz project, the IRIS Explorer visualization system has been extended to run the different modules (simulation, visualization, rendering) in a distributed fashion on the Grid, but the visualization and the rendering modules are still sequential. Recent works in the Uintah PSE (Problem Solving Environment) has addressed the problem of massively parallel computation connected to a remote parallel visualization module, but this latter module is only running on a shared-memory machine. Therefore, it would be particularly *valuable* for the scientist if a steering environment would be able to perform parallel visualization using a PC-based graphics cluster.

In the EPSN project, we intend to explore this latter purpose, called *M×N computational steering*. More precisely, the EPSN environment (Environment for the Steering of Parallel Numerical Simulations, see section 5.2) enables to interconnect parallel simulations with visualization systems, that can be parallel as well. In other words, we want to provide a framework that can benefit from immersive virtual reality technology (e.g. tiled display wall) and that might help scientists to better grasp the complexity of real-life simulations. Such a coupling between parallel numerical simulations and parallel visualization systems raises crucial issues we investigate in the EPSN project:

Parallel coordination of steering operations. The on-line visualization and the computational steering of parallel simulations come up against a serious coherence problem. Indeed, data distributed over parallel processes must be accessed carefully to ensure they are presented to the visualization system in a meaningful way. To overcome this problem, we introduce a Hierarchical Task Model (HTM) that represents the control-flow of the simulation, too often considered in other approaches as a "single-loop" program. Thanks to this representation, we schedule in parallel the user interaction on the simulation processes and satisfy the temporal coherence.

Parallel data redistribution. The efficient redistribution of data is a crucial issue to achieve an efficient coupling between simulation and visualization codes. Most of the previous works have limited their studies to structured data with regular distributions (e.g dense arrays with simple block-cyclic distributions). However, computational steering applications (or multi-physics simulations) frequently handle more irregular data, like particle sets, unstructured meshes or hierarchical grids. In such a context, the data transfer involves to switch from one distribution to another, and thus to "redistribute" the data. For this, we introduce a description model of distributed data, based on the notion of "complex object". Thanks to this model, we propose redistribution algorithms that generate symbolic messages, independently from a particular communication layer. We distinguish two main approaches: the spatial approach and the placement approach.

4. Application Domains

4.1. Introduction

The main objective of the ScAlApplix project is to analyze and solve scientific computing problems coming from complex research and industrial applications and involving scaling. This allows us to validate the numerical schemes, the algorithms and the associated softwares that we develop. We have today three reference application domains which are fluid mechanics, molecular dynamics and host-parasite systems in population dynamics. In these three domains, we study and simulate phenomena that are by nature multiscale and multiphysics, and that require enormous computing power. A major part of these works leads to industrial collaborations in particular with the CNES, ONERA, and with the french CEA/CESTA and CEA/Ile-de-France centers.

4.2. Fluid mechanics

Keywords: *fluid mechanics*, *multiphase flows*, *unsteady flows*, *unstructured meshes*.

The numerical simulation of unsteady flows is still a challenge since efficient schemes and efficient implementations are needed. This challenge is even higher if large size problems are to be tackled, and if the meshes are not regular.

Among the problems to be considered, one may list the computation of mixing layers, shock–vortices interactions, the noise generated by a flow. This last item clearly needs very high order schemes, and the today best schemes use regular structured meshes. Hence, one of our objectives is to construct very high order schemes for unstructured meshes.

Another example where large computer resources are needed is the simulation of multiphase flows. In that case, several difficulties must be faced: unsteady flows, complex geometries and a very complex physical model.

4.3. Molecular chemistry

Keywords: *biological simulation*, *continuum method*, *crack propagation*, *drug design*, *enzyme reaction*, *membrane*, *molecular dynamics*, *protein*, *quantum method*.

Due to the increase of available computer power, new applications such as reaction paths, free energy computations, biomolecular dynamics simulations or failure material simulations are now commonly performed by chemists. These computations simulate systems up to several thousands of atoms, for large time scales up to several nanoseconds. The larger the simulation is, the smaller the computational cost of the potential driving the phenomena is, resulting in low precision results. To achieve realistic results, simulations need to include the environment surrounding the molecules, such as water and membranes, resulting in system sizes up to about several hundred thousands of atoms. Furthermore, simulating the aggregation of proteins, which is critical for biologists studying viruses, requires models of up to one million atoms, with a simulation time up to one millisecond. This implies that atomistic simulations must be speeded up by several orders of magnitude. To obtain this speed, numerical and parallel algorithms must be improved, as well as their implementations on distributed or parallel architectures.

4.4. Population Dynamics

Keywords: *Biomathematics*, *deterministic model*, *health*, *host-macroparasite system*, *host-microparasite system*, *individual based model*, *parasite aggregation*, *population dynamics*, *spatially explicit model*.

In population dynamics, systems can present very complex behaviors and can be difficult to analyse from a purely mathematical point of view. The aim of this interdisciplinary project was to develop numerical tools for population dynamics models arising in modelling complex heterogeneous host-parasite systems. Some typical heterogeneities we consider are spatial locations, age or ability to recruit macroparasites for hosts, age of macroparasites. Our main goals are: understanding the impact of a host population structure on a parasite population dynamics, developing accurate numerical simulations using parallelization, designing prophylactic methods. For many host-parasite systems different time scales between the host population (e.g. a one year period) and the virus (e.g. an infected host dies with a few weeks) require a small time step. Numerical schemes of the resulting nonlinear epidemiological model in spatially heterogeneous environment are complex to perform and reliable numerical results become difficult to get when the size of the spatial domain is increasing. In addition, many input parameters (biological and environmental factors) are taken into account to compare results of simulations and observations from field studies. Therefore, a realistic simulator has a significant computation cost and parallelization is required.

Individual-Based Models (IBM) are becoming more and more useful to describe biological systems. Interactions between individuals are simple and local, yet can lead to complex patterns at a global scale. The principle is to replicate several times the simulation program to obtain statistically meaningful results. The Individual-Based Model approach contrasts with a more aggregate population modeling approach in providing low level mechanisms to manage the population interactions. Stochastic simulations reproduce elementary processes and often lead to prohibitive computations; thus we need parallel algorithmic.

In our developments of both stochastic and deterministic models, biological processes are combined to reach a good level of realism. For host-parasite systems, it make a big difference with purely mathematical models, for which numerical results could hardly be compared to observations. Parallel numerical simulations mimic some of the dynamics observed in the fields, and supply a usable tool to validate the models. This work is a collaborative effort in an interdisciplinary approach between population dynamics, mathematics and computer science.

5. Software

5.1. Introduction

We develop two kinds of software. The first one consists in generic libraries that will be used in the applications. We work on a (parallel) partitioner for large irregular graphs or meshes (Scotch), on high performance direct or hybrid solvers for very large sparse systems of equations (MUMPS, PaStiX, HIPS). The second kind of software corresponds to dedicated softwares for molecular chemistry (QC++), fluid mechanics (FluidBox), and to a platform for computational steering (EPSN). For these parallel software developments, we use the message passing (MPI) paradigm, the OpenMP programming language, threads, and the Java and/or CORBA technologies.

5.2. EPSN

Keywords: *code coupling*, *computational steering*, *data redistribution*, *numerical simulation*, *scientific visualization*.

Participants: Olivier Coulaud, Aurélien Esnard, Nicolas Richart, Mathieu Souchaud.

The EPSN project has been partially supported by both the ACI-GRID program from the french Ministry of Research (grant number PPL02-03), the ARC RedGRID, and is now supported by the ANR program (grant number ANR-05-MMSA-0008-03).

EPSN is a distributed computational steering environment which allows the steering of remote parallel simulations with sequential or parallel visualization tools (user interface). It is a distributed environment based on a simple client/server relationship between user interfaces (clients) and simulations (servers). The user interfaces can dynamically be connected to or disconnected from the simulation during its execution. Once a client is connected, it interacts with the simulation component through an asynchronous and concurrent request system. We distinguish three kinds of steering request. Firstly, the "control" requests (play, step, stop) allow to steer the execution flow of the simulation. Secondly, the "data access" requests (get, put) allow to read/write parameters and data from the memory of the remote simulation. Finally, the "action" requests enable to invoke user-defined routines in the simulation. In order to make a legacy simulation steerable, the end-user annotates its simulation source-code with the EPSN API. These annotations provide the EPSN environment with two kinds of information: the description of the program structure according to a Hierarchical Task Model (HTM) and the description of the distributed data that will be accessible by the remote clients.

Concerning the development of client applications, we also provide a front-end API that enables to integrate EPSN in a high-level visualization system such as *AVS/Express*, *VTK* or *Paraview*. We also provide a lightweight user interface, called *Simone*, that enables to easily connect any simulations and interact with them, by controlling the computational flow, viewing the current parameters or data on a simple data-sheet and modifying them optionally. *Simone* also includes simple visualization plug-ins to online display the intermediate results. Moreover, the EPSN framework offers the ability to exploit parallel visualization and rendering techniques thanks to the Visualization ToolKit (VTK). This approach allows to reduce the steering overhead of the EPSN platform and allow to process efficiently large dataset. To visualize with high resolution image and to improve the rendering time, EPSN can also exploit tiled-display wall based on ICE-T library developed at Sandia Laboratory.

As both the simulation and the visualization can be parallel applications, EPSN is based on the M×N redistribution library called RedGRID. This library is in charge of computing all the messages that will be exchanged between the two parallel components, and is also in charge of performing the data transfer in parallel. Thus, RedGRID is able to aggregate the bandwidth and to achieve high performance. Moreover, it is designed to consider a wide variety of distributed data structures usually found in the numerical simulations, such as structured grids, particle sets or unstructured meshes.

Both EPSN and RedGRID use a communication infrastructure based on CORBA which provides our platform with portability, interoperability and network transparency. The current version of EPSN and RedGRID libraries are now available at INRIA Gforge :

- EPSN : http://gforge.inria.fr/projects/epsn,
- RedGRID :http://gforge.inria.fr/projects/redgrid.

5.3. FluidBox

Keywords: *fluid mechanics*, *inert and reactive flows*, *multimaterial and multiphase flows*, *upwind and residual distribution scheme*.

Participants: Rémi Abgrall, Christophe Berthon, Cécile Dobrzynski-Proust, Boniface Nkonga [corresponding member], Mikaël Papin, Mario Ricchiuto.

FluidBox is a software dedicated to the simulation of inert or reactive flows. It is also able to simulate multiphase and multimaterial flows. There exist 2D and 3D dimensional versions. The 2D version is used to test new ideas that are later implemented in the 3D one. Two classes of schemes have been implemented: classical finite volume schemes and the more recent residual distribution schemes. Several low Mach preconditioning techniques are also implemented. The code has been parallelized with and without overlap of the domains. Recently, the PaStiX solver has been integrated in FluidBox. FluidBox has also been coupled with the EPSN plateform. In order to facilitate the project development, FluidBox has been downloaded on the INRIA/Gforge page. Up to now, it is only a private project, but we expect to open some part of the code to public.

5.4. MUMPS

Keywords: *parallel asynchronous sparse multifrontal solver*.

Participants: Patrick Amestoy, Abdou Guermouche [corresponding member].

In the context of PARASOL (Esprit IV Long Term Project, 1996-1999), CERFACS and ENSEEIHT-IRIT teams have initiated a parallel sparse solver MUMPS ("MUltifrontal Massively Parallel Solver"). Since the first public release of MUMPS (March 2000), this research and (also software) project is the context of a tight and fruitful collaboration with J. Y. L'Excellent (INRIA-LIP-ENS Lyon) and the INRIA project GRAAL. Recent work related to performance scalability, preprocessing of both symmetric and unsymmetric matrices, two by two pivots for symmetric indefinite matrices, and dynamic scheduling has been incorporated in the new improved version of the package (release 4.5.5 available since october 2005 at http://www.enseeiht.fr/apo/MUMPS or http://graal.ens-lyon.fr/MUMPS).

MUMPS is a package for solving linear systems of equations $Ax = b$, where the matrix A is sparse and can be either unsymmetric, symmetric positive definite, or general symmetric. It uses a multifrontal technique which is a direct method based on either the LU or the LDL^T factorization of the matrix. The main features of the MUMPS package include numerical pivoting during factorization, solution of the transposed system, input of the matrix in assembled format (distributed or centralized) or elemental format, error analysis, iterative refinement, scaling of the original matrix, and return of a Schur complement matrix. It also offers several built-in ordering algorithms, a tight interface to some external ordering packages such as Scotch and is available in various arithmetics (real or complex, single or double).

This year, we particularly focussed in a preliminary out-of-core functionality, where computed factors are written to disk. It has been made available to some of our users, in order to get their feedback before making this functionality more widely available. We have also been working on reducing the memory requirements for symmetric matrices (by using a packed format for temporary Schur complements) as well as for unsymmetric matrices and have modified the general memory management algorithms to allow for more flexibility in an out-of-core context.

5.5. PaStiX

Keywords: *complete and incomplete supernodal sparse parallel factorizations*.

Participants: Pascal Hénon, François Pellegrini, Pierre Ramet [corresponding member], Jean Roman.

This work is supported by the French "Commissariat à l'Energie Atomique CEA/CESTA" in the context of structural mechanics and electromagnetism applications.

PaStiX (Parallel Sparse matriX package) (http://pastix.gforge.inria.fr) is a scientific library that provides a high performance parallel solver for very large sparse linear systems based on block direct and block ILU(k) iterative methods. Numerical algorithms are implemented in simple or double precision (real or complex): LLt (Cholesky), LDLt (Crout) and LU with static pivoting (for non symmetric matrices having a symmetric pattern). This latter version is now used in FluidBox (see section 5.3). The PaStiX library is planed to be released this year under INRIA CeCILL licence.

The PaStiX library uses the graph partitioning and sparse matrix block ordering package Scotch (see section 5.8). PaStiX is based on an efficient static scheduling and memory manager, in order to solve 3D problems with more than 10 millions of unknowns. The mapping and scheduling algorithm handles a combination of 1D and 2D block distributions. This algorithm computes an efficient static scheduling of the block computations for our supernodal parallel solver which uses a local aggregation of contribution blocks. This can be done by taking into account very precisely the computational costs of the BLAS 3 primitives, the communication costs and the cost of local aggregations. We also improved this static computation and communication scheduling algorithm to anticipate the sending of partially aggregated blocks, in order to free memory dynamically. By doing this, we are able to reduce dramatically the aggregated memory overhead, while keeping good performances.

Another important point is that our study is suitable for any heterogeneous parallel/distributed architecture when its performances are predictable, such as clusters of SMP nodes. In particular, we propose now a high performance version with a low memory overhead for SMP node architectures, which fully exploits shared memory advantages by using an hybrid MPI-thread implementation.
Direct methods are numerically robust methods, but the very large three dimensional problems may lead to systems that would require a huge amount of memory despite any memory optimization. A studied approach consists to define an adaptive blockwise incomplete factorization that is much more accurate (and numerically more robust) than the scalar incomplete factorizations commonly used to precondition iterative solvers. Such incomplete factorization can take advantage of the latest breakthroughts in sparse direct methods and particularly should be very competitive in CPU time (effective power used from processors and good scalability) while avoiding the memory limitation encountered by direct methods.

5.6. HIPS

Keywords: *domain decomposition*, *multilevel method*, *parallel iterative solver*, *schur complement*.

Participants: Jérémie Gaidamour, Pascal Hénon [corresponding member], Jean Roman, Yousef Saad.

HIPS (Hierarchical Iterative Parallel Solver) is a scientific library that provides an efficient parallel iterative solver for very large sparse linear systems.

HIPS has been built on top of the PHIDAL library; it is based on the ordering and partitioning that were developed in PHIDALand it proposes some new hybrid direct iterative algorithms.

The keypoint of the methods implemented in HIPS is to defined an ordering and a partition of the unknowns that relies on a form of nested dissection ordering in which cross points in the separators play a special role (Hierarchical Interface Decomposition ordering). The subgraphs obtained by the nested dissection correspond to the unknowns that are eliminated using a direct method and the Schur complement system on the remaining of the unknowns (that correspond to the interface between the subgraphs viewed as subdomains) is solved using an iterative method (GMRES or Conjugate Gradient at the time being).

This special ordering and partitioning allow the use of dense block algorithms both in the direct and iterative part of the solver and provides a high degree of parallelism to these algorithms.

We propose several algorithmic variants to solve the Schur complement system that can be adapted to the geometry of the problem: typically some strategies are more suitable for systems coming from a 2D problem discretization and others for a 3D problem; the choice of the method also depends on the numerical difficulty of the problem. Thus HIPS is a generic library that provides several methods to build an efficient preconditioner in many of these situations. It handles symmetric, unsymmetric, real or complex matrices. It also provides the scalar preconditioner based on the multistage ILUT factorization that were developed in PHIDAL. HIPS has been downloaded as a private project on InriaGForge and will be released in the next year.

5.7. QC++

Keywords: *parallelism*, *quantum chemistry*, *semi-empirical*, *simulation*.

Participants: Olivier Coulaud [corresponding member], François Pellegrini, Gérald Monard.

QC++ is a Quantum Chemistry software written in C++. The current version of QC++ supports the semi-empirical quantum models MNDO, AM1 and PM3. It allows to calculate the energy of a molecular configuration by some Self Consistent Field (SCF) algorithms : fixed point, optimal damping and level shifting. It is also possible to optimize its geometry by using the minimization algorithms L-BFGS and BFGS.

The major new feature in the version 2.0 is the implementation of a linear scaling "*divide and conquer*" method both in sequential and parallel to calculate the electronic energy. Several type of subdomains and strategies to partition a molecule are available, one of them being based on Scotch.

5.8. Scotch

Keywords: *graph partitioning*, *mesh partitioning*, *parallel sparse matrix block ordering*, *sparse matrix block ordering*, *static mapping*.

Participants: Cédric Chevalier, François Pellegrini [corresponding member].

The initial purpose of Scotch was to provide an efficient software environment for partitioning and mapping statically applications modeled as valuated process graphs of arbitrary topologies. The original contribution consisted in developing a "*divide and conquer*" algorithm in which processes are recursively mapped onto processors by using graph bisection algorithms that are applied both to the process graph and to the architecture graph. This allows the mapper to take into account the topology and hetorogeneity of the valuated graph which models the interconnection network and its resources (processor speed, link bandwidth). This technique allowed to compute high quality mappings with low complexity.

The software has then be extended in order to produce vertex separators instead of edge separators, using a multi-level framework. Recursive vertex separation is used to compute reorderings of the unknowns of large sparse linear systems, which both preserve sparsity when factoring the matrix and preserve concurrency for computing and solving the factored matrix in parallel. The original contribution has been to study and implement a tight coupling between the nested dissection and the approximate minimum degree methods; this work was carried out in collaboration with Patrick Amestoy, of ENSEEIHT-IRIT.

Two years ago, new classes of methods have been added to the Scotch library, which allow it to compute efficient orderings of native meshes, resulting in the handling of larger problems than with standard graph partitioners. Meshes are represented as bipartite graphs, in which node vertices are connected to element vertices only, and vice versa. Since this structure is equivalent to an hypergraph, where nodes are connected to hyperedges only, and vice versa, the mesh partitioning routines of Scotch turn it into a hypergraph partitioner.

Version 4.0 of the Scotch software package (http://www.labri.fr/~pelegrin/scotch/) has been formally released in February 2006 on InriaGForge, as a LGPLed libre software, in order to encourage members of the community to use it as a testbed for the quick and easy development of new partitioning and ordering methods. As of December 1st, 2006, it has already been downloaded more than 1200 times. Scotch can be called from MUMPS, PaStiX and HIPS as an external ordering library. It is also part of the latest release of CODE_ ASTER, a GPLed thermal and mechanical analysis software developed by French state-owned electricity producer EDF.

Release 5.0 of Scotch, which will include the parallel ordering routines of the PT-SCOTCH library, is planned for the early days of 2007. It will extend the capabilities of Scotch in parallel, allowing users to compute in parallel orderings of distributed graphs. PT-SCOTCH is based on the MPI and Posix Pthread libraries.

6. New Results

6.1. Numerical schemes and algorithms for fluid mechanics.

Keywords: *Compact distribution schemes*, *mesh adaptation*, *multimaterial and multiphase flows*.

Participants: Rémi Abgrall, Christophe Berthon, Andreas Bollerman, Benjamin Braconnier, Cecile Dobrzynski, Adam Larat, Mikaël Lutz, Boniface Nkonga, Mikaël Papin, François Pellegrini, Vincent Perrier, Pierre Ramet, Mario Ricchiuto, Jean Roman, Cédric Tavé.

6.1.1. Residual distribution schemes

This year many developments have been conducted and implemented in the FluidBox software after [1] which has open up many doors.

One may list the developments of stabilised and quasi monotone centered RD schemes [57], [56], the approximation of viscous terms that is consistent with what is done on the convective/hyperbolic part. This item has been worked out in collaboration with N. Villedieu and H. Deconinck from VKI and Jiri Dobes from CTU in Prag[71], [59]. These schemes have all been extended to unsteady problems and quadrilateral meshes. Thanks to a careful analysis of the implicit phase of the scheme, we have been able to reduce the CPU cost by a half.

We have also started to work on very high order residual distribution schemes. Lagrange interpolant of degree 3 an 4 are considered with either an upwind or a centered formulation [37], [38].

The work on shallow water have been prolongated. The scheme is now able to handle dry beds. We have shown that genuinely steady 2D are numericaly preserved by the scheme [9].

With C.W. Shu (Brown University), we have also developped a new method which is somewhat between the Residual distribution schemes that need continuous interpolant and the Discontinuous Galerkin ones where the interpolant is discontinuous (article in preparation).This method is currently second order in space.

Mario Ricchiuto has written in collaboration with H. Deconinck a chapter for the Encyclopedia of Computational Mechanics, [68].

6.1.2. Relaxation schemes

The stability properties of this schemes, as well as the easyness with which they can be implemented makes them mandatory for many complex applications. This year two have been considered : multifluid flow problems [28], [24], and radiative transfert with CELIA [21], [25], [45].

6.1.3. MUSCL type scheme

We have developped a numerical procedure for which we can prove that mass and pressure remain positive under a standard CFL type property. The method has been validated against many standard test cases [4], [23], [46].

6.1.4. Interface and multifluid problems

In collaboration with CEA–CESTA, we are developping an interface tracking method using the level set method with the Ghost fluid technique. The method has been implemented in 2D and validated against several benchmark tests, in particular for interface between compressible and incompressible fluids[47], [72].

B. Braconnier is finishing his thesis on the development of numerical procedures for the simulation of low mach number flows[55]. The method use relaxation solvers. The scheme, which is implicit, uses the `PaStiX` library, and this strategy seems to be the most efficient. In the near future, we expect to be able to consider larger problems by using the iterative method library that is developped in the team.

6.1.5. Phase transition

V. Perrier has been studying the structure of discontinuities in the seven equation model and the five equation model using traveling wave techniques. This topic is done in collaboration with H. Guillard from the Smash project in Sophia Antipolis.

6.1.6. PaStiX and FluidBox

M. Papin has integrating the `PaStiX` high performance solver in the `FluidBox` code. The results have been presented in [48]. This technique will be used within the ADIGMA project, as well as more recent work on iterative solvers by J. Gaidamour, P. Hénon, P. Ramet and J. Roman.

6.1.7. Parallelization and fluid mechanics

We have developped a parallel strategy by domain decomposition which is tuned for aerodynamics problems (THOT code of CEA/CESTA). The problem was to construct domain decomposition criteria which preserve the multi-block structure of the mesh. The code is fully operational and will be delivered to CEA/CESTA in the near future.

6.1.8. Mesh adaptation

C. Dobrzynski has worked on fully parallel mesh adaptation procedure that uses standard sequential mesh adaptation codes. The idea is to adapt the mesh on each processor without changing the interfaces. Then the interfaces are modified. The main advantage is the simplicity because there is no need to parallelize mesh generation tools (insert/delete, swap, etc). The main techniques are described in [31], [30].

We also have started to work on the definition of an anisotropic metric which is computed from the he output of a Residual distribution code. Once this will be done, standard mesh adaptation method will be used so that the numerical error of the solution is controled.

6.2. Schemes and algorithms for computational chemistry

Keywords: *coupling model*, *hierarchical methods*, *molecular dynamics*, *quantum chemistry*.

Participants: Guillaume Anciaux, Olivier Coulaud, Pierre Fortin, Jean Roman.

6.2.1. Crack simulation

A study of crack propagation in silica glasses with a coupling method between molecular dynamics and elasticity begun in collaboration with the CEA Ile-de-France in December 2003. Simulations which follow crack propagation at atomistic level lead to huge number of atoms on a small domain. The coupling between two length scales allows us to treat larger domains with smaller number of atoms. Nevertheless 3D atomistic simulations involve several million atoms; they must be parallel and use a coupling with elasticity codes based on finite element approximation.

Our algorithm to couple such models is based on the Bridging method introduced by T. Belytschko. We have extended our previous work on 1D analysis of the model to higher dimension and we have developed a parallel framework to compute and visualize the coupling algorithms. This framework allows us to couple finite element technique with molecular dynamics. We validated the approach based on the Bridging Method on several multi-dimensional cases like wave propagation and crack propagation. The coupling algorithm solves a coupling linear equation and redistributes the corrections among degrees of freedom (atoms and finite elements nodes). Optimized data structures have been used in several parts of the coupling process. For example we build an efficient algorithm based on an initial computing of the finite element shape functions in order to accelerate the field's interpolation at atom positions. One other crucial service of the framework is the ability to control and forward the information on dynamic load balance strategies. Those strategies migrate atoms between processors; thus the communication scheme to update the variables attached to the coupling system (like dofs) needs to be updated [2]. Moreover, this framework integrates EPSN that allows a powerful monitoring. An article on the description of that framework and justifying all the choices that have been made is under writing.

6.3. Numerical schemes and algorithmic for population dynamics

Keywords: *coupling model*, *deterministic simulator*, *stochastic simulator*.

Participants: Guillaume Latu, Jean Roman, Gaël Tessier.

In population dynamics, systems can present very complex behaviors and can be difficult to analyse from a purely mathematical point of view. The aim of this interdisciplinary project was to develop numerical tools for population dynamics models arising in modeling complex heterogeneous host-parasite systems. Some typical heterogeneities we consider are spatial locations, age or ability to recruit macroparasites for hosts. Our main goals are: understanding the impact of a host population structure on a parasite population dynamics, developing accurate numerical simulations using parallelization, designing prophylactic methods. For many host-parasite systems different time scales between the host population (e.g. a one year period) and the virus (e.g. an infected host dies with a few weeks) require a small time step. Numerical schemes of the resulting nonlinear epidemiological model in spatially heterogeneous environment are complex to perform and reliable numerical results become difficult to get when the size of the spatial domain is increasing. In addition, many input parameters (biological and environnmental factors) are taken into account to compare results of simulations and observations from field studies. Therefore, a realistic simulator has a significant computation cost and parallelization is required.

Individual-Based Models (IBM) are becoming more and more useful to describe biological systems. Interactions between individuals are simple and local, yet can lead to complex patterns at a global scale. The principle is to replicate several times the simulation program to obtain statistically meaningful results. The Individual-Based Model approach contrasts with a more aggregate population modeling approach in providing low level mechanisms to manage the population interactions. Stochastic simulations reproduce elementary processes and often lead to prohibitive computations; thus we need parallel algorithmic.

In our developments of both stochastic and deterministic models, biological processes are combined to reach a good level of realism. For host-parasite systems, it make a big difference with purely mathematical models, for which numerical results could hardly be compared to observations. Parallel numerical simulations mimic some of the dynamics observed in the fields, and supply a usable tool to validate the models. This work is a collaborative effort in an interdisciplinary approach between population dynamics, mathematics and computer science.

A cooperation involving a biologist (Agnès Calonnec - INRA UMR Santé végétale 1065 - Villenave d'Ornon) and a thesis student in computer science (Gaël Tessier) began since october 2003. Using numerical methods and parallel technics, we are interested in modeling the spread of *powdery mildew*, a disease of vineyard. Correct prediction of this type of parasite epidemics needs an realistic simulator, and could have an industrial impact.

An architectural model of vine stocks is used for two purposes: the study of the growth of stocks and the influence of its structure on the dispersal of powdery mildew. In this model, we consider a large number of infectious elements and several spatially hetereogeneous environmental parameters. Indeed, the dispersal of powdery is a multiscale mechanism that takes place within vine stocks, and along and across the rows of the vineyard. An initial version of a parallel simulator using MPI communications has been developed. A characterization of the implemented algorithms is presented in [82]; we evaluate particularly the communication costs and the load imbalance. First results indicated a good scalability up to 24 processors. Further experiments were carried out on clusters of SMP nodes, up to 128 processors [73]. This revealed that the part of time spent for communications and synchronizations highly increases for simulations that uses 64 processors and more. Relative efficiency drops to 63 % with 128 processors.

An hybrid approach mixing processes and threads has been considered: the idea is to benefit from the high speed of shared memory accesses by replacing n monothreaded processes in the previous parallel simulator by n/p processes, each one containing one master thread responsible for inter-process MPI communications and p simulation threads running inside the same SMP node. Simulation threads compute the growth of vinestocks and colonies of powdery mildew, and the dispersal of aerial spores. Threads in a same process can exchange data via the shared memory, avoiding MPI communications. Furthermore, communications between nodes can be aggregated for all the threads of the nodes, and load-balancing can be improved by exchanging vinestocks between threads of a process. The implementation and the performances of this hybrid simulator were presented in [10]. A partial dynamic load-balancing turned out to be necessary to reduce the cost of synchronizations between threads, and permit to improve the scalability of the simulator.

6.4. Algorithms and high-performance solvers

Keywords: *direct and hybrid direct-iterative solvers*, *graphs and irregular meshes*, *high performance computing*, *parallel fast multipole methods*, *parallel sparse matrix ordering*, *scalable parallel graph partitioning*, *sparse matrix ordering*.

Participants: Patrick Amestoy, Olivier Beaumont, Cédric Chevalier, Olivier Coulaud, Mathieu Faverge, Pierre Fortin, Jérémie Gaidamour, Abdou Guermouche, Pascal Hénon, François Pellegrini, Pierre Ramet, Jean Roman.

6.4.1. Parallel domain decomposition and sparse matrix reordering

The work carried out within the Scotch project (see section 5.8) was focused on the design and coding of the parallel graph ordering routines which are the first building blocks of the PT-SCOTCH (for "Parallel Threaded Scotch") parallel graph partitioning and ordering library. PT-SCOTCH is being designed so as to be able to handle graphs up to a billion vertices on architectures of a thousand processors. This work is carried out in the context of the PhD thesis of Cédric Chevalier, who started on October 2004 and will graduate by September 2007. In order to achieve efficient and scalable parallel graph partitioning, it is necessary to implement a parallel multi-level framework in which distributed graphs are collapsed down to a size which can be handled by a single processor, on which a sequential partition is computed by means of the existing Scotch tool, after which this coarse solution is expanded and refined, level by level, up to obtain a partition of the original distributed graph.

The multi-level framework that we have developed is as follows. Distributed graphs are coarsened using an asynchronous multi-threaded algorithm, and coarsened graphs are folded on one half of the processors, while the other half receives a similar copy. The coarsening and folding process then goes on independently on each of the halves, then quarters, eigths, and so on, of the processors, resulting in the obtainment of different coarsened graphs on every processor, which provides for a better exploration of the problem space when initial partitions are sequentially computed on each of them. Then, best solutions are unfolded and propagated back to the finer graphs, selecting the best partitions of the two at each unfolding stage. In order to keep partition quality at the finest levels, best unfolded partitions have to be refined at every step by means of local optimization algorithms. The problem is that the best sequential local optimization algorithms do not parallelize well, as they are intrinsically sequential, and attempts to relax this strong sequential constraint can lead to severe loss of partition quality when the number of processors increase. However, as the refinement is local, we have overcome this difficulty by proposing in [5] an algorithm where the optimization is applied only to band graphs that contains vertices that are at a small fixed distance (typically 3) from the projected separators. What we have implemented is a multi-sequential approach: at every uncoarsening step, a distributed band graph is created. Centralized copies of this band graph are then created on every participating processor. These copies can be used collectively to run a scalable parallel multi-deme genetic optimization algorithm, or fully independent runs of a full-featured sequential optimization algorithm. The best refined band separator is projected back to the distributed graph, and the uncoarsening process goes on. All of these algorithms have been successfully implemented in PT-SCOTCH, which can compute in parallel orderings that are of the same, and even better, quality than the ones computed by the sequential Scotch [74]. However, due to the amount of communication to be exchanged at the coarsening phase, scalability is still an issue to be addressed.

6.4.2. High-performance direct solvers on multi-plateforms

In order to solve linear systems of equations coming from 3D problems and with more than 10 millions of unkowns, which is now a reachable challenge for new SMP supercomputers, the parallel solvers must keep good time scalability and must control memory overhead caused by the extra structures required to handle communications.

Static parallel supernodal approach. Some experiments were run on the TERA parallel computer at CEA, and the factorization times are close to the ones obtained on the IBM SP3 of CINES (Montpellier, France). For example, on our largest problem (26 millions of unknowns for a 2D 1/2 problem), we reach 500Gflops on 768 processors, that is, about 50% of the peak performance of the TERA computer. In the context of new SMP node architectures, we proposed to fully exploit shared memory advantages. A relevant approach is then to use an hybrid MPI-thread implementation. This not yet explored approach in the framework of direct solver aims at solve efficiently 3D problems with much more than 10 millions of unkowns. The rationale that motived this hybrid implementation was that the communications within a SMP node can be advantageously substituted by direct accesses to shared memory between the processors in the SMP nodes using threads. In addition, the MPI communications between processes are grouped by SMP node. We have shown that this approach allows a great reduction of the memory required for communications [79], [81],. Many factorization algorithms are now implemented in real or complex variables, for single or double precision: LLt (Cholesky), LDLt (Crout) and LU with static pivoting (for non symmetric matrices having a symmetric pattern). This latter version is now integrated in the `FluidBox` software [48]. A survey article on thoses techniques is under preparation and will be submitted to the SIAM journal on Matrix Analysis and Applications. It will present the detailed algorithms and the most recent results. We have to add numerical pivoting technique in our processing to improve the robustness of our solver. Moreover, in collaboration with the MUMPS developers (see section 5.4), we want to adapt Out-of-Core techniques to overcome the physical memory constraints.

Dynamic parallel multifrontal approach. The memory usage of sparse direct solvers can be the bottleneck to solve large-scale problems involving sparse systems of linear equations of the form Ax=b. If memory is not large enough to treat a given problem, disks must be used to store data that cannot fit in memory (*out-of-core* storage). In a previous work, we proposed a first out-of-core extension of a parallel multifrontal approach based on the solver MUMPS, where only the computed factors were written to disk during the factorization. We have studied this year in detail the minimum memory requirements of this parallel multifrontal approach and proposed several mechanisms to decrease further those memory requirements. For a given amount of memory, we have also studied the volume of disk accesses involved during the factorization of matrix A, providing insight on the extra cost that we can expect due to I/O. Furthermore, we have studied the impact of low-level I/O mechanisms, and have in particular shown the interest (and difficulty) of relying on direct I/O. Large-scale problems from applicative fields have been used to illustrate our discussions. This work is performed in the context of the PhD of Emmanuel Agullo, in collaboration with Jean-Yves L'Excellent (INRIA project GRAAL) and Patrick Amestoy (ENSEEIHT-IRIT). Once the factors are on disk, they have to be read back for the solution step. In order to improve that step, we collaborate with Tzvetomila Slavova (Ph.D. CERFACS) who focusses on it. For instance we are currently designing an algorithm which allows to schedule the I/O in a way that separates the L and U factors on disk during the factorization step in the unsymmetric case: this will allow to perform twice less reads at the solution step for unsymmetric matrices. A collaboration with Xiaoye S. Li and Esmond G. Ng (Lawrence Berkeley National Laboratory, Berkeley, California, USA) was started to compare the multifrontal factorization to the left-looking.

We work with Tzetomila Slavova (Ph.D. at CERFACS) on the study of the performance of the out-of-core solution phases (forward and backward substitutions). In many applications, the solution phase can be invoked many times for a unique factorization phase. In an out-of-core context, the solution phase can thus become even more costly than the factorization. In a first approach, we can rely on system buffers (or page cache) using a demand-driven scheme to access the disks. We have shown that this approach was not adapted because it cannot "choose" correctly data that must be kept in memory. Furthermore, it is important to really control the amount of memory effectively used (system buffers included). Therefore, a version with direct I/O mechanisms (which does not use system buffers) has been introduced, and we have shown that the performance was comparable with the system approach, with the advantage of effectively controlling the memory usage (there is no use of systeme caches hidden to the application). In a parallel environment we have also shown that the order in which the dependency tree is processed could have a very strong impact on performance, because of the (ir)regularity of the disk accesses involved. Finally, we proposed some heuristics that aim at influencing scheduling decisions taken during the solution step to ensure a high locality for disk accesses.

Finally, in the context of distributed NUMA architectures, a work with the INRIA RUNTIME team to study optimization strategies, to describe and implement communications, threads and I/O scheduling has recently begun. Our solvers will use *Madeleine* and *Marcel* libraries in order to provide an experimental application to validate those strategies. M. Faverge has started a Ph.D. (since october 2006) to study these aspects in the context of the NUMASIS ANR CIGC project. Note that in the Out-of-Core context, new problems linked to the scheduling and the management of the computational tasks may arise (processors may be slowed down by I/O operations). Thus, we have to design and study specific algorithms for this particular context (by extending our work on scheduling for heterogeneous platforms).

6.4.3. High-performance hybrid direct-iterative solvers for large sparse systems

The resolution of large sparse linear systems is often the most consuming step in scientific applications. Parallel sparse direct solver are now able to solve efficiently real-life three-dimensional problems having in the order of several millions of equations, but anyway they are limited by the memory requirement. On the other hand, the iterative methods require less memory, but they often fail to solve ill-conditioned systems.

We have developped two approaches in order to find some trade-off between these two classes of methods.

In these work, we consider an approach which, we hope, will bridge the gap between direct and iterative methods. The goal is to provide a method which exploits the parallel blockwise algorithmic used in the framework of the high performance sparse direct solvers. We want to extend these high-performance algorithms to develop robust parallel incomplete factorization based preconditioners for iterative methods such as GMRES or Conjugate Gradient.

Block ILUK preconditioner. The first idea is to define an adaptive blockwise incomplete factorization that is much more accurate (and numerically more robust) than the scalar incomplete factorizations commonly used to precondition iterative solvers. Such incomplete factorization can take advantage of the latest breakthroughts in sparse direct methods and particularly should be very competitive in CPU time (effective power used from processors and good scalability) while avoiding the memory limitation encountered by direct methods. By this way, we expect to be able to solve systems in the order of hundred millions of unknowns and even one billion of unknowns. Another goal is to analyse and justify the chosen parameters that can be used to define the block sparse pattern in our incomplete factorization.

The driving rationale for this study is that it is easier to incorporate incomplete factorization methods into direct solution software than it is to develop new incomplete factorizations. Our main goal at this point is to achieve a significant diminution of the memory needed to store the incomplete factors (with respect to the complete factors) while keeping enough fill-in to make the use of BLAS3 (in the factorization) and BLAS2 (in the triangular solves) primitives profitable.

In [78] and [80], we have shown the benefit of this approach over classic scalar implementation and also over direct factorisations. Indeed, on the AUDI problem (that is a reference 3D test case for direct solver with about one million of unknowns), we are able to solve the system in half the time required by the direct solver while using only one tenth of the memory needed (for a relative residual precision of 10^{-7}). We now expect to improve the convergence of our solver that fails on more difficult problems.

This research was included in a NSF/INRIA project and is carried out in collaboration with Yousef Saad (University of Minneapolis, USA).

Recently, we have focused on the critical problem to find approximate supernodes of ILU(k) factorizations. The problem is to find a coarser block structure of the incomplete factors. The "exact" supernodes that are exhibited from the incomplete factor non zero pattern are usually very small and thus the resulting dense blocks are not large enough for an efficient use of the BLAS3 routines. A remedy to this problem is to merge supernodes that have nearly the same structure. The preliminary results [53], [54], [77] have shown the benefits of the new approach.

Hybrid direct-iterative solver based on a Schur complement approach. In recent years, a few Incomplete LU factorization techniques were developed with the goal of combining some of the features of standard ILU preconditioners with the good scalability features of multi-level methods. The key feature of these techniques is to reorder the system in order to extract parallelism in a natural way. Often a number of ideas from domain decomposition are utilized and mixed to derive parallel factorizations.

Under this framework, we developped in collaboration with Yousef Saad (University of Minnesota) algorithms that generalize the notion of "faces" and "edge" of the "wire-basket" decomposition. The interface decomposition algorithm is based on defining a "hierarchical interface structure" (HID). This decomposition consists in partitioning the set of unknowns of the interface into components called connectors that are grouped in "classes" of independent connectors [8].

This year, in the context of robust preconditioner technique, we have developed an approach that uses the HID ordering to define a new hybrid direct-iterative solver. The principle is to build a decomposition of the adjacency matrix of the system into a set of small subdomains (the typical size of a subdomain is around a few hundreds or thousand nodes) with overlap. We build this decomposition from the nested dissection separator tree obtained using a sparse matrix reordering software as `Scotch` or `METIS`. Thus, at a certain level of the separator tree, the subtrees are considered as the interior of the subdomains and the union of the separators in the upper part of the elimination tree constitutes the interface between the subdomains.

The interior of these subdomains are treated by a direct method. Solving the whole system is then equivalent to solve the Schur complement system on the interface between the subdomains which has a much smaller dimension. We use the hierarchical interface decomposition (HID) to reorder and partition this system. Indeed, the HID gives a natural dense block structure of the Schur complement. Based on this partition, we define some efficient block preconditioners that allow the use of BLAS routines and a high degree of parallelism thanks to the HID properties.

We propose several algorithmic variants to solve the Schur complement system that can be adapted to the geometry of the problem: typically some strategies are more suitable for systems coming from a 2D problem discretization and others for a 3D problem; the choice of the method also depends on the numerical difficulty of the problem [52]. For the moment, only a sequential version of these techniques have been implemented in a library (`HIPS`). It provides several methods to build an efficient preconditioner in many of these situations. It handles symmetric, unsymmetric, real or complex matrices. `HIPS` has been built on top of the `PHIDAL` library and thus also provides some scalar preconditioner based on the multistage ILUT factorization (defined in [8]). Those works are the thesis subject of Jérémie Gaidamour that has started since october 2006.

6.4.4. Parallel fast multipole method

The Fast Multipole Method (FMM) is a hierarchical method which computes interactions for the N-body problem in O(N) time for any given precision. In order to compute energy and forces on large systems, we need to improve the computation speed of the method.

This has been realized thanks to a matrix formulation of the main operator in the far field computation : this matrix formulation is indeed implemented with BLAS routines (Basic Linear Algebra Subprograms). Even if it is straightforward to use level 2 BLAS (corresponding to matrix-vector operations), the use of level 3 BLAS (that corresponds to matrix-matrix operations) is interesting because much more efficient. So, thanks to a careful data memory storage, we have rewritten the algorithm in order to use level 3 BLAS, thus greatly improving the overall runtime. Other enhancements of the Fast Multipole Method, such as the use of Fast Fourier Transform, the use of « rotations » or the use of plane wave expansions, allow the reduction of the theoretical operation count. Comparison tests have shown that, for the required precisions in astrophysics or in molecular dynamics, our approach is either faster (compared to rotations and plane waves) or as fast and without any numerical instabilities (compared to the FFT based method), hence justifying our BLAS approach. These results are detailed in [76] and have been submitted for publication [67]. Our BLAS version has then been extended to non uniform distributions, requiring therefore a new octree data structure named octree with indirection, that is efficient for both uniform and non uniform distributions. We have also designed an efficient algorithm that detects uniform areas in structured non uniform distributions, since these areas are more suitable for BLAS computations. These results have been presented in [29] and in [75]. An efficient parallel code of our BLAS version has finally been developed and validated on shared and distributed memory architectures in [14].

We now plan to improve our parallelization thanks to a hybrid MPI-thread programming and to integrate our FMM in complete codes for real simulations in astrophysics and in molecular dynamics.

6.5. Parallel algorithms for heterogeneous platforms

Keywords: *collective communications*, *communication/computation overlapping*, *divisible tasks*, *heterogeneous platforms*, *master slave tasking*, *parallel algorithms*, *scheduling*.

Participants: Olivier Beaumont, Miroslaw Korzeniowski.

As already mentioned in section 3.4, makespan minimization turns out to be very difficult, even for simple homogeneous processors and links. Our objective is to lower the ambition of makespan minimization in order to build efficient scheduling algorithms for more realistic platform models. In our works, we usually adopt the so-called “one-port with overlap model”, where a processor can simultaneously send one message, receive one message, process one task, and contentions over communication links are taken into account. This requires a fine knowledge of the topology of the platform, but recently, some tools (like ENV and AlNEM) have been designed to build such platform models. An idea to circumvent the difficulty of makespan minimization is to lower the ambition of the scheduling objective. Instead of aiming at the absolute minimization of the execution time, why not consider asymptotic optimality ? After all, the number of tasks to be executed on the computing platform is expected to be very large: otherwise why deploy the corresponding application on computational grids ? This approach has been pioneered by Bertsimas and Gamarnik. The dramatic simplification of steady-state scheduling is to concentrate on steady-state operations ! The scheduling problem is relaxed in many ways. Initialization and clean-up phases are neglected. The initial integer formulation is replaced by a continuous or rational formulation. The precise ordering and allocation of tasks and messages are not required, at least in the first step. The main idea is to characterize the activity of each resource during each time-unit: which (rational) fraction is spent computing, which is spent receiving or sending to which neighbor. Such activity variables are gathered into a linear program, which includes conservation laws that characterize the global behavior of the system.

This approach has been applied with success to many scheduling problems. We have first considered very simple application models, such as master-slave tasking, where a processor initially holds all the data, and the makespan minimization counterpart has been studied. Generalizations, when some parallelism can be extracted within tasks, have been considered and the general case has been proven NP-Hard. The case of divisible tasks (perfect parallel tasks that can be arbitrarily divided) has been addressed in [44] in the case where return messages must be taken into account. More recently, we studied the case where several applications must be scheduled simultaneously on the same platform [43].

We have applied steady-state techniques to collective communication schemes, such as scatters, broadcasts, parallel prefix and multicasts. We have derived polynomial algorithms for broadcasts and scatters, both under one port bidirectional and unidirectional [3] models.

From the computational complexity point of view, considering steady state and throughput maximization instead of makespan minimization is both realistic and efficient in the case of large scale heterogeneous platforms. Nevertheless, as already noted, besides their heterogeneity, large scale distributed platforms exhibit some level of dynamicity. In the case of grid-like platforms, we can assume that the topology does not change during the execution of an application, but the performances of communication and processing resources may be affected by external load. In the case of peer to peer platforms, the topology itself may change during the execution.

These characteristics must change dramatically the algorithms used for scheduling both applications and communications on those platforms. In particular, it is not realistic to assume that the topology and the actual performances of all resources are centralized at a given point. This requires the design of decentralized algorithms for achieving good throughput, where nodes make their decision according to their current state and the states of their immediate neighbors. We already considered this constraint in [42] and our aim is to generalize this framework to all scheduling problems we already considered.

In order to achieve this goal, we recently concentrate on the solutions proposed by P2P community. Indeed, the tremendous success of peer-to-peer (P2P) applications for file sharing led to the design of a large number of dedicated protocols, that run in a fully distributed environment. These protocols support local decisions, and the P2P services (publication, search, node insertion, etc.) are supported by a (virtual) overlay network connecting the peers over the Internet. Up to some extent, the current P2P protocols are stable and fault-tolerant, as witnessed by their wide and intensive usage. Nevertheless, the P2P protocols have been initially designed for file sharing applications and also studied in the context of general purpose distributed applications. Yet, such protocols have not been optimized for scientific applications, neither they are adapted to sophisticated data-base applications. In particular, the type of request they accept is too limited to consider general purpose applications (such as independent tasks sharing files applications, that appear for instance in Monte Carlo simulations). In [65], we consider the extension of those protocols to range queries (instead of exact searches). Moreover, most of the protocols do not take resource performances (especially bandwidths) into account. Recently, Miroslaw Korzeniowski has been hired as INRIA post-doctorant and works on broadcast protocols that take network performances into account. This evolution is also the heart of the project proposal Cepage (http://www.labri.fr/perso/obeaumon/publis/cepage.pdf) that should be presented to INRIA Futurs Project Committee in 2007.

6.6. Computational steering environment for distributed numerical simulations

Keywords: *code coupling*, *computational steering*, *data redistribution*, *numerical simulation*, *scientific visualization*.

Participants: Olivier Coulaud, Aurélien Esnard, Nicolas Richart, Mathieu Souchaud.

Steering of a legacy simulation in astrophysics. This year, we have experimentally validated the EPSN framework with a legacy simulation in astrophysics, called Gadget2. It has demonstrated the interest of parallel visualization and rendering techniques to reduce the steering overhead and achieve better performance than for classical steering environment. In our case study, Gadget2 simulates the birth of a galaxy, which is represented by a gas cloud that collapses gravitationally until a central shock. The gas cloud is modelled by 1,000,000 particles distributed on 60 processes. The simulation has been deployed on the Grid'5000 cluster and connected to a visualization cluster with a 2 × 2 tiled-display. This experiment also validates the ability of EPSN to understand dynamic and complex data distribution. These results has been published in [6].

Placement approach for redistribution. We have introduced in [69] a new redistribution approach, called *placement*, that is well adapted to the context of $M \times N$ computational steering. In this case, the data distribution on the visualization code is not initially defined. This offers the opportunity for the redistribution layer to choose it at run-time in "the best way". Our strategy consists in the placement of the data elements, initially distributed on M simulation processes, to the N visualization processes. In order to equilibrate the number of elements on the visualization code and to minimize the number of messages, one simply realizes the intersection of the two distribution patterns, that results in the generation of $M + N - GCD(M, N)$ messages. In irregular cases, this requires to split the elements handled by a simulation process into several messages. For particle data, the split operator is quite trivial, while for unstructured meshes, it is defined thanks to graph partitioning techniques as those provided by Scotch or METIS. The RedGRID library has been extended to support this new redistribution approach, that we already use in EPSN.

In the context of the ANR MASSIM, we are now considering more complex data structures such as hierarchical grids with large amount of data.

Model for the steering of parallel-distributed simulations. The model that we have proposed in the EPSN framework can only steer efficiently SPMD simulations. A natural development is to consider more complex simulations such as coupled SPMD codes called M-SPMD (Multiple SPMD like multi-scale simulation for "crack-propagation") and client/server simulation codes. In order to steer these kinds of simulation, we have designed an extension to the Hierarchical Task Model (HTM), that affords to solve the coherency problem for such complex applications. In future works, we will implement our model in the EPSN framework and validate it with the multi-scale simulation for "crack-propagation"developed in the project.

Dynamic adaptation. We intend through the ARC COA to find a common approach for the dynamic adaptation and the computational steering. In this way, we have proposed a component model that afford to steering and adapt dynamically simulations. To validate this model we have integrated the EPSN environment into the dynamic adaptation framework called Dynaco, developed in the Paris project. It validates the idea that the steering and the dynamic adaptation are based on the same concepts.

7. Contracts and Grants with Industry

7.1. Research and development contracts

CEA research and development contracts:

- Parallel resolution of multifluid flows (Benjamin Braconnier, Boniface Nkonga);
- Numerical simulation of compressible multifluid flows (Rémi Abgrall, Michaël Papin);
- Simulation of multiscale multiphase flows (Rémi Abgrall and Vincent Perrier);
- Feasibility study of the new hybrid MPI – Threads version of the PaStiX parallel direct solver on the SMP supercomputer of CEA. Application to the electromagnetism code ODYSSEE. (Pascal Hénon, Pierre Ramet, Jean Roman);
- Numerical simulation of crack propagation in silica glass by coupling molecular dynamics and elasticity methods (Guillaume Anciaux, Olivier Coulaud, Jean Roman).

EDF research and development contracts:

- Application of a domain decomposition method to the neutronic SPn equations (Bruno Lathuilière, Pierre Ramet, Jean Roman);
- Improvment of the computation tool performances used for neutronic simulation of the EDF cores (Guilhem Caramel, Pierre Ramet, Jean Roman).

8. Other Grants and Activities

8.1. Regional initiatives

8.1.1. CNES – EADS/EXPERT

Participants: Rémi Abgrall (leader of the project), Pascal Hénon, Pierre Ramet, Mario Ricchiuto, Jean Roman, Cédric Tavé.

Grant: Conseil Régional d'Aquitaine, CNES and EADS – EXPERT project
Dates: 2004 – 2007
Overview: The objective of this work is to upgrade the numerical schemes in the aerodynamic modules of the ONERA code CEDRE using the know–how we have developed in residual distribution schemes. The main difficulty is to adapt these methods to the data structure of CEDRE. The residual distribution schemes are tuned for cell vertex data structure while CEDRE works with cell centered data structures. The scientific objective of this grant is to provide a bridge between residual distribution schemes and discontinuous Galerkin ones.

8.2. National initiatives

8.2.1. Plasma: modeling of magnetized plasmas

Participants: Rémi Abgrall, Olivier Coulaud, Guillaume Latu, Pierre Ramet, Mario Ricchiuto.

Grant: ARC INRIA
Dates: 2005-2006
Partners: CALVI (INRIA LORIA - leader of the project), MC2 (INRIA Futurs Bordeaux), SIMPAF (INRIA Futurs Lille), MIP (Toulouse), CEA Caradarache
Overview: The description of magnetized plasmas uses a hierarchy of models; this leads to several open problems: modeling and role of adimensionnalised parameters, mathematical analysis of the models and their asymptotic behavior when some parameters tends to infinity, numerical simulation of these (simplified) models. The role of this ARC is to cover this range of problems, from the analysis to the numerical simulation.
Web: http://www-irma.u-strasbg.fr/annexes/arc_plasma/

8.2.2. COA: a computational steering environment for distributed numerical simulations

Participants: Olivier Coulaud (leader of the project), Aurélien Esnard, Nicolas Richart.

Grant: ARC INRIA
Dates: 2005-2006
Partners:PARIS (INRIA Rennes), Projet JACQUARD (INRIA Futurs Lille)
Overview: This 2-year project is funded by the INRIA Cooperative Research Initiative (ARC) whose partners are the PARIS, Jacquard and `ScAlApplix` Project-Teams. Its objective is to design an experimental platform allowing dynamic adaptation and steering of distributed numerical simulation applications using aspect weaving technics on top of component models.
Web: http://www.irisa.fr/paris/COA/

8.2.3. Sire: computing simulation of enzymatic systems: from structural to functional aspects

Participant: Olivier Coulaud.

Grant: ACI IMPIO ("Action Concertée Incitative Informatique, Mathématiques, Physique en Biologie Moléculaire" – French Ministry of Research)
Dates: 2004 – 2006
Partners: CBT and MAEM (UHP Nancy 1, CNRS)
Overview: The goal of this action is to study the using of linear scaling algorithms in order to understand the behavior of Methionine synthase reductase enzymes.
Web: http://www.cbt.uhp-nancy.fr/sire/

8.2.4. MASSIM: development of a software environment for interactively processing and visualizing complex large scale data coming from numerical simulations

Participants: Olivier Coulaud, Aurélien Esnard, Guillaume Latu, Nicolas Richart.

Grant: ANR MMSA - ARA MAsses de données
Dates: 2005 – 2008
Partners: IRMA (Strasbourg, UMR 7001)), LSIIT (leader of the project, Strasbourg, UMR 7005)

Overview: Numerical simulation is a continuously growing area, especially with the increasing computational power of current computer technology, thus covering larger and larger scientific application fields. But at these days, monitoring tools are still seriously lacking, since developers and users desire more and more to get faster and faster feedbacks of the simulation results. In this project, we are interested in large scale simulations dealing with complex data (multivariate and multidimensional). Our aim is to realize a plate-form / framework to couple parallel and distributed simulations, like in GRID'5000, with an interactive monitoring and visualization system. This plate-form will be validated on two types of large scale applications: plasma and fracture propagation simulation using multi-scale approaches. For these applications, the simulation codes are definitely very complex and need some highly efficient tools to represent the large amount of data, to redistribute the data using visualization and to control and validate the corresponding computation algorithms. Since, results may be multivariate and multidimensional, they need also specific data exploration and visualization tools.
Web: http://www.labri.fr/epsn/

8.2.5. NUMASIS: adaptation and optimization of the applicatives performances on NUMA architectures

Participants: Mathieu Faverge, Abdou Guermouche, Pascal Hénon, Pierre Ramet, Jean Roman.

Grant: ANR-05-CIGC-002
Dates: 2006 – 2009
Partners: Bull, Total, BRGM, CEA, ID-Imag (leader of the project), PARIS (IRISA), Runtime (INRIA Futurs Bordeaux).
Overview: The multiprocessor machines of tomorrow will rely on an NUMA architecture introducing multiple levels of hierarchy into computers (multi-modules, chips multi-body, multithreading material, etc). To exploit these architectures, parallel applications must use powerful runtime supports making possible the distribution of execution and data streams without compromising their portability. Project NUMASIS proposes to evaluate the functionalities provided by the current systems, to apprehend the limitations, to design and implement new mechanisms of management of the processes, data and communications within the basic softwares (operating system, middleware, libraries). The target algorithmic tools that we retained are parallel linear sparse solvers with application to seismology.
Web: http://numasis.gforge.inria.fr/

8.3. International initiatives

8.3.1. ADIGMA: a european project on the development of adaptive higher order variational methods for aerospace applications

Participants: Rémi Abgrall (leader of the project), Jérémie Gaidamour, Pascal Hénon, Adam Larat, Pierre Ramet, Mario Ricchiuto, Jean Roman.

Grant: European Commission
Dates: 2006-2009
Partners: AIRBUS F et AIRBUS D, DASSAULT, ALENIA, DLR, ONERA, NLR, ARA, VKI, INRIA, Nankin University, Universities of Stuttgart, Bergame, Twente, Nottingham, Swansea, Charles (Prague), Varsovie, CENAERO, ENSAM Paris)

Overview: Computational Fluid Dynamics is a key enabler for meeting the strategic goals of future air transportation. However, the limitations of today numerical tools reduce the scope of innovation in aircraft development, keeping aircraft design at a conservative level. Within the 3rd Call of the 6th European Research Framework Programme, the strategic target research project ADIGMA has been initiated. The goal of ADIGMA is the development and utilization of innovative adaptive higher-order methods for the compressible flow equations enabling reliable, mesh independent numerical solutions for large-scale aerodynamic applications in aircraft design. A critical assessment of the newly developed methods for industrial aerodynamic applications will allow the identification of the best numerical strategies for integration as major building blocks for the next generation of industrial flow solvers. In order to meet the ambitious objectives, a partnership of 22 organizations from universities, research organizations and aerospace industry from 10 countries with well proven expertise in CFD has been set up guaranteeing high level research work with a clear path to industrial exploitation.

9. Dissemination

9.1. Participation to the Scientific Community

Rémi Abgrall is scientific associate editor of the international journals “Mathematical Modelling and Numerical Analysis”, “Computer and Fluids”, “Journal of Computational Physics”, “Journal of Scientific Computing” and “Journal of Computing Science and Mathematics”. He is member of the scientific committee of the international conference ICCFD. He is member of the scientific committee of the international conference ECOMAS2006. He is also member of the scientific committee of CERFACS.
Olivier Beaumont is member of the scientific committee of the following international conferences: IPDPS’06, HeteroPar’06, PMAA’06, ICPADS’06. He is the program co-chair of Renpar’06 (French conference on parallel algorithms).
Olivier Coulaud has been member of the scientific committee of the international conference VECPAR’06.
Pascal Hénon has been member of the scientific committee of Renpar’06.
François Pellegrini has been member of program committee of the Third High-Performance Grid Computing Workshop, HPCG-06.
Pierre Ramet is member of the commitee of Researchers at CINES for the thematic number 6 (mathematics).
Jean Roman is President of the Project Committee of INRIA Futurs and member of the National Evaluation Committee of INRIA. He has been member of the scientific committee of the national conference RenPar’06. He has been member of the ANR steering committee for the “Intensive Computation and Simulation” theme. He has been member of the scientific commitee of the international conference PMAA’06, and is co-editor of a special issue for the previous PMAA in the international journal “Parallel Computing”.

9.2. Teaching

In complement of the normal teaching activity of the university members and of ENSEIRB members, Olivier Coulaud and Pascal Hénon teach at ENSEIRB (computer science engineering school).
François Pellegrini has given a lecture at the CEA-EDF-INRIA school on “Conception of scientific high performance applications” (July 2006, at St Lambert des Bois).

O. Coulaud, P. Hénon and J. Roman have given three lectures at the CEA-EDF-INRIA school on “Intensive scientific computation” (November 2006, at INRIA Rocquencourt).

10. Bibliography

Major publications by the team in recent years

[1] R. ABGRALL. *Essentially non-oscillatory residual distribution schemes for hyperbolic problems*, in "J. Comput. Phys.", vol. 214, n^{o} 2, 2006, p. 773–808.

[2] G. ANCIAUX, O. COULAUD, J. ROMAN. *High Performance Multiscale Simulation for Crack Propagation*, in "Proceedings of HPSEC'2006", vol. ICPPW'06, IEEE Computer Society Press, August 2006, p. 473-480.

[3] O. BEAUMONT, L. MARCHAL, Y. ROBERT. *Complexity results for collective communications on heterogeneous platforms*, in "Int. Journal of High Performance Computing Applications", vol. 20, n^o 1, 2006, p. 5–17.

[4] C. BERTHON. *Robustness of MUSCL schemes for 2D unstructured meshes*, in "J. Comp. Physics", vol. 218, 2006, p. 495-509.

[5] C. CHEVALIER, F. PELLEGRINI. *Improvement of the Efficiency of Genetic Algorithms for Scalable Parallel Graph Partitioning in a Multi-Level Framework*, in "Proceedings of Euro-Par 2006, Dresden, Germany", LNCS, vol. 4128, Springer Verlag, August 2006, p. 243-252.

[6] A. ESNARD, N. RICHART, O. COULAUD. *A Steering Environment for Online Parallel Visualization of Legacy Parallel Simulations*, in "Proceedings of the 10th International Symposium on Distributed Simulation and Real-Time Applications (DS-RT 2006), Torremolinos, Malaga, Spain", vol. ds-rt, IEEE Press, October 2006, p. 7-14.

[7] A. GUERMOUCHE, J.-Y. L'EXCELLENT. *Constructing Memory-minimizing Schedules for Multifrontal Methods*, in "ACM Transactions on Mathematical Software", vol. 32, n^o 1, 2006, p. 17–32.

[8] P. HÉNON, Y. SAAD. *A Parallel Multistage ILU Factorization based on a Hierarchical Graph Decomposition*, in "SIAM Journal of Scientific Computing", to appear, 2006.

[9] M. RICCHIUTO, R. ABGRALL, H. DECONINCK. *Application of conservative residual distribution schemes to the solution of the shallow water equations on unstructured meshes*, in "J. Comput. Phys.", to appear, 2006.

[10] G. TESSIER, J. ROMAN, G. LATU. *Hybrid MPI-Thread Implementation on a Cluster of SMP Nodes of a Parallel Simulator for the Propagation of Powdery Mildew in a Vineyard*, in "High Performance Computing and Communications, Second International Conference, HPCC 2006, Munich, Germany, September 13-15, 2006, Proceedings", LNCS, vol. 4208, Springer Verlag, September 2006, p. 833-842.

Year Publications

Books and Monographs

[11] O. BEAUMONT, V. BOUDET (editors). *Perpignan, Octobre 2006*, Actes de Renpar 2006, 2006.

[12] H. DECONINCK, M. RICCHIUTO (editors). , von Karman Institute Lecture Series, ISBN: 2-930389-63-X, vol. VKI LS 06-01, von Karman Institute for Fluid Dynamics, 2006.

Doctoral dissertations and Habilitation theses

[13] C. BERTHON. *Numerical Simulations of some hyperbolic systems*, Habilitation à Diriger des Recherches, Université Bordeaux 1, October 2006.

[14] P. FORTIN. *Algorithmique hiérarchique parallèle haute performance pour les problèmes à N-corps*, PhD Thesis, Université Bordeaux 1, November 2006.

Articles in refereed journals and book chapters

[15] R. ABGRALL. *Essentially non-oscillatory residual distribution schemes for hyperbolic problems*, in "J. Comput. Phys.", vol. 214, n^{o} 2, 2006, p. 773–808.

[16] R. ABGRALL. *Residual distribution schemes : current status and future trends*, in "Comp. and Fluids", to appear, 2006.

[17] R. ABGRALL, F. MARPEAU. *Residual distribution schemes on quadrilateral meshes*, in "J. Sci. Comput", to appear, 2006.

[18] R. ABGRALL, V. PERRIER. *Asymptotic expansion of a multiscale numerical scheme for compressible multiphase flow*, in "Multiscale Model. Simul.", vol. 5, n^{o} 1, 2006, p. 84–115 (electronic).

[19] P. R. AMESTOY, A. GUERMOUCHE, J.-Y. L'EXCELLENT, S. PRALET. *Hybrid scheduling for the parallel solution of linear systems*, in "Parallel Computing", vol. 32, n^{o} 2, 2006, p. 136–156.

[20] O. BEAUMONT, L. MARCHAL, Y. ROBERT. *Complexity results for collective communications on heterogeneous platforms*, in "Int. Journal of High Performance Computing Applications", vol. 20, n^{o} 1, 2006, p. 5–17.

[21] C. BERTHON. *Numerical approximations of the 10-moment Gaussian closure*, in "Math. Comp.", vol. 75, 2006, p. 1809-1831.

[22] C. BERTHON. *Robustness of MUSCL schemes for 2D unstructured meshes*, in "J. Comp. Physics", vol. 218, 2006, p. 495-509.

[23] C. BERTHON. *Why the MUSCL-Hancock scheme is* L^1*-stable*, in "Numer. Math.", vol. 104, 2006, p. 27-46.

[24] C. BERTHON, M. BREUSS, M.-O. TITEUX. *A relaxation scheme for pressureless gas dynamics*, in "Num. Method Partial Diff. Equations", vol. 22, 2006, p. 484-505.

[25] C. BERTHON, P. CHARRIER, B. DUBROCA. *An HLLC Scheme to Solve The M1 Model of Radiative Transfer in Two Space Dimensions*, in "J. Sci. Comput.", to appear, 2006.

[26] C. BERTHON, F. COQUEL. *Multiple solutions for compressible turbulent flow models*, in "Comm. Math. Sciences", vol. 4, 2006, p. 497-511.

[27] C. BERTHON, F. COQUEL. *Nonlinear projection methods for multi-entropies Navier-Stokes systems*, in "Math. Comput.", to appear, 2006.

[28] C. BERTHON, B. NKONGA. *Multifluid numerical approximations based on multipressure formulation*, in "Comp. Fluids", vol. 36, 2006, p. 467-479.

[29] O. COULAUD, P. FORTIN, J. ROMAN. *High-performance BLAS formulation of the Adaptive Fast Multipole Method*, in "Journal of Supercomputing", accepted under revision, 2006.

[30] C. DOBRZYNSKI, P. FREY, B. MOHAMMADI, O. PIRONNEAU. *Couplage et adaptation de maillage anisotrope pour des simulations de flux d'air dans des géométries complexes*, in "Revue européenne des éléments finis", to appear, 2006.

[31] C. DOBRZYNSKI, P. FREY, B. MOHAMMADI, O. PIRONNEAU. *Fast and accurate simulations of air-cooled structures*, in "Comput. Methods Appli. Mech. Engrg.", vol. 195, 2006, p. 3168-3180.

[32] A. GUERMOUCHE, J.-Y. L'EXCELLENT. *Constructing Memory-minimizing Schedules for Multifrontal Methods*, in "ACM Transactions on Mathematical Software", vol. 32, n^{o} 1, 2006, p. 17–32.

[33] P. HÉNON, Y. SAAD. *A Parallel Multistage ILU Factorization based on a Hierarchical Graph Decomposition*, in "SIAM Journal of Scientific Computing", to appear, 2006.

[34] S. T. MUNKEJORD, M. PAPIN. *The effect of interfacial pressure in the discrete-equation multiphase model*, in "Comp. and Fluids", to appear, 2006.

[35] M. RICCHIUTO, R. ABGRALL, H. DECONINCK. *Application of conservative residual distribution schemes to the solution of the shallow water equations on unstructured meshes*, in "J. Comput. Phys.", to appear, 2006.

Publications in Conferences and Workshops

[36] R. ABGRALL, S. KARNI. *A relaxation scheme for the two–layers shallow water system*, in "Proceeding of HYP2006", D. SERRE (editor). , 2006.

[37] R. ABGRALL, M. RICCHIUTO, N. VILLEDIEU-LIGOUT, C. TAVÉ, H. DECONINCK. *Very high order residual distribution on triangular grids*, in "ECCOMAS 2006, European Conference on Computational Fluid Dynamics , Egmond aan Zee (The Netherlands)", P. WESSELING, E. ONATE, J. PÉRIAUX (editors). , September 2006.

[38] R. ABGRALL, C. TAVÉ. *Construction of very high order residual distribution schemes*, in "Proceeding of the 6th International Conference on CFD", H. DECONINCK, E. DICK (editors). , Springer Verlag, 2006.

[39] E. AGULLO, A. GUERMOUCHE, J.-Y. L'EXCELLENT. *A Preliminary Out-of-core Extension of a Parallel Multifrontal Solver*, in "Proceedings of Euro-Par 2006, Dresden, Germany", Lecture Notes in Computer Science, vol. 4128, Springer Verlag, August 2006, p. 1053-1063.

[40] P. AMESTOY, I. DUFF, A. GUERMOUCHE, T. SLAVOVA. *A Preliminary Analysis of the Out-of-Core Solution Phase of a Parallel Multifrontal Approach*, 4th International Workshop on Parallel Matrix Algorithms and Applications (PMAA'06),September 7-9, 2006, IRISA, Rennes, France, sept 2006.

[41] G. ANCIAUX, O. COULAUD, J. ROMAN. *High Performance Multiscale Simulation for Crack Propagation*, in "Proceedings of HPSEC'2006", vol. ICPPW'06, IEEE Computer Society Press, August 2006, p. 473-480.

[42] C. BANINO, O. BEAUMONT, L. NATVIG. *Master-Slave Tasking on Asymmetric Networks*, in "Proceedings of Euro-Par 2006, Dresden, Germany", LNCS, vol. 4128, Springer Verlag, August 2006, p. 437-447.

[43] O. BEAUMONT, L. CARTER, J. FERRANTE, A. LEGRAND, L. MARCHAL, Y. ROBERT. *Centralized Versus Distributed Schedulers for Multiple Bag-of-Task Applications*, in "International Parallel and Distributed Processing Symposium IPDPS'2006", to appear, IEEE Computer Society Press, 2006.

[44] O. Beaumont, L. Marchal, V. Rehn, Y. Robert. *FIFO scheduling of divisible loads with return messages under the one-port model*, in "Heterogeneous Computing Workshop HCW'2006", to appear, IEEE Computer Society Press, 2006.

[45] C. Berthon. *A relaxation scheme to approximate the 10-moment Gaussian closure*, in "European Conference on Computational Fluid dynamics, ECCOMAS CFD 2006", P. Wesseling, E. Onate, J. Periaux (editors). , 2006.

[46] C. Berthon. *A stable gradient reconstruction for the MUSCL schemes applied to systems of conservation laws*, in "Special Issue: Numerical Simulation of Complex and Multiphase Flows", T. Flow, Combustion (editors). , vol. 76, 2006, p. 343-351.

[47] C. Berthon, B. Braconnier, J. Claudel, B. Nkonga. *A relaxation method for a two phase flow with surface tension*, in "Numerical mathematics and advanced applications", Springer Verlag, 2006, p. 843-851.

[48] B. Braconnier, B. Nkonga, M. Papin, P. Ramet, M. Ricchiuto, J. Roman, R. Abgrall. *Efficient solution technique for low Mach number compressible multiphase problems*, in "Proceedings of PMAA'2006, Rennes, France", 2006.

[49] C. Chevalier, F. Pellegrini. PT-Scotch*: A tool for efficient parallel graph ordering*, in "Proceedings of PMAA'2006, IRISA, Rennes, France", October 2006, http://www.labri.fr/~pelegrin/papers/scotch_parallelordering_pmaa.pdf.

[50] C. Chevalier, F. Pellegrini. *Improvement of the Efficiency of Genetic Algorithms for Scalable Parallel Graph Partitioning in a Multi-Level Framework*, in "Proceedings of Euro-Par 2006, Dresden, Germany", LNCS, vol. 4128, Springer Verlag, August 2006, p. 243-252.

[51] A. Esnard, N. Richart, O. Coulaud. *A Steering Environment for Online Parallel Visualization of Legacy Parallel Simulations*, in "Proceedings of the 10th International Symposium on Distributed Simulation and Real-Time Applications (DS-RT 2006), Torremolinos, Malaga, Spain", vol. ds-rt, IEEE Press, October 2006, p. 7-14.

[52] J. Gaidamour, P. Hénon, J. Roman, Y. Saad. *An hybrid direct-iterative solver based on the Schur complement approach*, in "Compstat, 8th Workshop of the ERCIM Working group, Salernes, Italie", September 2006.

[53] P. Hénon, P. Ramet, J. Roman. *On finding approximate supernodes for an efficient ILU(k) factorization*, in "Proceedings of PMAA'2006, Rennes, France", September 2006, http://www.labri.fr/~ramet/restricted/pmaa06a.pdf.gz.

[54] P. Hénon, P. Ramet, J. Roman. *Partitioning and Blocking Issues for a Parallel Incomplete Factorization*, in "Proceedings of PARA'2006, Umea, Sweden", to appear, Springer Verlag, June 2006.

[55] B. Nkonga. *Superconsistency property for nonconservative operator approximation : application to compressible multifluid flow.*, in "Proceedings of The 2nd International Conference on Scientific Computing and Partial Differential Equations and The First East Asia SIAM Symposium, Hong Kong.", 2006.

[56] M. Ricchiuto, R. Abgrall. *Stable and convergent RD schemes for steady and time dependent conservation laws*, in "Proc. of the Conf. Topical problems of fluid mechanics, Prague (CZ)", 2006.

[57] M. RICCHIUTO, R. ABGRALL. *Stable and convergent residual distribution for time-dependent conservation laws*, in "ICCFD4, 4th International Conference on Computational Fluid Dynamics", Springer Verlag, 2006.

[58] G. TESSIER, J. ROMAN, G. LATU. *Hybrid MPI-Thread Implementation on a Cluster of SMP Nodes of a Parallel Simulator for the Propagation of Powdery Mildew in a Vineyard*, in "High Performance Computing and Communications, Second International Conference, HPCC 2006, Munich, Germany, September 13-15, 2006, Proceedings", LNCS, vol. 4208, Springer Verlag, September 2006, p. 833-842.

[59] N. VILLEDIEU-LIGOUT, M. RICCHIUTO, H. DECONINCK. *High-order residual distribution : discontinuity capturing crosswind dissipation and diffusion*, in "ICCFD4, 4th International Conference on Computational Fluid Dynamics", to appear, Springer Verlag, 2006.

Miscellaneous

[60] R. ABGRALL, S. KARNI. *Two–shallow water systems : a relaxation approach*, submitted, 2006.

[61] E. AGULLO, A. GUERMOUCHE, J.-Y. L'EXCELLENT. *Étude mémoire d'une méthode multifrontale parallèle hors-mémoire16iemes Rencontres Francophones en Parallélisme (RenPar'16), Perpignan, France*, 2006.

[62] P. AMESTOY, I. DUFF, A. GUERMOUCHE, T. SLAVOVA. *Efficient triangular solution in sparse out-of-core solvers*, The Sparse Days Meeting at CERFACS, June 15-16, 2006, Toulouse, France, June 2006.

[63] G. ANCIAUX. *Un simulateur HPC des matériaux par couplage de la dynamique moléculaire avec les éléments finis17ème Rencontres francophones du parallélisme (RenPar'17)*, 2006.

[64] O. BEAUMONT, P. DUTOT, A. GUERMOUCHE, D. TRYSTRAM. *Scheduling trees of tasks for parallel mutifrontal methods*, Workshop on Scheduling Algorithms for new Emerging Applications, CIRM, Marseille, France, June 2006.

[65] O. BEAUMONT, ANNE-MARIE. KERMARREC, L. MARCHAL, É. RIVIÈRE. *VoroNet, réseau objet-à-objet sur le modèle petit monde. Actes de la cinquième Conférence Francophone en Systèmes d'Exploitation (CFSE), Perpignan, France*, October 2006.

[66] C. CHEVALIER, F. PELLEGRINI. PT-SCOTCH *: Un outil pour la renumérotation parallèle efficace de grands graphes dans un contexte multi-niveauxProc. RenPar'17, Perpignan, France*, October 2006, http://www.labri.fr/~pelegrin/papers/scotch_parallelordering_renpar.pdf.

[67] O. COULAUD, P. FORTIN, J. ROMAN. *High performance BLAS formulation of the multipole-to-local operator in the Fast Multipole Method*, submitted to Journal of Computational Physics, 2006.

[68] H. DECONINCK, M. RICCHIUTO. *Residual Distribution schemes : foundations and analysis.Encyclopedia of Computational Mechanics*, John Wiley and Sons, Ltd, 2006.

[69] A. ESNARD. *Approche placement pour la redistribution de données irrégulières. 17ème Rencontres francophones du parallélisme (RenPar'17)*, 2006.

[70] A. GUERMOUCHE. *Parallel multifrontal method with out-of-core techniques*, Workshop On State-of-the-art In Scientific And Parallel Computing (PARA'06), 2006, Umea, Sweden, June 2006.

[71] M. RICCHIUTO, N. VILLEDIEU-LIGOUT, R. ABGRALL, H. DECONINCK. *Construction of uniformly accurate residual distribution schemes for advection–diffusion problems*, submitted to J. Computational and Applied mechanics, 2006.

References in notes

[72] C. BERTHON, B. NKONGA. *Numerical model of a compressible multi-fluid fluctuating flow*, in "Inter. J. finite volumes", 2005.

[73] A. CALONNEC, G. LATU, J.-M. NAULIN, J. ROMAN, G. TESSIER. *Parallel Simulation of the Propagation of Powdery Mildew in a Vineyard*, in "11th International Euro-Par Conference", Lecture Notes in Computer Science, vol. 3648, Springer Verlag, September 2005, p. 1254 – 1263.

[74] C. CHEVALIER, F. PELLEGRINI. PT-SCOTCH*: A tool for efficient parallel graph ordering*, in "Proc. PMAA'2006, Rennes, France", October 2006, http://www.labri.fr/~pelegrin/papers/scotch_parallelordering_pmaa.pdf.

[75] O. COULAUD, P. FORTIN, J. ROMAN. *High-performance BLAS formulation of the Adaptive Fast Multipole Method*, in "Advances in Computational Methods in Sciences and Engineering 2005, Selected Papers from the International Conference of Computational Methods in Sciences and Engineering (ICCMSE 2005)", T. SIMOS, G. MAROULIS (editors). , Special Volume of the Lecture Series on Computer and Computational Sciences, vol. 4B, VSP/Brill, 2005, p. 1796-1799.

[76] P. FORTIN. *Multipole-to-local operator in the Fast Multipole Method: comparison of FFT, rotations and BLAS improvements*, Research Report, n^{o} 5752, INRIA, 2005, https://hal.inria.fr/inria-00070267.

[77] P. HÉNON, F. PELLEGRINI, P. RAMET, J. ROMAN. *Blocking Issues for an Efficient Parallel Block ILU Preconditioner*, in "International SIAM Conference On Preconditioning Techniques For Large Sparse Matrix Problems In Scientific And Industrial Applications, Atlanta, USA", May 2005, http://www.labri.fr/~ramet/restricted/pred2005.ps.gz.

[78] P. HÉNON, F. PELLEGRINI, P. RAMET, J. ROMAN, Y. SAAD. *Applying parallel direct solver skills to build robust and highly performant preconditioners*, in "PARA'04, Workshop on state-of-the-art in scientific computing, Copenhague, Danemark", June 2004.

[79] P. HÉNON, F. PELLEGRINI, P. RAMET, J. ROMAN, Y. SAAD. *High Performance Complete and Incomplete Factorizations for Very Large Sparse Systems by using Scotch and PaStiX softwares*, in "Eleventh SIAM Conference on Parallel Processing for Scientific Computing, San Francisco, USA", February 2004.

[80] P. HÉNON, P. RAMET, J. ROMAN. *A Blockwise Algorithm for Parallel Incomplete Cholesky Factorization*, in "Proceedings of PMAA'2004, Marseille, France", October 2004.

[81] P. HÉNON, P. RAMET, J. ROMAN. *On using an hybrid MPI-Thread programming for the implementation of a parallel sparse direct solver on a network of SMP nodes*, in "Proceedings of Sixth International Conference on Parallel Processing and Applied Mathematics, Workshop HPC Linear Algebra, Poznan, Pologne,", September 2005, http://www.labri.fr/~ramet/restricted/ppam05.ps.gz.

[82] G. TESSIER. *Simulation parallèle de la propagation de l'oïdium dans une parcelle de vigne.*, in "RenPar'16", April 2005.