



RESEARCH CENTER
Bordeaux - Sud-Ouest

FIELD

Activity Report 2015

Section Software

Edition: 2016-03-21

1. CAGIRE Team	4
2. CARDAMOM Team	7
3. CARMEN Team	13
4. CQFD Project-Team	14
5. FLOWERS Project-Team	16
6. GEOSTAT Project-Team	27
7. HIEPACS Project-Team	28
8. LFANT Project-Team	33
9. MAGIQUE-3D Project-Team	36
10. MANAO Project-Team	38
11. MEMPHIS Team	41
12. MNEMOSYNE Project-Team	43
13. Monc Team	45
14. PHOENIX Project-Team	47
15. PLEIADE Team	54
16. POSET Team	56
17. POTIOC Project-Team	57
18. REALOPT Project-Team	58
19. SISTM Project-Team	59
20. STORM Team	60
21. TADAAM Team	64

CAGIRE Team

6. New Software and Platforms

6.1. AeroSol

Participants: Simon Delmas [Cagire], Benjamin Lux [Cagire], Nikolaos Pattakos [Cardamom], Vincent Perrier [Cagire, correspondent], Mario Ricchiuto [Cardamom].

Developed since 2011 by V. Perrier in partnership with the Cardamom Inria team, the AeroSol library is a high order finite element library written in C++. The code design has been carried for being able to perform efficient computations, with continuous and discontinuous finite element methods on hybrid and possibly curvilinear meshes.

The work of the Cardamom team is focused on continuous finite element methods, while we focus on discontinuous Galerkin methods. However, everything is done for sharing the largest possible part of code. The distribution of the unknowns is made with the software PaMPA, first developed within the Inria teams Bacchus and Castor, and currently maintained in the Tadaam team.

The generic features of the library are

- **High order.** It can be theoretically any order of accuracy, but the finite element basis, and quadrature formula are implemented for having up to a fifth order of accuracy.
- **Hybrid and curvilinear meshes.** AeroSol can deal with up to fifth order conformal meshes composed of lines, triangles, quadrangles, tetrahedra, hexahedra, prism, and pyramids.
- **Continuous and discontinuous discretization.** AeroSol deals with both continuous and discontinuous finite element methods.

We would like to emphasize three assets of this library:

- **Its development environment** For allowing a good collaborative work and a functional library, a strong emphasis has been put on the use of modern collaborative tools for developing our software. This includes the active use of a repository, the use of CMake for the compilation, the constant development of unitary and functional tests for all the parts of the library (using CTest), and the use of the continuous integration tool Jenkins for testing the different configurations of AeroSol and its dependencies. Efficiency is regularly tested with direct interfacing with the PAPI library or with tools like scalasca.
- **Its genericity** A lot of classes are common to all the discretization, for example classes concerning I/O, finite element functions, quadrature, geometry, time integration, linear solver, models and interface with PaMPA. Adding simple features (e.g. models, numerical flux, finite element basis or quadrature formula) can be easily done by writing the class, and declaring its use in only one class of the code.
- **Its efficiency** This modularity is achieved by means of template abstraction for keeping good performances. Dedicated efficient implementation, based on the data locality of the discontinuous Galerkin method has been developed. As far as parallelism is concerned, we use point-to-point communications, the HDF5 library for parallel I/O. The behavior of the AeroSol library at medium scale (1000 to 2000 cores) was studied in [19].

The AeroSol project fits with the first axis of the Bordeaux Sud Ouest development strategy, which is to build a coherent software suite scalable and efficient on new architectures, as the AeroSol library relies on several tools developed in other Inria teams, especially for the management of the parallel aspects.

At the end of 2014, AeroSol had the following features:

- **Development environment** Use of CMake for compilation (gcc, icc and xlc), CTest for automatic tests and memory checking, lcov and gcov for code coverage reports. Development of a CDash server for collecting the unitary tests and the memory checking. Beginning of the development of an interface for functional tests. Optional linking with HDF5, PAPI, with dense small matrices libraries (BLAS, Eigen)
- **In/Out** Link with the XML library for handling with parameter files. Parallel reader for GMSH, with an embedded geometrical pre-partitioner. Writer on the VTK-ASCII legacy format (cell and point centered). Parallel output in vtu and pvtu (Paraview) for cell-centered visualization, and XDMF/HDF5 format for both cell and point centered visualization. Ability of saving the high order solution and restarting from it. Computation of volumic and probe statistics. Ability of saving averaged layer data in quad and hexa meshes. Ability of defining user defined output visualization variables.
- **Quadrature formula** up to 11th order for Lines, Quadrangles, Hexaedra, Pyramids, Prisms, up to 14th order for tetrahedron, up to 21st order for triangles. Gauss-Lobatto type quadrature formula for lines, triangles, quadrangles and hexaedra.
- **Finite elements** up to fourth degree for Lagrange finite elements and hierarchical orthogonal finite element basis (with Dubiner transform on simplices) on lines, triangles, quadrangles, tetrahedra, prisms, hexaedra and pyramids. Finite element basis that are interpolation basis on Gauss-Legendre points for lines, quadrangles, and hexaedra, and triangle (only 1st and 2nd order).
- **Geometry** Elementary geometrical functions for first order lines, triangles, quadrangles, prisms, tetrahedra, hexaedra and pyramids. Handling of high order meshes.
- **Time iteration** explicit Runge-Kutta up to fourth order, explicit Strong Stability Preserving schemes up to third order. Optimized CFL time schemes: SSP(2,3) and SSP(3,4). CFL time stepping. Implicit integration with BDF schemes from 2nd to 6th order Newton method for stationary problems. Implicit unstationary time iterator non consistent in time for stationary problems. Implementation of in house GMRES and conjugate gradient based on Jacobian free iterations.
- **Linear Solvers** Link with the external linear solver UMFPack, PETSc and MUMPS. Internal solver for diagonal and block-diagonal matrices.
- **Memory handling** discontinuous and continuous, sequential and parallel discretizations based on PaMPA for generic meshes, including hybrid meshes.
- **Models** Perfect gas Euler system, real gas Euler system (template based abstraction for a generic equation of state), scalar advection, Waves equation in first order formulation, generic interface for defining space-time models from space models. Diffusive models: isotropic and anisotropic diffusion, compressible Navier-Stokes. Scalar advection-diffusion model.
- **Numerical schemes** Continuous Galerkin method for the Laplace problem (up to fifth order) with non consistent time iteration or with direct matrix inversion. Explicit and implicit discontinuous Galerkin methods for hyperbolic systems, diffusive and advection-diffusion problems. Beginning of optimization by stocking the geometry for advection problems. SUPG and Residual distribution schemes. Optimization of DG schemes for advection-diffusion problems: stocking of the geometry and use of BLAS for all the linear phases of the scheme.
- **Numerical fluxes** Centered fluxes, exact Godunov' flux for linear hyperbolic systems, and Lax-Friedrich flux. Riemann solvers for Low Mach flows. Numerical flux accurate for steady and unsteady computations.
- **Boundary conditions** Periodic boundary conditions, time-dependent inlet and outlet boundary conditions. Adiabatic wall and isothermal wall. Steger-Warming based boundary condition.
- **Parallel computing** Mesh redistribution, computation of Overlap with PaMPA. Collective asynchronous communications (PaMPA based). Asynchronous point to point communications. Tests on the cluster Avakas from MCIA, and on Mésocentre de Marseille, and PlaFRIM. Tier-1 Turing (Blue-Gene).

- **C++/Fortran interface** Tests for binding fortran with C++.
- **Instrumentation** Aerosol can give some traces on memory consumption/problems with an interfacing with the PAPI library. Tests have also been performed with VTUNE and TAU. Tests with Maqao and Scalasca (VIHPS workshop).
- **Validation** Poiseuille, Taylor-Green vortex. Laplace equation on a ring and Poiseuille flow on a ring. Implementation of volumic forcing based on wall dissipation.

In 2015, N. Pattakos was hired in the team Cardamom, in order to improve the code architecture and for easing the installation of the library. The following features were added or improved:

- **Development environment** The use of CMake was strongly improved, which induced also easier test launching. Documentation, code cleaning and refactorization have also been led. The shared project of Plafrim was updated, and so was the joint Aerosol/Scotch/PaMPA project on the continuous integration platform. Integration of SPack for handling dependencies has begun. Interface with ESSL was fixed.
- **Multigrid** Development of p -multigrid methods. This includes also the possibility of beginning a computation with an order and to decrease or increase the order of approximation when restarting. For the p multigrid methods, V and W cycle have been developed, and restriction and prolongation operators have also been developed. Implementation of h -multigrid has started, with the development of tests of the aggregation methods of PaMPA, and the definition of finite element basis on arbitrary cells.
- **Boundary conditions** Development of the Synthetic Eddy Method boundary condition.
- **Models** Linearized Euler equations, and Sutherland model for non isothermal diffusive flows. Shallow-water model.
- **Parallel computing** Weighted load balancing for hybrid meshes.
- **Validation** Turbulent channel flow.
- **Postprocessing** Development of high order projections over line postprocessing, possibility of stocking averaged data, such as the average flow and the Reynolds stresses.

CARDAMOM Team

6. New Software and Platforms

6.1. AeroSol

Developed in partnership with the Cagire Inria team, the AeroSol library is a high order finite element library written in C++. The code design has been carried for being able to perform efficient computations, with continuous and discontinuous finite element methods on hybrid and possibly curvilinear meshes.

The work of the Cardamom team is focused on continuous finite element methods, while we focus on discontinuous Galerkin methods. However, everything is done for sharing the largest possible part of code. The distribution of the unknowns is made with the software PaMPA, first developed within the Inria teams Bacchus and Castor, and currently maintained in the Tadaam team.

The generic features of the library are

- **High order.** It can be theoretically any order of accuracy, but the finite element basis, and quadrature formula are implemented for having up to a fifth order of accuracy.
- **Hybrid and curvilinear meshes.** AeroSol can deal with up to fifth order conformal meshes composed of lines, triangles, quadrangles, tetrahedra, hexahedra, prism, and pyramids.
- **Continuous and discontinuous discretization.** AeroSol deals with both continuous and discontinuous finite element methods.

We would like to emphasize three assets of this library:

- **Its development environment** For allowing a good collaborative work and a functional library, a strong emphasis has been put on the use of modern collaborative tools for developing our software. This includes the active use of a repository, the use of CMake for the compilation, the constant development of unitary and functional tests for all the parts of the library (using CTest), and the use of the continuous integration tool Jenkins for testing the different configurations of AeroSol and its dependencies. Efficiency is regularly tested with direct interfacing with the PAPI library or with tools like scalasca.
- **Its genericity** A lot of classes are common to all the discretization, for example classes concerning I/O, finite element functions, quadrature, geometry, time integration, linear solver, models and interface with PaMPA. Adding simple features (e.g. models, numerical flux, finite element basis or quadrature formula) can be easily done by writing the class, and declaring its use in only one class of the code.
- **Its efficiency** This modularity is achieved by means of template abstraction for keeping good performances. Dedicated efficient implementation, based on the data locality of the discontinuous Galerkin method has been developed. As far as parallelism is concerned, we use point-to-point communications, the HDF5 library for parallel I/O.

The AeroSol project fits with the first axis of the Bordeaux Sud Ouest development strategy, which is to build a coherent software suite scalable and efficient on new architectures, as the AeroSol library relies on several tools developed in other Inria teams, especially for the management of the parallel aspects.

At the end of 2014, AeroSol had the following features:

- **Development environment** Use of CMake for compilation (gcc, icc and xlc), CTest for automatic tests and memory checking, lcov and gcov for code coverage reports. Development of a CDash server for collecting the unitary tests and the memory checking. Beginning of the development of an interface for functional tests. Optional linking with HDF5, PAPI, with dense small matrices libraries (BLAS, Eigen)

- **In/Out Link** with the XML library for handling with parameter files. Parallel reader for GMSH, with an embedded geometrical pre-partitioner. Writer on the VTK-ASCII legacy format (cell and point centered). Parallel output in vtu and pvtu (Paraview) for cell-centered visualization, and XDMF/HDF5 format for both cell and point centered visualization. Ability of saving the high order solution and restarting from it. Computation of volumic and probe statistics. Ability of saving averaged layer data in quad and hexa meshes. Ability of defining user defined output visualization variables.
- **Quadrature formula** up to 11th or
- April-June 2015: A. Javadi (PhD student) from Chalmers University, Gothenburg, Sweden (3 months).der for Lines, Quadrangles, Hexaedra, Pyramids, Prisms, up to 14th order for tetrahedron, up to 21st order for triangles. Gauss-Lobatto type quadrature formula for lines, triangles, quadrangles and hexaedra.
- **Finite elements** up to fourth degree for Lagrange finite elements and hierarchical orthogonal finite element basis (with Dubiner transform on simplices) on lines, triangles, quadrangles, tetrahedra, prisms, hexaedra and pyramids. Finite element basis that are interpolation basis on Gauss-Legendre points for lines, quadrangles, and hexaedra, and triangle (only 1st and 2nd order)
- **Geometry** Elementary geometrical functions for first order lines, triangles, quadrangles, prisms, tetrahedra, hexaedra and pyramids. Handling of high order meshes.
- **Time iteration** explicit Runge-Kutta up to fourth order, explicit Strong Stability Preserving schemes up to third order. Optimized CFL time schemes: SSP(2,3) and SSP(3,4). CFL time stepping. Implicit integration with BDF schemes from 2nd to 6th order Newton method for stationary problems. Implicit unstationary time iterator non consistent in time for stationary problems. Implementation of in house GMRES and conjugate gradient based on Jacobian free iterations.
- **Linear Solvers** Link with the external linear solver UMFPack, PETSc and MUMPS. Internal solver for diagonal and block-diagonal matrices.
- **Memory handling** discontinuous and continuous, sequential and parallel discretizations based on PaMPA for generic meshes, including hybrid meshes.
- **Models** Perfect gas Euler system, real gas Euler system (template based abstraction for a generic equation of state), scalar advection, Waves equation in first order formulation, generic interface for defining space-time models from space models. Diffusive models: isotropic and anisotropic diffusion, compressible Navier-Stokes. Scalar advection-diffusion model.
- **Numerical schemes** Continuous Galerkin method for the Laplace problem (up to fifth order) with non consistent time iteration or with direct matrix inversion. Explicit and implicit discontinuous Galerkin methods for hyperbolic systems, diffusive and advection-diffusion problems. Beginning of optimization by stocking the geometry for advection problems. SUPG and Residual distribution schemes. Optimization of DG schemes for advection-diffusion problems: stocking of the geometry and use of BLAS for all the linear phases of the scheme.
- **Numerical fluxes** Centered fluxes, exact Godunov' flux for linear hyperbolic systems, and Lax-Friedrich flux. Riemann solvers for Low Mach flows. Numerical flux accurate for steady and unsteady computations.
- **Boundary conditions** Periodic boundary conditions, time-dependent inlet and outlet boundary conditions. Adiabatic wall and isothermal wall. Steger-Warming based boundary condition.
- **Parallel computing** Mesh redistribution, computation of Overlap with PaMPA. Collective asynchronous communications (PaMPA based). Asynchronous point to point communications. Tests on the cluster Avakas from MCIA, and on Mésocentre de Marseille, and PlaFRIM. Tier-1 Turing (Blue-Gene).
- **C++/Fortran interface** Tests for binding fortran with C++.

- **Instrumentation** Aerosol can give some traces on memory consumption/problems with an interfacing with the PAPI library. Tests have also been performed with VTUNE and TAU. Tests with Maqao and Scalasca (VIHPS workshop).
- **Validation** Poiseuille, Taylor-Green vortex. Laplace equation on a ring and Poiseuille flow on a ring. Implementation of volumic forcing based on wall dissipation.

In 2015, N. Pattakos was hired in the team Cardamom, in order to improve the code architecture and for easing the installation of the library. The following features were added or improved:

- **Development environment** The use of CMake was strongly improved, which induced also easier test launching. Documentation, code cleaning and refactorization have also been led. The shared project of Plafrim was updated, and so was the joint Aerosol/Scotch/PaMPA project on the continuous integration platform. Integration of SPack for handling dependencies has begun. Interface with ESSL was fixed.
- **Multigrid** Development of p -multigrid methods. This includes also the possibility of beginning a computation with an order and to decrease or increase the order of approximation when restarting. For the p multigrid methods, V and W cycle have been developed, and restriction and prolongation operators have also been developed. Implementation of h -multigrid has started, with the development of tests of the aggregation methods of PaMPA, and the definition of finite element basis on arbitrary cells.
- **Boundary conditions** Development of the Synthetic Eddy Method boundary condition.
- **Models** Linearized Euler equations, and Sutherland model for non isothermal diffusive flows. Shallow-water model.
- **Parallel computing** Weighted load balancing for hybrid meshes.
- **Validation** Turbulent channel flow.
- **Postprocessing** Development of high order projections over line postprocessing, possibility of stocking averaged data, such as the average flow and the Reynolds stresses.

6.2. Cut-ANOVA

Participants: Pietro Marco Congedo, Kunkun Tang [Corresponding member].

An anchored analysis of variance (ANOVA) method is proposed to decompose the statistical moments. Compared to the standard ANOVA with mutually orthogonal component functions, the anchored ANOVA, with an arbitrary choice of the anchor point, loses the orthogonality if employing the same measure. However, an advantage of the anchored ANOVA consists in the considerably reduced number of deterministic solver's computations, which renders the uncertainty quantification of real engineering problems much easier. Different from existing methods, the covariance decomposition of the output variance is used in this work to take account of the interactions between non-orthogonal components, yielding an exact variance expansion and thus, with a suitable numerical integration method, provides a strategy that converges. This convergence is verified by studying academic tests. In particular, the sensitivity problem of existing methods to the choice of anchor point is analyzed via the Ishigami case, and we point out that covariance decomposition survives from this issue. Also, with a truncated anchored ANOVA expansion, numerical results prove that the proposed approach is less sensitive to the anchor point. The covariance-based sensitivity indices (SI) are also used, compared to the variance-based SI. Furthermore, we emphasize that the covariance decomposition can be generalized in a straightforward way to decompose higher-order moments. For academic problems, results show the method converges to exact solution regarding both the skewness and kurtosis. The proposed method can indeed be applied to a large number of engineering problems.

The Cut-ANOVA code (Fortran 90, MPI + OpenMP) is devoted to the stochastic analysis of numerical simulations. The method implemented is based on the spectral expansion of "anchored ANOVA", allowing the covariance-based sensitivity analysis. Compared to the conventional Sobol method, "Cut-ANOVA" provides three sensitivity indices instead of one, which allows a better analysis of the reliability of the numerical prediction. On the other hand, "Cut-ANOVA" is able to compute the higher order statistical moments such as the Skewness (3-rd order moment) and Kurtosis (4-th order moment). Several dimension reduction techniques have also been implemented to reduce the computational cost. Finally, thanks to the innovative method implemented into the Code Cut-ANOVA, one can obtain a similar accuracy for stochastic quantities by using a considerably less number of deterministic model evaluations, compared with the classical Monte Carlo method.

6.3. Sparse-PDD

Participants: Pietro Marco Congedo, Kunkun Tang [Corresponding member].

The polynomial dimensional decomposition (PDD) is employed in this code for the global sensitivity analysis and uncertainty quantification (UQ) of stochastic systems subject to a moderate to large number of input random variables. Due to the intimate structure between the PDD and the Analysis of Variance (ANOVA) approach, PDD is able to provide a simpler and more direct evaluation of the Sobol' sensitivity indices, when compared to the Polynomial Chaos expansion (PC). Unfortunately, the number of PDD terms grows exponentially with respect to the size of the input random vector, which makes the computational cost of standard methods unaffordable for real engineering applications. In order to address the problem of the curse of dimensionality, this code proposes essentially variance-based adaptive strategies aiming to build a cheap meta-model (i.e. surrogate model) by employing the sparse PDD approach with its coefficients computed by regression. Three levels of adaptivity are carried out in this code: 1) the truncated dimensionality for ANOVA component functions, 2) the active dimension technique especially for second- and higher-order parameter interactions, and 3) the stepwise regression approach designed to retain only the most influential polynomials in the PDD expansion. During this adaptive procedure featuring stepwise regressions, the surrogate model representation keeps containing few terms, so that the cost to resolve repeatedly the linear systems of the least-square regression problem is negligible. The size of the finally obtained sparse PDD representation is much smaller than the one of the full expansion, since only significant terms are eventually retained. Consequently, a much less number of calls to the deterministic model is required to compute the final PDD coefficients.

6.4. RobUQ

Participants: Pietro Marco Congedo [Corresponding member], Maria Giovanna Rodio, Kunkun Tang.

The RobUQ platform has been conceived to solve problems in uncertainty quantification and robust design. It includes the optimization code ALGEN, and the uncertainty quantification code NISP. It includes also some methods for the computation of high-order statistics, efficient strategies for robust optimization, the Simplex2 method. Some methods are developed in partnership with the Stanford University (in the framework of the associated team AQUARIUS). Other methods are developed in the context of ANR UF0.

6.5. ORComp

Participants: Pietro Marco Congedo [Corresponding member], Nassim Razaaly, Maria-Giovanna Rodio.

The ORComp platform is a simulation tool permitting to design an ORC cycle. Starting from the solar radiation, this platform computes the cycle providing the best performance with optimal choices of the fluid and the operating conditions. It includes RobUQ, a simulation block of the ORC cycles, the fluidbox code for the simulation of the turbine and of the heat exchanger, the software FluidProp (developed at the University of Delft) for computing the fluid thermodynamic properties.

6.6. sDEM

Participants: Pietro Marco Congedo [Corresponding member], Maria-Giovanna Rodio.

The sDEM platform is a simulation tool permitting to simulate multiphase flows with transition modelling. In particular, the code relies on the formulation of a DEM method, the use of a complex thermodynamics, the possibility to model cavitating phenomena. Moreover, the method has been generalized in order to take into account directly uncertainty, thus proposing the so-called Stochastic DEM (sDEM) method. This is one of the first stochastic semi-intrusive scheme, permitting to consider uncertainties in multiphase flows including heat and mass transfer terms. This software is developed together with the University of Zurich.

6.7. SLOWS

Participants: Luca Arpaia, Andrea Filippini, Maria Kazolea, Mario Ricchiuto [Corresponding member].

SLOWS is a C-platform allowing the simulation of free surface shallow water flows with friction. It can be used to simulate near shore hydrodynamics, wave transformations processes, etc. Both hydrostatic (shallow water) and non-hydrostatic (Boussinesq-type) versions exist. The latter are currently based on the dispersive model of Madsen and Sorensen (1992). A fully nonlinear (Green-Naghdi) version is under development based on the one dimensional prototype discussed in [99]. Three different approaches are available, based on conditionally depth-positivity preserving implicit schemes, or on conditionally depth-positivity preserving genuinely explicit discretizations, or on an unconditionally depth-positivity preserving space-time approach. Newton and frozen Newton loops are used to solve the implicit nonlinear equations. The linear algebraic systems arising in the discretization are solved with the MUMPS library. This year implicit and explicit (extrapolated) multistep higher order time integration methods have been implemented, and a mesh adaptation technique based on mesh deformation (r -adaptation) has been also included. A node-centred high order MUSCL finite volume discretisation has also been added to benchmark the mesh adaptation strategies, and compare with the residual based method constituting the kernel of the code. To date, SLOWS is the only existing near shore code allowing to choose between a classical finite volume approximation, and the more recent non-linear residual distribution methods developed in CARDAMOM. It allows an accurate simulation of free surface flows on arbitrary topographies with both static and time dependent unstructured mesh adaptation, accounting for both hydrostatic and non-hydrostatic effects.

6.8. TUCWave

Participant: Maria Kazolea [Corresponding member].

TUCWave, developed within the PhD of M. Kazolea, is a high-order well-balanced unstructured finite volume (FV) solver for weakly nonlinear and weakly dispersive water waves over varying bathymetries, as described by the 2D depth-integrated extended Boussinesq equations of Nwogu (1993). The FV scheme numerically solves the conservative form of the equations following the median dual node-centered approach, for both the advective and dispersive part of the equations. The code uses an efficient edge based structure. For the advective fluxes Roe's approximate Riemann solver is used along with a well-balanced treatment of the topography source. Higher order accuracy is achieved through a MUSCL-type reconstruction technique, and via a strong stability preserving explicit Runge-Kutta time stepping. The numerical techniques implemented in TUCWave are being imported in SLOWS.

6.9. Realfluids

Participants: Héloïse Beaugendre [Corresponding member], Pietro Marco Congedo, Andrea Cortesi, Léo Nouveau, Quentin Viville.

RealFluids (developed in the BACCHUS team) solves compressible viscous turbulent flow equations, with real-gas effects and arbitrarily complex equations of state, with the most recent residual distribution schemes. It is currently used for simulating turbines in ORC optimization, and for the immersed boundary simulations for de-anti icing applications. It is being coupled to the Mutation library to be able to perform some CFD simulations of low-altitude re-entry flows.

6.10. FMG

Participants: Luca Arpaia, Cécile Dobrzynski [Corresponding member], Andrea Cortesi, Léo Nouveau, Mario Ricchiuto.

FMG is a library deforming an input/reference simplicial mesh w.r.t. a given smoothness error monitor (function gradient or Hessian), metric field, or given mesh size distribution. Displacements are computed by solving an elliptic Laplacian type equation with a continuous finite element method. The library returns an adapted mesh with a corresponding projected solution, obtained by either a second order projection, or by an ALE finite element remap. The addition of a new mass conservative approach developed ad-hoc for shallow water flows is under way.

6.11. MMG platform

Participants: Cécile Dobrzynski [Corresponding member], Algiane Froehly.

MMG is an open source software for surface and volume remeshing. It provides three applications : 1) mmg2d: generation of a triangular mesh , adaptation and optimization of a triangular mesh 2) mmgs: adaptation and optimization of a surface triangulation representing a piecewise linear approximation of an underlying surface geometry 3) mmg3d: adaptation and optimization of a tetrahedral mesh and implicit domain meshing

URL : <http://www.mmgtools.org>

6.12. Nomesh

Participants: Cécile Dobrzynski [Corresponding member], Ghina El Jannoun.

Nomesh is a software allowing the generation of third order curved simplicial meshes. Starting from a "classical" mesh with straight elements composed by triangles and/or tetrahedra, we are able to curve the boundary mesh. Starting from a mesh with some curved elements, we can verify if the mesh is valid, that means there is no crossing elements and only positive Jacobian. If the curved mesh is non valid, we modify it using linear elasticity equations until having a valid curved mesh.

CARMEN Team

6. New Software and Platforms

6.1. CEPS: a Cardiac ElectroPhysiology Simulator

The Carmen team develops a software code to perform high performance numerical simulations in cardiac electrophysiology using unstructured three-dimensional grids. The software, called CEPS (*Cardiac Electrophysiology Simulation*) is developed as a common tool for researchers in the Carmen team and for our partners and colleagues in scientific computing and biomedical engineering. The goal of CEPS is to easily allow the development of new numerical methods and new physical models.

As compared to other existing softwares, CEPS aims at providing a more general framework of integration for new methods or models and a better efficiency in parallel. CEPS is designed to run on massively parallel architectures, and to make use of state-of-the-art and well known computing libraries to achieve realistic and complex heart simulations. CEPS also includes software engineering and validation tools. We use the platform GForge ([ceps](#)) based on Git. This allows to keep a history of developments for developers and users. Some of our collaborators actively participate to the testing and discussion for the development of CEPS, namely:

- C. Pierre, LMA University of Pau et des Pays de l'Adour;
- R. Turpault, IMB University of Bordeaux;

Several people work and make an useful code for researchers and users.

- Development of an external procedure to compile dependencies for CEPS. This allows a very simple way to install CEPS for partners or students.
- Improve continuous integration test cases in order to have a best coverage of the code as possible.
- Overwrite C++ class for ionic models and adding new models in collaboration with A. Gérard.
- Integration of the partitionneur PTScotch in order to realise a partitioning on the nodes.
- Tutorials for beginners on the code (linear algebra, installation, compilation...).
- New implementation of the bilayer model developed by L. Simon during his PhD thesis. The most important part will be provided by M. Fuentes works. Difficulties are currently the specification of two layers in CEPS and how to connect them.
- The strategy remains at this time is to have two meshes in entry (on global mesh for the auricles and another *under-mesh* corresponding to the two layers domain). This means, that we have to create a connectivity table between the global mesh and the *under-mesh*.
- Development of an interface for users in order to specify data in a text file for the code such as physical values for ionic models or numerical values for numerical methods (especially in time) used.

6.2. IDAM

The goal of the IDAM project is to define a conceptual module in MUSIC in order to create realistic meshes for the CEPS code. Information comes from IRM done by doctors. Furthermore, objectives are the continuation of used methods in the team and the visualisation of numerical results obtained by CEPS (<https://bil.inria.fr/software/view/2630/tab>).

This project started on 1st December 2015 for two years. M. Juhor is in charge of this project in collaboration with the MedInria team and the IHU LIRYC.

CQFD Project-Team

6. New Software and Platforms

6.1. Package PCAmixdata

FUNCTIONAL DESCRIPTION

Mixed data type arise when observations are described by a mixture of numerical and categorical variables. The R package PCAmixdata extends standard multivariate analysis methods to incorporate this type of data. The key techniques included in the package are PCAmix (PCA of a mixture of numerical and categorical variables), PCArot (rotation in PCAmix) and MFAmix (multiple factor analysis with mixed data within a dataset). The MFAmix procedure handles a mixture of numerical and categorical variables within a group - something which was not possible in the standard MFA procedure. We also included techniques to project new observations onto the principal components of the three methods in the new version of the package.

- Participants: Marie Chavent, Amaury Labenne, Jérôme Saracco
- Contact: Marie Chavent
- URL: <https://cran.r-project.org/web/packages/PCAmixdata/index.html>

6.2. Package divclust

FUNCTIONAL DESCRIPTION DIVCLUS-T is a divisive hierarchical clustering algorithm based on a monothetic bipartitional approach allowing the dendrogram of the hierarchy to be read as a decision tree. It is designed for numerical, categorical (ordered or not) or mixed data. Like the Ward agglomerative hierarchical clustering algorithm and the k-means partitioning algorithm, it is based on the minimization of the inertia criterion. However, it provides a simple and natural monothetic interpretation of the clusters. Indeed, each cluster is described by set of binary questions. The inertia criterion is calculated on all the principal components of PCAmix (and then on standardized data in the numerical case).

- Participants: Marie Chavent, Marc Fuentes
- Contact: Marie Chavent
- URL: <https://github.com/chavent/divclust>

6.3. Package ClustGeo

FUNCTIONAL DESCRIPTION This R package is dedicated to the clustering of objects with geographical positions. The clustering method implemented in this package allows the geographical constraints of proximity to be taken into account within the ascendant hierarchical clustering.

- Marie Chavent, Amaury Labenne, Vanessa Kuentz, Jérôme Saracco
- Contact: Amaury Labenne
- URL: <https://cran.r-project.org/web/packages/ClustGeo/index.html>

6.4. Package QuantifQuantile

FUNCTIONAL DESCRIPTION This R package is dedicated to the estimation of conditional quantiles using optimal quantization. It allows the construction of an optimal grid of N quantizers, the estimation of conditional quantiles and the data driven selection of the size N of the grid. Graphical illustrations are available for the selection of N and of resulting estimated curves or surfaces when the dimension of the covariate is one or two.

- Isabelle Charlier, Jérôme Saracco
- Contact: Isabelle Charlier
- URL: <https://cran.r-project.org/web/packages/QuantifQuantile/index.html>

6.5. Biips: Software for Bayesian Inference with Interacting Particle Systems

FUNCTIONAL DESCRIPTION

Biips is a software platform for automatic Bayesian inference with interacting particle systems. Biips allows users to define their statistical model in the probabilistic programming BUGS language, as well as to add custom functions or samplers within this language. Then it runs sequential Monte Carlo based algorithms (particle filters, particle independent Metropolis-Hastings, particle marginal Metropolis-Hastings) in a black-box manner so that to approximate the posterior distribution of interest as well as the marginal likelihood. The software is developed in C++ with interfaces with the softwares R, Matlab and Octave.

- Participants: François Caron, Adrien Todeschini and Pierrick Legrand
- Contact: Adrien Todeschini
- URL: <http://biips.gforge.inria.fr>

6.6. VCN: Software for analysis of VCN

FUNCTIONAL DESCRIPTION

VCN is a software for the analysis of the vigilance of the patient based on the analysis of the EEG signals. The code is written in Matlab and provides an interface easy to use for someone without informatics skills.

- Participants: Pierrick Legrand, Julien Clauzel, Laurent Vezard, Charlotte Rodriguez, Borjan Geshkovski.
- Contact: Pierrick Legrand

6.7. EMGView: Software for visualisation and time-frequency analysis of bio signals

FUNCTIONAL DESCRIPTION

EMGView is a software for the visualisation and the analysis of bio-signals. The code is written in Matlab and provides an interface easy to use for someone without informatics skills.

- Participants: Luis Herrera, Eric Grivel, Pierrick Legrand, Gregory Barriere
- Contact: Pierrick Legrand

FLOWERS Project-Team

6. New Software and Platforms

6.1. Poppy project

6.1.1. Introduction

- Participants: Matthieu Lapeyre, Pierre Rouanet, Nicolas Rabault, Theo Segonds, Jonathan Grizou and Pierre-Yves Oudeyer
- Contact: Pierre-Yves Oudeyer
- URL: <https://www.poppy-project.org/>

The Poppy Project develops open-sources 3D printed robots platforms based on robust, flexible, easy-to-use and reproduce hardware and software. In particular, the use of 3D printing and rapid prototyping technologies is a central aspect of this project, and makes it easy and fast not only to reproduce the platform, but also to explore morphological variants. Poppy targets three domains of use: science, education and art.

6.1.2. Poppy Ergo Jr (hardware)

Poppy Ergo Jr is a new open hardware robot developed by the Poppy Project to explore the use of robots in classroom for learning robotic and computer science. It is a 6-Dofs arm designed to be both expressiv and low-cost. This is achieved by the use of FDM 3D printing, low cost Robotis XL-320 actuators. In addition we have added a Pi camera so the robot can detect object, faces or QR codes.

The Ergo Jr is controlled by the pypot software running on a Raspberry pi 2. The communication between the Raspberry pi and the actuators is made possible by the Pixl board we have designed.

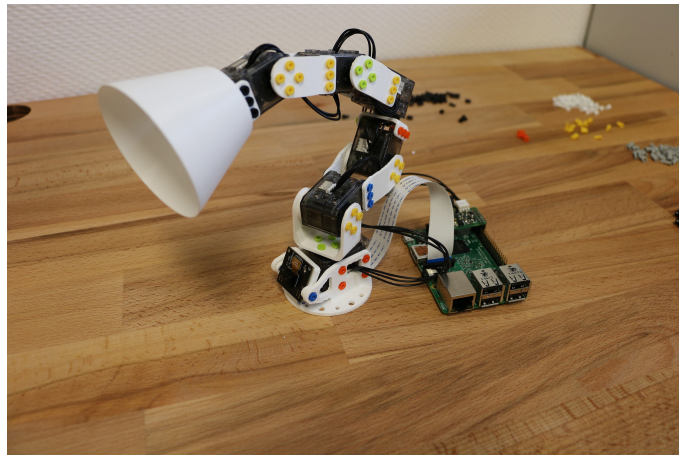


Figure 1. The new open hardware Poppy Ergo Jr, 6-DoFs arm robot for education

The sources are availables on the following web platforms:

- Github repository: <https://github.com/poppy-project/poppy-ergo-jr>
- CAD files: <https://cad.onshape.com/documents/10951c2120eb4209abfff972/w/2b2ed99178db4a72aa4ebcc9>

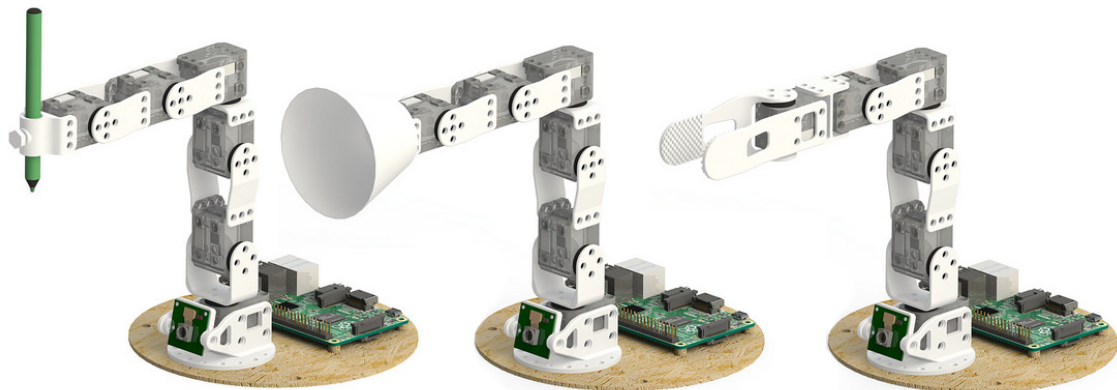


Figure 2. The Ergo Jr has 3 tools, a pen holder, a lampshade and a gripper

The Poppy Ergo Jr has several 3D printed tools to extend its capabilities. There are currently the lampshade, the gripper and a pen holder.

6.1.3. Pixl (Electronics)

Pixl is a tiny board used to create low cost robots based on raspberry pi and XL320 motors. This board have been created by the Poppy project team. Pixl have 2 main features, the power part, and the communication part :

- The power part allow the user to plug an 7V AC/DC converter directly into the Pixl. This power will be distributed to all XL320 motors and will be converted to 5V for the raspberry pi.
- The communication part convert full duplex to half duplex and vice versa. The half duplex part switch between RX and TX automatically. Another connector allow the user to connect his XL320 network.

6.1.4. Poppy Com

FUNCTIONAL DESCRIPTION

Poppy-com is a low level library who manage the new poppy system communication protocol. This work in progress library, written in C/C++, is completely open-source and available on github. Actually poppy-com work with Atmel ATMEGA series microcontroler and a test version can be run on X86 architectures. His main objectiv is to give access to the user at the lowest level of code. Each poppy system module run this code to be detected into the robt network and to communicate with others modules. Users can write their owns code and write it in any module. With this new level the user can create basic behavior directly into a poppy system module.

- Bootloader : The bootloader have to manage robot network discovery, auto-addressing, module specific firmware update validation, and user code update.
- User : The user side have a small part of code dedicated to the module management like motor management routines, sensors synchronizations, numerical treatment of signals... The empty memory of each modules can be used by the user and he can add his own code to manage his robot.

This library will be compatible with the Arduino univers, that allow non expert people to use it anyway. To simplify the robot functionality developpement and function execution localization we want to try to create a way to write a code for all a robot modules on only one code. Each function of this code could be redirected in a specific module, the execution can be completely distributed.

- Participants: Pierre Rouanet, Matthieu Lapeyre, Nicolas Rabault and Pierre-Yves Oudeyer
- Contact: Pierre-Yves Oudeyer
- URL: <https://www.poppy-project.org/>
- Contact: Nicolas Rabault
- URL: <https://github.com/poppy-project/poppy-com>

6.1.5. Poppy System

FUNCTIONAL DESCRIPTION

In the Poppy project we are working on the Poppy System which is a new modular and open-source robotic architecture. It is designed to help people create and build custom robots. It permits, in a similar approach as Lego, building robots or smart objects using standardized elements.

Poppy System is an unified system where each essential robotic components (actuators, sensors, ...) is an independant module, connected with other through standardized interfaces:

- Unified mechanical interfaces which simplifies the assembly process and the design of 3D printable parts.
- Unified communication between elements using the same connector and bus for each module.
- Unified software makes it easy to program each module independantly.

The current Poppy robots (Humanoid, Torso, Ergo) will be updated using this novel architecture.

Our ambition is to create an ecosystem around this system so communities can develop custom modules, following the Poppy System standards, which can be compatible with all other Poppy robots.

6.1.6. Pypot

SCIENTIFIC DESCRIPTION

Pypot is a framework developed to make it easy and fast to control custom robots based on dynamixel motors. This framework provides different levels of abstraction corresponding to different types of use. More precisely, you can use pypot to:

- directly control robotis motors through a USB2serial device,
- define the structure of your particular robot and control it through high-level commands,
- define primitives and easily combine them to create complex behavior.

Pypot is part of the Poppy project. It is the core library used by the different Poppy robots. This abstraction layer allows to seamlessly switch from one of the Poppy robot to another. It also provides a common set of tools, such as forward and inverse kinematics, simple computer vision, recording and replaying moves, or easy access to the autonomous exploration library Explauto.

To extend pypot application domains and connection to outside world, it also provides an HTTP REST API. On top of providing an easy way to connect to smart sensors or connected devices, it is notably used to connect to Snap! a variant of the well-known Scratch visual programming language.

FUNCTIONAL DESCRIPTION

Pypot has been entirely written in Python to allow for fast development, easy deployment and quick scripting by non-necessary expert developers. It can also benefits from the scientific and machine learning libraries existing in Python. The serial communication is handled through the standard library and thus allows for rather high performance (10ms sensorimotor loop). It is crossed-platform and has been tested on Linux, Windows and Mac OS.

Pypot is also compatible with the V-REP simulator. This allows the transparent switch from a real robot to its simulated equivalent without having to modify the code.

Finally, it has been developed to permit an easy and fast extension to other types of motors and sensors.

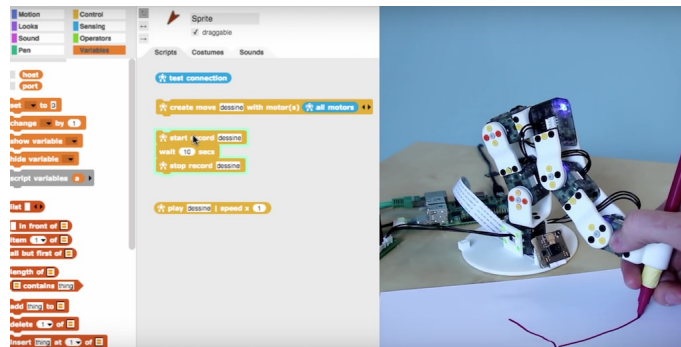


Figure 3. Example of using pypot to program a robot to reproduce a drawn shape

It works with Python 2.7 or Python 3.3 or later. It has also been adapted to the Raspberry-Pi board.

CONNECTION WITH THE VISUAL PROGRAMMING LANGUAGE SNAP!

Pypot has been connected to Snap!, a variant of the famous Scratch visual language, developed for teaching computer science to children. It is based on blocks that can be dragged-and-drop, and assembled to write scripts.

Thanks to the Snap! HTTP block, a connection can be made to pypot allowing users to directly control robots through their visual interfaces. A set of dedicated Snap! blocks have been designed, such as *set motor position* or *get motor temperature*.



Figure 4. Using Snap! to program a robot by demonstration and create complex choreographies

Snap! is also used as a tool for programming the robot by demonstration. Thanks to *record* and *play* blocks, users can easily trigger kinesthetic recording of the robots (or only subpart such as one arm). They can then be played or "mixed" - either played in sequence or in parallel - with other recordings to compose

complex choreographies. The moves are encoded as a model of mixture of gaussians (GMM) which allows the definition of clean mathematical operators for combining them.

This tool has been developed and used in collaboration with artists who investigate the concept of robotic moves.

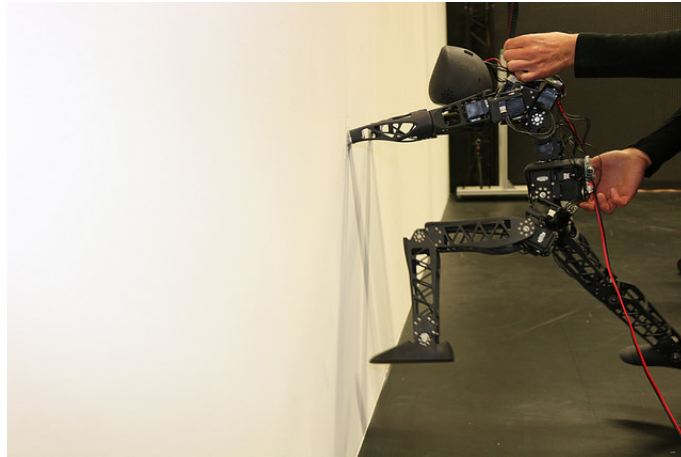


Figure 5. Artistic project exploring the concept of robotic move.

DISSEMINATION AND CONTRIBUTION

The pypot source files are released under the GPLv3 license and can be accessed on Github. They have been downloaded about 50k times (source Python Package Index) and forked about 50 times (source Github).

18 contributors have participated to its development.

Pypot are also largely based on jupyter notebooks to provide examples, tutorials or scientific experiments.

- Participants: Pierre Rouanet, Theo Segonds, Matthieu Lapeyre
- Contact: Pierre Rouanet
- URL: <https://github.com/poppy-project/pypot>

6.1.7. Inverse kinematics library

FUNCTIONAL DESCRIPTION

IKPy is a Python Inverse Kinematics library, designed to be simple to use and extend. It provides Forward and Inverse kinematics functionalities, bundled with helper tools such as 3D plotting of the kinematics chains. Being written entirely in Python, IKPy is lightweight and is based on numpy and scipy for fast optimization. IKPy is compatible with many robots, by automatically parsing URDF files. It also supports other (such as DH-parameters) and custom representations. Moreover, it provides a framework to easily implement new Inverse Kinematics strategies. Originally developed for the Poppy project, it can also be used as a standalone library. IKPy is open-source, and can be found at: <https://github.com/Phylliade/ikpy>

- Participants: Pierre Manceron, Pierre Rouanet, Pierre-Yves Oudeyer
- Contact: Pierre Rouanet
- URL: <https://github.com/Phylliade/ikpy>

6.2. Tools for robot learning, control and perception

6.2.1. CARROMAN

FUNCTIONAL DESCRIPTION

This software implements a control architecture for the Meka humanoid robot. It integrates the Stanford Whole Body Control in the M3 architecture provided with the Meka robot, and provides clear and easy to use interfaces through the URBI scripting language. This software provides a modular library of control modes and basic skills for manipulating objects, detecting objects and humans which other research projects can reuse, extend and enhance. An example would be to locate a cylindrical object on a table using stereo vision, and grasping it using position and force control.

- Contact: David Filliat

6.2.2. DMP-BBO

Black-Box Optimization for Dynamic Movement Primitives

FUNCTIONAL DESCRIPTION

The DMP-BBO Matlab library is a direct consequence of the insight that black-box optimization outperforms reinforcement learning when using policies represented as Dynamic Movement Primitives. It implements several variants of the PIBB algorithm for direct policy search. The `dmp_bbo` C++ library (<https://github.com/stulp/dmpbbo>) has been extended to include the “unified model for regression”, see Section 7.2.3 . The implementation of several of the function approximators have been made real-time compatible.

- Participant: Freek Stulp
- Contact: Freek Stulp
- URL: <https://github.com/stulp/dmpbbo>

6.2.3. DyNAMoS

FUNCTIONAL DESCRIPTION

This simulation software comes in the form of a PYTHON module and allows a user to define and simulate complex neural architectures while making use of the parallelism inherent to modern multi-core processors. A special focus lies on on-line learning, processing inputs one by one, in contrast to batch processing of whole databases at a time.

- Participants: Alexander Gepperth and Mathieu Lefort
- Contact: Mathieu Lefort

6.2.4. Multimodal Concept Learning with Non-negative Matrix Factorization

FUNCTIONAL DESCRIPTION

The python code provides a minimum set of tools and associated libraries to reproduce the experiments in [98] , together with the choreography datasets. The code is primarily intended for reproduction of the multimodal learning experiment mentioned above. It has already been reused in several experimentations by other member of the team and is expected to play an important role in further collaborations. It is also expected that the public availability of the code encourages further experimentation by other scientists with data coming from other domains, thus increasing both the impact of the aforementioned publication and the knowledge on the algorithm behaviors.

- Participant: Olivier Mangin
- Contact: Olivier Mangin
- URL: <https://github.com/omangin/multimodal>

6.2.5. Explorers

FUNCTIONAL DESCRIPTION

The Explorers framework is aimed at creating, testing and comparing autonomous exploration strategies for sensorimotor spaces in robots. The framework is largely strategy-agnostic, and is aimed at expressing motor babbling, goal babbling and intrinsically motivated exploration algorithms, among other. It is also able to express strategies that feature transfer learning, such as the reuse algorithm.

- Participants: Pierre-Yves Oudeyer and Fabien Benureau
- Contact: Pierre-Yves Oudeyer
- URL: <https://github.com/humm/explorers>

6.2.6. *Of 3-D point cloud*

FUNCTIONAL DESCRIPTION

This software scans the 3-D point cloud of a scene to find objects and match them against a database of known objects. The process consists in 3 stages. The segmentation step finds the objects in the point cloud, the feature extraction computes discriminating properties to be used in the classification stage for object recognition.

- Participants: David Filliat, Alexander Gepperth and Louis-Charles Caron
- Contact: Alexander Gepperth

6.2.7. *OptiTrack*

FUNCTIONAL DESCRIPTION

This python library allows you to connect to an OptiTrack from NaturalPoint. This camera permits the tracking of 3D markers efficiently and robustly. With this library, you can connect to the Motive software used by the OptiTrack and retrieve the 3D position and orientation of all your tracked markers directly from python.

- Participant: Pierre Rouanet
- Contact: Pierre Rouanet

6.2.8. *PEDETECT*

FUNCTIONAL DESCRIPTION

PEDETECT implements real-time person detection in indoor or outdoor environments. It can grab image data directly from one or several USB cameras, as well as from pre-recorded video streams. It detects multiple persons in 800x600 color images at frame rates of >15Hz, depending on available GPU power. In addition, it also classifies the pose of detected persons in one of the four categories "seen from the front", "seen from the back", "facing left" and "facing right". The software makes use of advanced feature computation and nonlinear SVM techniques which are accelerated using the CUDA interface to GPU programming to achieve high frame rates. It was developed in the context of an ongoing collaboration with Honda Research Institute USA, Inc.

- Participant: Alexander Gepperth
- Contact: Alexander Gepperth

6.2.9. *pyStreamPlayer*

FUNCTIONAL DESCRIPTION

This Python software is intended to facilitate the application of machine learning algorithms by avoiding to work directly with an embodied agent but instead with data recorded in such an agent. Assuming that non-synchronous data from multiple sensors (e.g., camera, Kinect, laser etc.) have been recorded according to a flexible format defined by the pyStreamPlayer architecture, pyStreamPlayer can replay these data while retaining the exact temporal relations between different sensor measurements. As long as the current task does not involve the generation of actions, this software allows to process sensor data as if it was coming from an agent which is usually considerably easier. At the same time, pyStreamPlayer allows to replay arbitrary supplementary information such as, e.g., object information, as if it was coming from a sensor. In this way, supervision information can be stored and accessed together with sensory measurements using a unified interface. pyStreamPlayer has been used to facilitate real-world object recognition tasks, and several of the

major databases in this field (CalTech Pedestrian database, HRI RoadTraffic traffic objects database, CVC person database, KITTI traffic objects database) have been converted to the pyStreamPlaeer format and now serve as a source of training and test data for learning algorithms.

- Participant: Alexander Gepperth
- Contact: Alexander Gepperth

6.2.10. Aversive++

FUNCTIONAL DESCRIPTION

Aversive++ is a C++ library that eases microcontroller programming. Its aim is to provide an interface simple enough to be able to create complex applications, and optimized enough to enable small microcontrollers to execute these applications. The other aspect of this library is to be multiplatform. Indeed, it is designed to provide the same API for a simulator (named SASIAE) and for AVR-based and ARM-based microcontrollers.

- Participants: Loïc Dauphin
- Contact: Loïc Dauphin
- Website: <http://aversiveplusplus.com/>

6.3. Explauto: Autonomous Exploration and Learning Benchmarking

An autonomous exploration library

SCIENTIFIC DESCRIPTION

An important challenge in Developmental Robotics is how robots can efficiently learn sensorimotor mappings by experience, i.e. the mappings between the motor actions they make and the sensory effects they produce. This can be a robot learning how arm movements make physical objects move, or how movements of a virtual vocal tract modulates vocalization sounds. The way the robot will collect its own sensorimotor experience have a strong impact on learning efficiency because for most robotic systems the involved spaces are high dimensional, the mapping between them is non-linear and redundant, and there is limited time allowed for learning. If robots explore the world in an unorganized manner, e.g. randomly, learning algorithms will be often ineffective because very sparse data points will be collected. Data are precious due to the high dimensionality and the limited time, whereas data are not equally useful due to non-linearity and redundancy. This is why learning has to be guided using efficient exploration strategies, allowing the robot to actively drive its own interaction with the environment in order to gather maximally informative data to feed the sensorimotor model.

In the recent year, work in developmental learning has explored various families of algorithmic principles which allow the efficient guiding of learning and exploration.

Explauto is a framework developed to study, model and simulate curiosity-driven learning and exploration in virtual and robotic agents. Explauto's scientific roots trace back from Intelligent Adaptive Curiosity algorithmic architecture [120], which has been extended to a more general family of autonomous exploration architecture by [73] and recently expressed as a compact and unified formalism [114]. The library is detailed in [115].

In Explauto, the strategies to explore sensorimotor models are called interest models. They implements the active exploration process, where sensorimotor experiments are chosen to improve the forward or inverse prediction of the sensorimotor model. The simplest strategy is to randomly draw goals in the motor or sensory space. More efficient strategies are based on the active choice of learning experiments that maximize learning progress, for e.g. improvement of predictions or of competences to reach goals [120]. This automatically drives the system to explore and learn first easy skills, and then explore skills of progressively increasing complexity. Both random and learning progress models can act either on the motor or on the sensory space, resulting in motor babbling or goal babbling strategies.

- Motor babbling consists in sampling commands in the motor space according to a given strategy (random or learning progress), predicting the expected sensory consequence, executing the command through the environment and observing the actual sensory effect. Both sensorimotor and interest models are finally updated according to this experience.
- Goal babbling consists in sampling goals in the sensory effect space and to use the current state of the sensorimotor model to infer a motor action supposed to reach the goals (inverse prediction). The robot/agent then executes the command through the environment and observes the actual sensory effect. Both sensorimotor and interest models are finally updated according to this experience.

It has been shown that this second strategy allows a progressive covering of the reachable sensory space much more uniformly than in a motor babbling strategy, where the agent samples directly in the motor space [73].

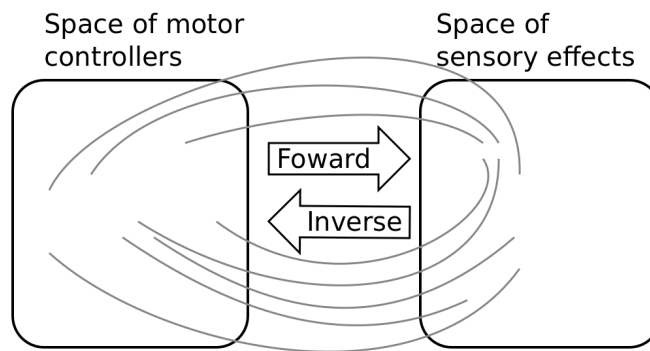


Figure 6. Complex sensorimotor mappings involve a high dimensional sensorimotor spaces. For the sake of visualization, the motor M and sensory S spaces are only 2D each in this example. The relationship between M and S is non-linear, dividing the sensorimotor space into regions of unequal stability: small regions of S can be reached very precisely by large regions of M , or large regions in S can be very sensitive to variations in M : s as well as a non-linear and redundant relationship. This non-linearity can imply redundancy, where the same sensory effect can be attained using distinct regions in M .

FUNCTIONAL DESCRIPTION

This library provides high-level API for an easy definition of:

- Virtual and robotics setups (Environment level),
- Sensorimotor learning iterative models (Sensorimotor level),
- Active choice of sensorimotor experiments (Interest level).

The library comes with several built-in environments. Two of them corresponds to simulated environments: a multi-DoF arm acting on a 2D plan, and an under-actuated torque-controlled pendulum. The third one allows to control real robots based on Dynamixel actuators using the Pypot library.

Learning sensorimotor mappings involves machine learning algorithms, which are typically regression algorithms to learn forward models, from motor controllers to sensory effects, and optimization algorithms to learn inverse models, from sensory effects, or goals, to the motor programs allowing to reach them. We call these sensorimotor learning algorithms sensorimotor models. The library comes with several built-in sensorimotor models: simple nearest-neighbor look-up, non-parametric models combining classical regressions and optimization algorithms, online mixtures of Gaussians, and discrete Lidstone distributions. Explauto sensorimotor models are online learning algorithms, i.e. they are trained iteratively during the interaction of the robot in the environment in which it evolves.

Explauto provides also a unified interface to define exploration strategies using the InterestModel class. The library comes with two built-in interest models: random sampling as well as sampling maximizing the learning progress in forward or inverse predictions.

This library has been used in many experiments including:

- the control of a 2D simulated arm,
- the exploration of the inverse kinematics of a poppy humanoid (both on the real robot and on the simulated version),
- acoustic model of a vocal tract.

Explauto is cross-platform and has been tested on Linux, Windows and Mac OS. It has been released under the GPLv3 license.

- Participants: Pierre Rouanet, Clément Moulin-Frier, Sébastien Forestier, Pierre-Yves Oudeyer
- Contact: Pierre Rouanet
- URL: <https://github.com/flowersteam/explauto>

6.4. KidLearn: active teaching in Intelligent Tutoring Systems

KEYWORD: Automatic Learning

FUNCTIONAL DESCRIPTION

KidLearn is a software which adaptively personalizes sequences of learning activities to the particularities of each individual student. It aims at proposing to the student the right activity at the right time, maximizing concurrently his learning progress and its motivation.

- Participants: Benjamin Clement, Pierre Yves Oudeyer, Didier Roy and Manuel Lopes
- Contact: Manuel Lopes
- URL: <https://flowers.inria.fr/research/kidlearn/>

6.5. Self-calibration BCI

KEYWORDS: Neurosciences - Health - Brain-Computer Interface

FUNCTIONAL DESCRIPTION

Self-calibration BCI is a Matlab library which allows a robot to be instructed a new task by a human using communicative signals initially totally unknown to the robot. It is extended and improved in the context of EEG-based brain-machine interfaces (BMIs).

It results in a BCI based control of sequential tasks with feedback signals that do not require any calibration process. As a by-product, the method provides an unsupervised way to train a decoder with the same performance than state-of-the-art supervised classifiers, while keeping the system operational and solving, with a lower performance during the first steps, the unknown task. The algorithm has been tested with online experiments, showing that the users were able to guide from scratch an agent to a desired position.

- Participants: Manuel Lopes, Jonathan Grizou and Pierre-Yves Oudeyer
- Contact: Jonathan Grizou
- URL: https://github.com/flowersteam/self_calibration_BCI_plosOne_2015/

6.6. Platforms

6.6.1. Platform: Collaborative assemblies with Baxter

- Participant: Yoan Mollard, Baptiste Busch, Thibaut Munzer
- Contact: Yoan Mollard

FUNCTIONAL DESCRIPTION This platform is a set of software components and hardware robotic components designed as an experimental setup for performing scientific experiments with the Baxter robot illustrating human-robot collaboration. It comes with a set of capabilities (pick objects, handover, hold objects in place, ...) and physical objects (screwdriver, landmarks, camera mounts, ...) created on purpose or hacked to serve these capabilities. The initial capabilities focus on industrial activities and allow the robot to provide assistance to workers in manufacturing factories for their daily tasks (pieces fetching, screwing, assembly, ...). We simulated an industrial environment with a trolley acting as a feeder where all spare parts are initially located, a workspace for the worker and 7 spare parts composing a wooden toolbox that users and robot will handle for their collaborative tasks. This industrial environment is the one of the 3rd hand project, but the platform and its capabilities will progressively be improved and enriched to be used for other projects. The picture 7 illustrates the experimental setup.

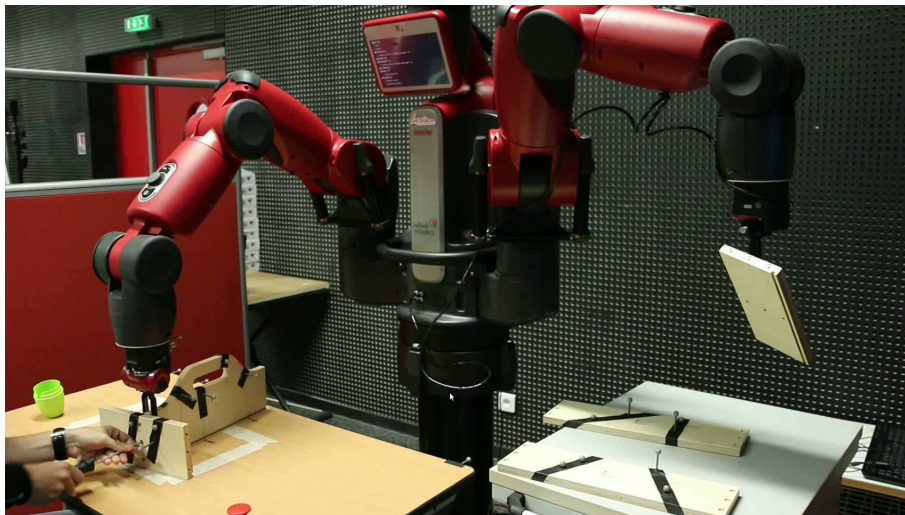


Figure 7. Experimental setup for human-robot collaborative assemblies of a wooden toolbox

GEOSTAT Project-Team

6. New Software and Platforms

6.1. Fluex

KEYWORDS: Signal - Signal processing

SCIENTIFIC DESCRIPTION

Fluex is a nonlinear signal processing software for 1D, 2D 3D and 3D+t general signals.

FUNCTIONAL DESCRIPTION

- Fluex is a library in nonlinear signal processing, written in C++, developed under Gforge, able to analyze turbulent and natural complex signals.
- Fluex is able to determine low level features in natural signals that cannot be determined using standard linear techniques.
- Participants: Hussein Yahia, Denis Arrivault, Rémi Paties.
- Contact: Hussein Yahia.
- URL: <https://geostat.bordeaux.inria.fr/index.php/downloads.html>.
- Fluex is deposited APP, Inter Deposit Digital Number: IDDN.FR.001.51.0028.000.S.P.2015.000.21000

6.2. FluidExponents

- Participants: Hussein Yahia and Antonio Turiel
- Contact: Hussein Yahia
- URL: <https://geostat.bordeaux.inria.fr/index.php/downloads.html>.

HIEPACS Project-Team

6. New Software and Platforms

6.1. Introduction

We describe in this section the software that we are developing. The first list will be the main milestones of our project. The other software developments will be conducted in collaboration with academic partners or in collaboration with some industrial partners in the context of their private R&D or production activities. For all these software developments, we will use first the various (very) large parallel platforms available through GENCI in France (CCRT, CINES and IDRIS Computational Centers), and next the high-end parallel platforms that will be available via European and US initiatives or projects such that PRACE.

6.2. Chameleon

Participant: Emmanuel Agullo [corresponding member].

Chameleon is part of the MORSE (Matrices Over Runtime Systems @ Exascale) project. The overall objective is to develop robust linear algebra libraries relying on innovative runtime systems that can fully benefit from the potential of those future large-scale complex machines.

6.3. MaPHyS

Participant: Emmanuel Agullo [corresponding member].

MaPHyS (Massively Parallel Hybrid Solver) is a software package that implements a parallel linear solver coupling direct and iterative approaches. The underlying idea is to apply to general unstructured linear systems domain decomposition ideas developed for the solution of linear systems arising from PDEs. The interface problem, associated with the so called Schur complement system, is solved using a block preconditioner with overlap between the blocks that is referred to as Algebraic Additive Schwarz.

The **MaPHyS** package is very much a first outcome of the research activity described in Section 3.3 . Finally, **MaPHyS** is a preconditioner that can be used to speed-up the convergence of any Krylov subspace method. We foresee to either embed in **MaPHyS** some Krylov solvers or to release them as standalone packages, in particular for the block variants that will be some outcome of the studies discussed in Section 3.3 .

MaPHyS can be found at <http://maphys.gforge.inria.fr>.

6.4. PaStiX

Participant: Pierre Ramet [corresponding member].

Complete and incomplete supernodal sparse parallel factorizations.

PaStiX (Parallel Sparse matriX package) 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 single or double precision (real or complex): LLt (Cholesky), LDLt (Crout) and LU with static pivoting (for non symmetric matrices having a symmetric pattern).

The **PaStiX** library uses the graph partitioning and sparse matrix block ordering package **Scotch**. **PaStiX** is based on an efficient static scheduling and memory manager, in order to solve 3D problems with more than 50 million 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 the aggregated memory overhead, while keeping good performance.

Another important point is that our study is suitable for any heterogeneous parallel/distributed architecture when its performance is predictable, such as clusters of multicore nodes. In particular, we now offer a high performance version with a low memory overhead for multicore node architectures, which fully exploits the advantage of shared memory 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 in defining 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 breakthroughs 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.

PaStiX is publicly available at <http://pastix.gforge.inria.fr> under the Inria CeCILL licence.

6.5. HIPS

Participant: Pierre Ramet [corresponding member].

Multilevel method, domain decomposition, Schur complement, parallel iterative solver.

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

The key point of the methods implemented in **HIPS** is to define 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 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 sub-graphs viewed as sub-domains) is solved using an iterative method (GMRES or Conjugate Gradient at the time being). This special ordering and partitioning allows for 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. The code provides a hybrid method which blends direct and iterative solvers. **HIPS** exploits the partitioning and multistage ILU techniques to enable a highly parallel scheme where several subdomains can be assigned to the same process. It also provides a scalar preconditioner based on the multistage ILUT factorization.

HIPS can be used as a standalone program that reads a sparse linear system from a file ; it also provides an interface to be called from any C, C++ or Fortran code. It handles symmetric, unsymmetric, real or complex matrices. Thus, **HIPS** is a software library that provides several methods to build an efficient preconditioner in almost all situations.

HIPS is publicly available at <http://hips.gforge.inria.fr> under the Inria CeCILL licence.

6.6. MetaPart

Participant: Aurélien Esnard [corresponding member].

MetaPart is a library that addresses the challenge of (dynamic) load balancing for emerging complex parallel simulations, such as multi-physics or multi-scale coupling applications. First, it offers a uniform API over state-of-the-art (hyper-) graph partitioning software packages such as **Scotch**, PaToH, METIS, Zoltan, Mondriaan, etc. etc. Based upon this API, it provides a framework that facilitates the development and the evaluation of high-level partitioning methods, such as MxN repartitioning or coupling-aware partitioning (co-partitioning).

The framework is publicly available at Inria Gforge: <http://metapart.gforge.inria.fr>.

6.7. MPICPL

Participant: Aurélien Esnard [corresponding member].

MPICPL (MPI CouPLing) is a software library dedicated to the coupling of parallel legacy codes, that are based on the well-known MPI standard. It proposes a lightweight and comprehensive programming interface that simplifies the coupling of several MPI codes (2, 3 or more). MPICPL facilitates the deployment of these codes thanks to the *mpicplrun* tool and it interconnects them automatically through standard MPI inter-communicators. Moreover, it generates the universe communicator, that merges the world communicators of all coupled-codes. The coupling infrastructure is described by a simple XML file, that is just loaded by the *mpicplrun* tool.

MPICPL was developed by HIEPACS for the purpose of the ANR NOSSI. It uses advanced features of MPI2 standard. The framework is publicly available at Inria Gforge: <http://mpicpl.gforge.inria.fr>.

6.8. ScalFMM

Participant: Olivier Coulaud [corresponding member].

ScalFMM (Parallel Fast Multipole Library for Large Scale Simulations) is a software library to simulate N-body interactions using the Fast Multipole Method.

ScalFMM intends to offer all the functionalities needed to perform large parallel simulations while enabling an easy customization of the simulation components: kernels, particles and cells. It works in parallel in a shared/distributed memory model using OpenMP and MPI. The software architecture has been designed with two major objectives: being easy to maintain and easy to understand. There are two main parts: 1) the management of the octree and the parallelization of the method ; 2) the kernels. This new architecture allows us to easily add new FMM algorithm or kernels and new paradigm of parallelization. The code is extremely documented and the naming convention fully respected. Driven by its user-oriented philosophy, **ScalFMM** is using CMAKE as a compiler/installer tool. Even if **ScalFMM** is written in C++ it will support a C and fortran API soon.

The library offers two methods to compute interactions between bodies when the potential decays like $1/r$. The first method is the classical FMM based on spherical harmonic expansions and the second is the Black-Box method which is an independent kernel formulation (introduced by E. Darve at Stanford). With this method, we can now easily add new non oscillatory kernels in our library. For the classical method, two approaches are used to decrease the complexity of the operators. We consider either matrix formulation that allows us to use BLAS routines or rotation matrix to speed up the M2L operator.

The **ScalFMM** package is available at <http://scalform.gforge.inria.fr>

6.9. ViTE

Participant: Mathieu Faverge [corresponding member].

Visualization, Execution trace

ViTE is a trace explorer. It is a tool made to visualize execution traces of large parallel programs. It supports Pajé, a trace format created by Inria Grenoble, and OTF and OTF2 formats, developed by the University of Dresden and allows the programmer a simpler way to analyse, debug and/or profile large parallel applications. It is an open source software licenced under CeCILL-A.

The **ViTE** software is available at <http://vite.gforge.inria.fr> and has been developed in collaboration with the Inria Bordeaux - Sud-Ouest SED team, Telecom SudParis and Inria Grenoble.

In the same context we also contribute to the EZtrace and GTG libraries in collaboration with F. Trahay from Telecom SudParis. EZTrace (<http://eztrace.gforge.inria.fr>) is a tool that aims at generating automatically execution trace from HPC programs. It generates execution trace files thanks to the GTG library (<http://gtg.gforge.inria.fr>) that can be later interpreted by visualization tools such as **ViTE**.

6.10. Other software

For the materials physics applications, a lot of development will be done in the context of older ANR projects (**NOSSI** and **OPTIDIS**) in collaboration with LaBRI, CPMOH, IPREM, EPFL and with CEA Saclay and Bruyère-le-Châtel.

- **OptiDis**

Participant: Olivier Coulaud [corresponding member].

OptiDis is a new code for large scale dislocation dynamics simulations. Its aim is to simulate real life dislocation densities (up until $5 \cdot 10^{22}$ dislocations/ m^{-2}) in order to understand plastic deformation and study strain hardening. The main application is to observe and understand plastic deformation on irradiated zirconium. Zirconium alloys is the first containment barrier against the dissemination of radioactive elements. More precisely, with neutron irradiated zirconium alloys we are talking of channeling mechanism, which means to stick with the reality, more than tens of thousands of induced loops so 10^8 degrees of freedom in the simulation.

The code is based on Numodis code developed at CEA Saclay and the **ScalFMM** library developed in our Inria project. The code is written in C++ language and using the last features of C++11. One of the main aspects is the hybrid parallelism MPI/OpenMP that gives the software the ability to scale on large cluster while the computation load rises. In order to achieve that, we use different levels of parallelism. First of all, the simulation box is spread over MPI processes, we then use a thinner level for threads, dividing the domain using an Octree representation. All these parts are driven by the **ScalFMM** library. On the last level our data are stored in an adaptive structure absorbing dynamic of this kind of simulation and handling well task parallelism.

The two following packages are mainly designed and developed in the context of a US initiative led by ICL and to which we closely collaborate through the associate team **MORSE**.

- **PLASMA**

Participant: Mathieu Faverge [corresponding member].

The **PLASMA** (Parallel Linear Algebra for Scalable Multi-core Architectures) project aims at addressing the critical and highly disruptive situation that is facing the Linear Algebra and High Performance Computing community due to the introduction of multi-core architectures.

The **PLASMA** ultimate goal is to create software frameworks that enable programmers to simplify the process of developing applications that can achieve both high performance and portability across a range of new architectures.

The development of programming models that enforce asynchronous, out of order scheduling of operations is the concept used as the basis for the definition of a scalable yet highly efficient software framework for Computational Linear Algebra applications.

The **PLASMA** library is available at <http://icl.cs.utk.edu/plasma>.

- **PaRSEC/DPLASMA**

Participant: Mathieu Faverge [corresponding member].

PaRSEC Parallel Runtime Scheduling and Execution Controller, is a generic framework for architecture aware scheduling and management of micro-tasks on distributed many-core heterogeneous architectures. Applications we consider can be expressed as a Direct Acyclic Graph of tasks with labeled edges designating data dependencies. DAGs are represented in a compact problem-size independent format that can be queried on-demand to discover data dependencies in a totally distributed fashion. **PaRSEC** assigns computation threads to the cores, overlaps communications and computations and uses a dynamic, fully-distributed scheduler based on architectural features such as NUMA nodes and algorithmic features such as data reuse.

The framework includes libraries, a runtime system, and development tools to help application developers tackle the difficult task of porting their applications to highly heterogeneous and diverse environments.

DPLASMA (Distributed Parallel Linear Algebra Software for Multicore Architectures) is the leading implementation of a dense linear algebra package for distributed heterogeneous systems. It is designed to deliver sustained performance for distributed systems where each node featuring multiple sockets of multicore processors, and if available, accelerators like GPUs or Intel Xeon Phi. **DPLASMA** achieves this objective through the state of the art **PaRSEC** runtime, porting the **PLASMA** algorithms to the distributed memory realm.

The **PaRSEC** runtime and the **DPLASMA** library are available at <http://icl.cs.utk.edu/parsec>.

6.11. Platforms

6.11.1. *PlaFRIM: an experimental parallel computing platform*

PLAFRIM is an open experimental platform for research and development in modeling, simulations and high performance computing. This platform has been set up from 2009 under the leadership of Inria Bordeaux Sud-Ouest in collaboration with computer science and mathematics laboratories, respectively Labri and IMB with a strong support in the region Aquitaine. Since mid-2015, this platform is now open to laboratories involved in the CPU cluster and SMIs located in the region Aquitaine.

It aggregates different kinds of computational resources for research and development purposes. The latest technologies in terms of processors, memories and architecture are added when they are available on the market. It is now more than 2,500 cores (excluding GPU and Xeon Phi) that are available for all research teams of Inria Bordeaux, Labri and IMB. Those computers are in particular used by all the engineers who work in HiePACS and are advised by F. Rue from the SED.

The PlaFRIM platform initiative is coordinated by O. Coulaud.

LFANT Project-Team

5. New Software and Platforms

5.1. APIP

Another Pairing Implementation in PARI

SCIENTIFIC DESCRIPTION

Apip , Another Pairing Implementation in PARI, is a library for computing standard and optimised variants of most cryptographic pairings.

The following pairings are available: Weil, Tate, ate and twisted ate, optimised versions (à la Vercauteren–Hess) of ate and twisted ate for selected curve families.

The following methods to compute the Miller part are implemented: standard Miller double-and-add method, standard Miller using a non-adjacent form, Boxall et al. version, Boxall et al. version using a non-adjacent form.

The final exponentiation part can be computed using one of the following variants: naive exponentiation, interleaved method, Avanzi–Mihalescu’s method, Kato et al.’s method, Scott et al.’s method.

Part of the library has been included into PARI/GP proper.

FUNCTIONAL DESCRIPTION

APIP is a library for computing standard and optimised variants of most cryptographic pairings.

- Participant: Jérôme Milan
- Contact: Jérôme Milan
- URL: <http://www.lix.polytechnique.fr/~milanj/apip/apip.xhtml>

5.2. Arb

FUNCTIONAL DESCRIPTION

Arb is a C library for arbitrary-precision floating-point ball arithmetic. It supports real and complex numbers, polynomials, power series, matrices, and evaluation of many transcendental functions. All is done with automatic, rigorous error bounds. It has been accepted for inclusion in SageMath.

- Participant: Fredrik Johansson
- Contact: Fredrik Johansson
- URL: <http://fredrikj.net/arb/>

5.3. AVIsogenies

Abelian Varieties and Isogenies

FUNCTIONAL DESCRIPTION

AVIsogenies is a Magma package for working with abelian varieties, with a particular emphasis on explicit isogeny computation.

Its prominent feature is the computation of (l,l) -isogenies between Jacobian varieties of genus-two hyperelliptic curves over finite fields of characteristic coprime to l , practical runs have used values of l in the hundreds.

It can also be used to compute endomorphism rings of abelian surfaces, and find complete addition laws on them.

- Participants: Gaëtan Bisson, Romain Cosset and Damien Robert
- Contact: Damien Robert
- URL: <http://avisogenies.gforge.inria.fr/>

5.4. CM

FUNCTIONAL DESCRIPTION

The Cm software implements the construction of ring class fields of imaginary quadratic number fields and of elliptic curves with complex multiplication via floating point approximations. It consists of libraries that can be called from within a C program and of executable command line applications.

- Participant: Andreas Enge
- Contact: Andreas Enge
- URL: <http://www.multiprecision.org/index.php?prog=cm&page=home>

5.5. CMH

Computation of Igusa Class Polynomials

KEYWORDS: Mathematics - Cryptography - Number theory

FUNCTIONAL DESCRIPTION

Cmh computes Igusa class polynomials, parameterising two-dimensional abelian varieties (or, equivalently, Jacobians of hyperelliptic curves of genus 2) with given complex multiplication.

- Participants: Emmanuel Thomé, Andreas Enge and Regis Dupont
- Contact: Emmanuel Thomé
- URL: <http://cmh.gforge.inria.fr>

5.6. CUBIC

FUNCTIONAL DESCRIPTION

Cubic is a stand-alone program that prints out generating equations for cubic fields of either signature and bounded discriminant. It depends on the Pari library. The algorithm has quasi-linear time complexity in the size of the output.

- Participant: Karim Belabas
- Contact: Karim Belabas
- URL: <http://www.math.u-bordeaux.fr/~belabas/research/software/cubic-1.2.tgz>

5.7. Euclid

FUNCTIONAL DESCRIPTION

Euclid is a program to compute the Euclidean minimum of a number field. It is the practical implementation of the algorithm described in [38]. Some corresponding tables built with the algorithm are also available. Euclid is a stand-alone program depending on the PARI library.

- Participants: Pierre Lezowski and Jean-Paul Cerri
- Contact: Pierre Lezowski
- URL: <http://www.math.u-bordeaux1.fr/~plezowsk/euclid/index.php>

5.8. GNU MPC

FUNCTIONAL DESCRIPTION

Mpc is a C library for the arithmetic of complex numbers with arbitrarily high precision and correct rounding of the result. It is built upon and follows the same principles as Mpfr. The library is written by Andreas Enge, Philippe Théveny and Paul Zimmermann.

- Participants: Andreas Enge, Paul Zimmermann, Philippe Théveny and Mickaël Gastineau
- Contact: Andreas Enge
- URL: <http://www.multiprecision.org/>

5.9. KleinianGroups

FUNCTIONAL DESCRIPTION

KleinianGroups is a Magma package that computes fundamental domains of arithmetic Kleinian groups.

- Participant: Aurel Page
- Contact: Aurel Page
- URL: <http://www.normalesup.org/~page/Recherche/Logiciels/logiciels-en.html>

5.10. MPFRCX

FUNCTIONAL DESCRIPTION

Mpfrcx is a library for the arithmetic of univariate polynomials over arbitrary precision real (Mpfr) or complex (Mpc) numbers, without control on the rounding. For the time being, only the few functions needed to implement the floating point approach to complex multiplication are implemented. On the other hand, these comprise asymptotically fast multiplication routines such as Toom-Cook and the FFT.

- Participant: Andreas Enge
- Contact: Andreas Enge
- URL: <http://www.multiprecision.org/index.php?prog=mpfrcx>

5.11. Nemo

FUNCTIONAL DESCRIPTION Nemo is a computer algebra package for the Julia programming language maintained by William Hart with code by William Hart, Tommy Hofmann, Claus Fieker, Fredrik Johansson, Oleksandr Motsak).

The features of Nemo include multiprecision integers and rationals, integers modulo n , p -adic numbers, finite fields (prime and non-prime order), number field arithmetic, maximal orders of number fields, arithmetic of ideals in maximal orders, arbitrary precision real and complex balls, generic polynomials, power series, fraction fields, residue rings and matrices.

- Participant: Fredrik Johansson
- Contact: William Hart
- URL: <http://nemocas.org/>

5.12. PARI/GP

FUNCTIONAL DESCRIPTION

PARI/GP is a widely used computer algebra system designed for fast computations in number theory (factorisation, algebraic number theory, elliptic curves, ...), but it also contains a large number of other useful functions to compute with mathematical entities such as matrices, polynomials, power series, algebraic numbers, etc., and many transcendental functions.

- Participants: Karim Belabas, Henri Cohen, Andreas Enge and Hamish Ivey-Law
- Contact: Karim Belabas
- URL: <http://pari.math.u-bordeaux.fr/>

MAGIQUE-3D Project-Team

5. New Software and Platforms

5.1. Elasticus

SCIENTIFIC DESCRIPTION

Elasticus simulates acoustic and elastic wave propagation in 2D and in 3D, formulated as a first order system, using Discontinuous Galerkin Methods. The space discretization is based on two kind of basis functions, using Lagrange or Jacobi polynomials. Different kinds of fluxes (upwind and centered) are implemented, coupled with RK2 and RK4 time schemes.

FUNCTIONAL DESCRIPTION

Elasticus is a sequential library, independent of DIVA and developed in Fortran, to simulate wave propagation in geophysical environment, based on a DG method. It is meant to help PhD students and post-doctoral fellows to easily implement their algorithms in the library. Thus, readability of the code is privileged over optimization of its performances. Developed features should be easily transferred in the computing platform of Total. Contrary to DIVA which only computes approximate solutions with P1, P2 and P3 elements, Elasticus manages arbitrary orders for the spatial discretization with DG method.

IMPROVEMENT

The various kinds of fluxes and the RK4 time schemes were implemented by Simon Ettouati. The elasto-acoustic coupling was implemented by Elvira Shishenina in the framework of her Master internship, in collaboration with Simon Ettouati and Lionel Boillot. The TTI elastic kernel as well as the Absorbing Boundary Conditions were developed by Lionel Boillot.

- Participants: Simon Ettouati, Julien Diaz, Lionel Boillot and Elvira Shishenina.
- Partner: TOTAL
- Contact: Julien Diaz

5.2. Hou10ni

SCIENTIFIC DESCRIPTION

Hou10ni simulates acoustic and elastic wave propagation in time domain and in harmonic domain, in 2D and in 3D, formulated as a second order system. It is also able to model elasto acoustic coupling. It is based on the second order formulation of the wave equation and the space discretization is achieved using Interior Penalty Discontinuous Galerkin (IPDG) Method. Recently, the harmonic domain solver has been extended to handle Hybridizable Discontinuous Galerkin Methods.

FUNCTIONAL DESCRIPTION

This software simulates the propagation of waves in heterogeneous 2D and 3D media in time-domain and in frequency domain. It is based on an Interior Penalty Discontinuous Galerkin Method (IPDGM) and allows for the use of meshes composed of cells of various order (p-adaptivity in space).

IMPROVEMENT

The main improvements are related to the frequency domain part. The Hybridizable Discontinuous Galerkin was implemented in 2D and in 3D elastodynamics by Marie Bonnasse Gahot in the framework of her PhD thesis. The IPDG Method was implemented in 3D acoustics and for the 3D elasto-acoustic coupling by Conrad Hillairet in the framework of his Master thesis. The Perfectly Matched Layers were optimized (the length of the layers and the amplitude of the absorption parameters are now automatically computed) by Andrew Wang in the framework of his two months internship.

- Participants: Julien Diaz, Marie Bonnasse Gahot, Conrad Hillairet and Lionel Boillot
- Contact: Julien Diaz

5.3. Montjoie

SCIENTIFIC DESCRIPTION

Montjoie is designed for the efficient solution of time-domain and time-harmonic linear partial differential equations using high-order finite element methods. This code is mainly written for quadrilateral/hexahedral finite elements, partial implementations of triangular/tetrahedral elements are provided. The equations solved by this code, come from the "wave propagation" problems, particularly acoustic, electromagnetic, aeroacoustic, elastodynamic problems.

FUNCTIONAL DESCRIPTION

Montjoie is a code that provides a C++ framework for solving partial differential equations on unstructured meshes with finite element-like methods (continuous finite element, discontinuous Galerkin formulation, edge elements and facet elements). The handling of mixed elements (tetrahedra, prisms, pyramids and hexahedra) has been implemented for these different types of finite elements methods. Several applications are currently available : wave equation, elastodynamics, aeroacoustics, Maxwell's equations.

IMPROVEMENT

This year, new high-order schemes (internship of Guillaume Marty and thesis of Mamadou N'diaye) have been implemented. The code has been strongly modified in order to obtain a fast compilation. The software has been used by Chloe team (through the internship of Laurene Hume) and compared to COMSOL providing a similar efficiency.

- Participants: Marc Duruflé, Juliette Chabassier, Mamadou N'diaye, Guillaume Marty
- Contact: Marc Duruflé
- URL: <http://montjoie.gforge.inria.fr/>

5.4. TMBM-DG

SCIENTIFIC DESCRIPTION

TMBM-DG simulates acoustic and elastic wave propagation in 2D and in 3D, formulated as a first order system, using Discontinuous Galerkin Methods. The space discretization is based on two kinds of basis functions, using Lagrange or Jacobi polynomials coupled with RK2 and RK4 time schemes.

FUNCTIONAL DESCRIPTION

TMBM-DG is the follow up to DIVA-DG that we develop in collaboration with our partner Total. Its purpose is more general than DIVA-DG and should contains various DG schemes, basis functions and time schemes. It models wave propagation in acoustic media, elastic (isotropic and TTI) media and elasto-acoustic media, in two and three dimensions.

IMPROVEMENT

The first version of the code was recently developed jointly with our industrial partner Total. The main developer in MAGIQUE-3Dis Lionel Boillot.

- Participants: Lionel Boillot, Julien Diaz and Simon Ettouati
- Partner: TOTAL
- Contact: Julien Diaz

MANAO Project-Team

6. New Software and Platforms

6.1. ALTA Lib

The ALTA Library

KEYWORDS: Statistic analysis - Fitting - Measures

FUNCTIONAL DESCRIPTION

ALTA is a multi-platform software library to analyze, fit and understand Bidirectional Reflection Distribution Functions (BRDFs). It provides a set of command line software to fit measured data to analytical forms, tools to understand models and data.

In 2015, we continued the development of ALTA and added different unit and integration tests to reach a new milestone with our first Beta version.

- Participants: Laurent Belcour, Romain Pacanowski, Xavier Granier and Pascal Barla
- Partner: LP2N (CNRS - UMR 5298)
- Contact: Romain Pacanowski
- URL: <http://alta.gforge.inria.fr/>

6.2. Elasticity Skinning

SCIENTIFIC DESCRIPTION

Geometric skinning techniques are very popular in the industry for their high performances, but fail to mimic realistic deformations. With elastic implicit skinning the skin stretches automatically (without skinning weights) and the vertices distribution is more pleasing. Our approach is more robust, for instance the angle's range of joints is larger than implicit skinning.

This software has been ported as a plugin for the Modo software (The Foundry) in collaboration with *Toulouse Tech Transfer*. This plugin has been bought by The Foundry, which maintains and sells it.

- Participants: Rodolphe Vaillant, Loïc Barthe, Florian Canezin, Gaël Guennebaud, Marie-Paule Cani, Damien Rohmer, Brian Wyvill, Olivier Gourmel and Mathias Paulin
- Partners: Université de Bordeaux - CNRS - INP Bordeaux - Université de Toulouse - Institut Polytechnique de Grenoble - Ecole Supérieure de Chimie Physique Electronique de Lyon
- Contact: Gaël Guennebaud
- URL: <http://rodolphe-vaillant.fr/?e=59>

6.3. Eigen

FUNCTIONAL DESCRIPTION

Eigen is an efficient and versatile C++ mathematical template library for linear algebra and related algorithms. In particular it provides fixed and dynamic size matrices and vectors, matrix decompositions (LU, LLT, LDLT, QR, eigenvalues, etc.), sparse matrices with iterative and direct solvers, some basic geometry features (transformations, quaternions, axis-angles, Euler angles, hyperplanes, lines, etc.), some non-linear solvers, automatic differentiations, etc. Thanks to expression templates, Eigen provides a very powerful and easy to use API. Explicit vectorization is performed for the SSE, AltiVec and ARM NEON instruction sets, with graceful fallback to non-vectorized code. Expression templates allow to perform global expression optimizations, and to remove unnecessary temporary objects.

In 2015, we released four revisions of the 3.2 branch, and the beta-1 of the next 3.3 version.

- Participant: Gaël Guennebaud
- Contact: Gaël Guennebaud
- URL: <http://eigen.tuxfamily.org>

6.4. HDRSee

KEYWORDS: OpenGL-GLSL HDR/LDR Viewer **FUNCTIONAL DESCRIPTION** HDRSee is a OpenGL/GLSL software that displays High Dynamic Range (HDR) and Low Dynamic Range (LDR) images. It is based on several libraries (e.g., glut, see below for full dependencies). To display HDR images, HDRSee implements a few tone-mapping operators. Moreover, it is designed with a plugin mechanism that let developers add, as easily as possible, their own tone-mapping operator. All tone-mapping operations are done using Graphics Hardware through pixel shader operations. The GUI currently used is nvWidgets.

- Participants: Romain Pacanowski, Xavier Granier.
- Partner: LP2N (CNRS - UMR 5298)
- Contact: Romain Pacanowski
- URL: <http://mhdrviewer.gforge.inria.fr/>

6.5. PFSTools

KEYWORD: HDR Merging, radiometric calibration, HDR tonemapping
FUNCTIONAL DESCRIPTION

The pfstools package is a set of command line programs for reading, writing, manipulating and viewing high-dynamic range (HDR) images and video frames. All programs in the package exchange data using a simple generic high dynamic range image format, pfs , and they use unix pipes to pass data between programs and to construct complex image processing operations.

pfstools comes with a library for reading and writing pfs files. The library can be used for writing custom applications that can integrate with the existing pfstools programs. It also offers a good integration with high-level mathematical programming languages, such as MATLAB or GNU Octave. pfstools can be used as an extension for MATLAB or Octave for reading and writing HDR images or simply to effectively store large matrices. The pfstools package integrates existing high dynamic range image formats by providing a simple data format that can be used to exchange data between applications. It is accompanied by the pfscalibration and pfstmo packages.

- Participants: Rafal Mantiuk, Ivo Ihrke
- Contact: Ivo Ihrke
- URL: <http://pfstools.sourceforge.net/>

6.6. Shiver

KEYWORD: HDR Viewer
FUNCTIONAL DESCRIPTION

Shiver is a Scientific HDR Image Viewer with a convenient GUI. It features fast display / zoom OpenGL capabilities, the comparison of several images in different tabs, LDR, HDR, and raw-support through a plugin architecture, and more.

In addition, Shiver is an image processing program providing the ability to execute algorithms that are programmed as plugins on one or more images. Different frontends like the command line or a QT-based graphical user interface are available. Depending on the frontend different work flows are possible.

The console frontend can be used, if no X11 server is available or a large number of images have to be processed. The QT Gui allows for intuitive work and a test of processing plugins. It allows for example pixel picking and a comfortable way to compare different processed images.

Available Shiver plugins implement, e.g., the CalTag system for automatically detecting checkerboard corners in camera calibration images.

- Participants: Ivo Ihrke
- Contact: Ivo Ihrke
- URL: <http://sourceforge.net/projects/shiverfork/>

6.7. Maori

KEYWORD: Matlab optical raytracing toolbox

FUNCTIONAL DESCRIPTION

The purpose of the Maori project is to provide a simple, extensible, optical raytracing library in Matlab that incorporates some modern concepts from computer graphics. In particular it features scene graph integration, a shader model, CSG objects and uses non-sequential raytracing as default. The goal is to provide a simple-to-use 3D system. In contrast to most commercial systems, 2D rotationally symmetric systems are treated as special cases of the 3D setting.

- Participants: Ivo Ihrke
- Contact: Ivo Ihrke

6.8. Patate Lib

KEYWORDS: Expressive rendering - Multi-scale analysis - Material appearance - Vector graphics - 2D animation

FUNCTIONAL DESCRIPTION

Patate is a header only C++/CUDA library for graphics applications. It provides a collection of Computer Graphics techniques that incorporate the latest innovations from Inria research teams working in the field. It strives for efficiency and ease-of-use by focusing on low-level core operators and key algorithms, organized in modules, each tackling a specific set of issues. The central goal of the library is to drastically reduce the time and efforts required to turn a research paper into a ready-to-use solution, for both commercial and academic purposes.

The library is still in its infancy and we are actively working on it to include the latest of our published research techniques. Modules will be dealing with graphics domains as varied as multi-scale analysis, material appearance, vector graphics, expressive rendering and 2D animation.

- Participants: Gaël Guennebaud, Pascal Barla, Simon Boyé, Gautier Ciaudo and Nicolas Mellado
- Contact: Gaël Guennebaud
- URL: <http://patate.gforge.inria.fr/html/>

6.9. Radiance Scaling for MeshLab

FUNCTIONAL DESCRIPTION The Radiance Scaling technique has received some interest in the Archaeology community, for enhancing details in carved stones in particular. For this reason, we have made it available as a plugin for the Open Source software Meshlab.

- Participants: Romain Vergne, Olivier Dumas and Pascal Barla
- Contact: Pascal Barla
- URL: <http://manao.inria.fr/category/software/>

MEMPHIS Team

6. New Software and Platforms

6.1. New Software

6.1.1. NaSCar

This code is devoted to solve 3D-flows past moving and deformable bodies. The incompressible Navier-Stokes equations are solved on fixed grids, and the bodies are taken into account thanks to penalization and/or immersed boundary methods. The interface between the fluid and the bodies is tracked with a level set function or in a Lagrangian way. The numerical code is fully second order (time and space). The numerical method is based on projection schemes of Chorin-Temam type. The code is written in C language and use Petsc (<http://www.mcs.anl.gov/petsc/petsc-as/>) library for the resolution of large linear systems in parallel. NaSCar can be used to simulate both hydrodynamic bio-locomotion as fish like swimming and aerodynamic flows such wake generated by a wind turbine.

- Main developer: M. Bergmann.
- Version: 1
- Keywords: numerical analysis, fluid mechanics, language C, PETSc
- Software benefit: flow around deformable obstacles, moving into a fluid.
- APP: in progress
- OS/Middleware: unix, linux, mac os
- Required library or software: PETSc item Programming language: C
- Documentation: in progress

6.1.2. NSMulti

The code is devoted to solve incompressible flows modeled by Navier-Stokes equations in two or three-dimensions. The equation of temperature can be added as well as Oldroyd-B model for viscous-elastic fluids. The two-dimensional version allows many sets of boundary conditions based on Dirichlet boundary condition, open boundary condition and periodic boundary condition. Bodies immersed in the fluid are taken into account by means of the volume penalization method as the code uses only uniform Cartesian meshes. The approximation is performed efficiently by a second order scheme for the linear terms and an upwind third order scheme for the convection terms. An efficient multigrid algorithm is used to accelerate the convergence. The whole code is written in FORTRAN 95 with MPI parallelization. When it is possible an hybrid MPI/OPEN MP parallelization is applied. The code yields the approximate solution at chosen times as well as the mean flow. In addition the time evolution of the main quantities at given points and global physical quantities such as the energy, the enstrophy, the lift, the drag are provided.

- Main developer: C.-H. Bruneau
- Version: 3
- Keywords: Incompressible flows, language FORTRAN95, MPI, OPEN MP.
- Software benefit : flow around solid or porous obstacles.
- APP: in progress
- OS/Middleware: unix, linux, mac os
- Required library or software: none
- Documentation: integrated

6.1.3. CoCoFlo

This is a research code to solve compressible multi-material flows modeled by conservation laws and hyperelastic constitutive models in three-dimensions. The whole code is written in FORTRAN 95 with MPI parallelization.

- Main developer: experimental code with contributions from past PhD students, mainly A. de Brauer and Y. Gorsse under the supervision of A. Iollo.
- Version: 0
- Keywords: Compressible material, language FORTRAN95, MPI, OPEN MP.
- Software benefit : impacts.
- APP: not foreseen
- OS/Middleware: unix, linux, mac os
- Required library or software: none
- Documentation: integrated

6.1.4. KOPPA

This code solves a polyatomic extension of the BGK or the ES-BGK models on octree meshes in parallel (Kinetic Octree Parallel Poly Atomic: KOPPA). It is a finite-volume code second-order accurate scheme in space and time with immersed boundaries. In collaboration with STORM team of Inria, Optimad and CINECA a porting on multi-integrated cores (XEON Phi for the moment) of this code is in progress.

- Main developer: F. Bernard
- Version: 0
- Keywords: Rarefied flows language C++, MPI, OPEN MP.
- Software benefit: simulation of non-equilibrium reentry flows, satellite nozzle plumes.
- APP: not foreseen at the moment
- OS/Middleware: unix, linux, mac os
- Required library or software: PABLO for octree.
- Documentation: integrated

MNEMOSYNE Project-Team

6. New Software and Platforms

6.1. Positioning

Our previous works in the domain of well-defined distributed asynchronous adaptive computations [62], [59], [64] have already made us define a library (DANA [58]), closely related to both the notion of artificial neural networks and cellular automata. From a conceptual point of view, the computational paradigm supporting the library is grounded on the notion of a unit that is essentially a (vector of) potential that can vary along time under the influence of other units and learning. Those units can be organized into layers, maps and networks.

We will also have to interact with the High Performance Computing (HPC) community, since having large scale simulations at that mesoscopic level is an important challenge in our systemic view of computational neuroscience. Our approach implies to emulate the dynamics of thousands, or even millions, of integrated computational units, each of them playing the role of a whole elementary neural circuit (e.g. the microcolumn for the cortex). Mesoscopic models are considered in such an integrative approach, in order to exhibit global dynamical effect that would be hardly reachable by compartment models involving membrane equations or even spiking neuron networks.

The vast majority of high performance computing softwares for computational neuroscience addresses sub-neural or neural models [48], but coarser grained population models are also demanding for large scale simulations, with fully distributed computations, without global memory or time reference, as it is specified in (*cf.* § 3.2).

6.2. DANA

Distributed Asynchronous Numerical & Adaptive computing framework
FUNCTIONAL DESCRIPTION

DANA is a python framework whose computational paradigm is grounded on the notion of a unit that is essentially a set of time dependent values varying under the influence of other units via adaptive weighted connections. The evolutions of a unit's value are defined by a set of differential equations expressed in standard mathematical notation which greatly ease their definition. The units are organized into groups that form a model. Each unit can be connected to any other unit (including itself) using a weighted connection. The DANA framework offers a set of core objects needed to design and run such models. The modeler only has to define the equations of a unit as well as the equations governing the training of the connections. The simulation is completely transparent to the modeler and is handled by DANA. This allows DANA to be used for a wide range of numerical and distributed models as long as they fit the proposed framework (e.g. cellular automata, reaction-diffusion system, decentralized neural networks, recurrent neural networks, kernel-based image processing, etc.).

- Participant: Nicolas Rougier
- Contact: Nicolas Rougier
- URL: <http://dana.loria.fr/>

6.3. Virtual Enaction

KEYWORDS: Neurosciences - Simulation - Health
FUNCTIONAL DESCRIPTION

VirtualEnaction: A Platform for Systemic Neuroscience Simulation. The computational models studied in this project have applications that extend far beyond what is possible to experiment yet in human or non-human primate subjects. Real robotics experimentations are also impaired by rather heavy technological constraints, for instance, it is not easy to dismantle a given embedded system in the course of emerging ideas. The only versatile environment in which such complex behaviors can be studied both globally and at the level of details of the available modeling is a virtual environment, as in video games, Such a system can be implemented as “brainy-bot” (a programmed player based on our knowledge of the brain architecture) which goal is to survive in a complete manipulable environment.

In order to attain this rather ambitious objective we both (i) deploy an existing open-source video game middleware in order to be able to shape the survival situation to be studied and (ii) revisit the existing models in order to be able to integrate them as an effective brainy-bot. It consists of a platform associated to a scenario that is the closest possible to a survival situation (foraging, predator-prey relationship, partner approach to reproduction) and in which it is easy to integrate an artificial agent with sensory inputs (visual, touch and smell), emotional and somatosensory cues (hunger, thirst, fear, ..) and motor outputs (movement, gesture, ..) connected to a "brain" whose architecture will correspond to the major anatomical regions involved in the issues of learning and action selection (cortex areas detailed here, basal ganglia, hippocampus, and areas dedicated to sensorimotor processes). The internal game clock can be slowed down enough to be able to run non trivial brainy-bot implementations. This platform has already being used by two students of the team and is now a new deliverable of the KEOpS project.

- Participants: André Garenne, Frédéric Alexandre, Nicolas Rougier and Thierry Viéville
- Contact: Frédéric Alexandre
- URL: <http://virtualenaction.gforge.inria.fr/>

Monc Team

6. New Software and Platforms

6.1. CADMOS

KEYWORDS: Health - Cancer - Partial differential equation - Cartesian grid

- Participants: Olivier Saut and Julien Jouganous
- Partners: Université de Bordeaux - CNRS - INP Bordeaux
- Contact: Olivier Saut

6.2. Carcinom (Computer-Assisted Research about Cancer growth and INSights on Oncological Mechanisms)

KEYWORDS: Cancer - Data modeling - Regression

- Participants: Sébastien Benzekry
- Contact: Sébastien Benzekry
- FUNCTIONAL DESCRIPTION A software for nonlinear regression of tumor growth and therapy models and statistical inference. This software is primarily designed to perform a modeling analysis of tumor growth kinetics. Given a data set of longitudinal measurements of tumor size in a population, it fits several models of tumor growth, computes several goodness-of-fit statistical metrics, identifies the parameters of the models and estimates the uncertainty associated to their determination. It provides several graphical and numerical outputs (in the form of LaTeX tables).

6.3. ELMO (Numerical Simulation of cell electroporation)

KEYWORDS: Bioinformatics - Biology - Numerical electroporation - Finite difference method in 2D-3D

- Participants: Clair Poignard and Michael Leguebe
- Partners: Université de Bordeaux - CNRS - INP Bordeaux
- Contact: Michael Leguebe
- URL: http://www.math.u-bordeaux1.fr/~mleguebe/phd_fr.html
- SCIENTIFIC DESCRIPTION 2D-3D code of finite difference method in C++ to compute the electroquasistatic field in a biological cell, with non-linear model of membrane conductance and lateral diffusion of lipids.
- FUNCTIONAL DESCRIPTION Compute the electroquasistatic field and the porated region of the cell membrane. The aim is to provide a user-friendly code for applied mathematicians and biophysicists.

6.4. Meta-poumon

KEYWORDS: Health - Evolution - Cancer - Medical imaging

- Participants: Olivier Saut, Thierry Colin, Marie Martin and Julien Jouganous
- Partners: Université de Bordeaux - CNRS - IPB
- Contact: Olivier Saut
- FUNCTIONAL DESCRIPTION The software evaluates the aggressiveness of pulmonary metastasis or response to treatment for predictive goal. To do this, we use a mathematical model based on a set of equations to nonlinear partial differential equations. This model is calibrated to the patient data using a longitudinal sequence of CT or MRI of the patient.

6.5. Nenuphar

KEYWORDS: Modeling - Oncologie - Cancer - Partial differential equation - Medical - Medical imaging

- Partners: CNRS - INP Bordeaux - Université Bordeaux 1
- Contact: Marie Martin
- FUNCTIONAL DESCRIPTION The goal of project is to evaluate the aggressiveness of a tumor or its response to therapy. For that purpose, we use a mathematical model based on a set of nonlinear partial differential equations. This model is calibrated on patient data using a longitudinal sequence of CT Scan or MRI of the patient. This approach has been validated on about 35 clinical cases of lung metastases from various primary tumors (kidney, bladder, thyroid). Using two initial images showing the targeted lesion, we recover the patient-specific parameters of the model. The evolution of the disease is then predicted by letting the model run for later times with these parameters.

6.6. SESAR (Monitor of the effect of RT on Retroperitoneal Sarcoma)

KEYWORDS: Segmentation - Health - DICOM - Cancer - Medical imaging

- Partner: Institut Bergonié
- Contact: Olivier Saut

6.7. SegmentIt

KEYWORDS: Health - Signal - Registration of 2D and 3D multimodal images - 3D - Image analysis - Image - Processing - Medical imaging

- Participants: Thierry Colin, Olivier Saut, Vivien Pianet, Agathe Peretti, Marie Martin, Sébastien Benzekry, Baudoin Denis De Senneville, Cynthia Perier, Benjamin Taton, Nicolas Grenier and Christian Combe
- Contact: Benjamin Taton
- FUNCTIONAL DESCRIPTION Image processing software for anatomical and functional data. Segmentation, registration and digital filtering. Assesment of the kidney perfusion and the kidney function (to be continued).

PHOENIX Project-Team

6. New Software and Platforms

6.1. DiaSuite: a Development Environment for Sense/Compute/Control

Applications

Participants: Charles Consel [correspondent], Julien Durand, Milan Kabac, Paul Van Der Walt, Adrien Carteron.

Despite much progress, developing a pervasive computing application remains a challenge because of a lack of conceptual frameworks and supporting tools. This challenge involves coping with heterogeneous devices, overcoming the intricacies of distributed systems technologies, working out an architecture for the application, encoding it in a program, writing specific code to test the application, and finally deploying it.

DIASUITE is a suite of tools covering the development life-cycle of a pervasive computing application.

Defining an application area. First, an expert defines a catalog of entities, whether hardware or software, that are specific to a target area. These entities serve as building blocks to develop applications in this area. They are gathered in a taxonomy definition, written in the taxonomy layer of the DIASPEC language.

Designing an application. Given a taxonomy, the architect can design and structure applications. To do so, the DIASPEC language provides an application design layer [39]. This layer is dedicated to an architectural pattern commonly used in the pervasive computing domain [35]. Describing the architecture application allows to further model a pervasive computing system, making explicit its functional decomposition.

Implementing an application. We leverage the taxonomy definition and the architecture description to provide dedicated support to both the entity and the application developers. This support takes the form of a Java programming framework, generated by the DIAGEN compiler. The generated programming framework precisely guides the developer with respect to the taxonomy definition and the architecture description. It consists of high-level operations to discover entities and interact with both entities and application components. In doing so, it abstracts away from the underlying distributed technologies, providing further separation of concerns.

Testing an application. DIAGEN generates a simulation support to test pervasive computing applications before their actual deployment. An application is simulated in the DIASIM tool, without requiring any code modification. DIASIM provides an editor to define simulation scenarios and a 2D-renderer to monitor the simulated application. Furthermore, simulated and actual entities can be mixed. This hybrid simulation enables an application to migrate incrementally to an actual environment.

Deploying a system. Finally, the system administrator deploys the pervasive computing system. To this end, a distributed systems technology is selected. We have developed a back-end that currently targets the following technologies: Web Services, RMI, SIP and OSGI. This targeting is transparent for the application code. The variety of these target technologies demonstrates that our development approach separates concerns into well-defined layers.

This development cycle is summarized in the Figure 2 .

See also the web page <http://diasuite.inria.fr>.

6.1.1. DiaSpec: a Domain-Specific Language for Networked Entities

The core of the DIASUITE development environment is the domain specific language called DIASPEC and its compiler DIAGEN.

DIASPEC It is composed of two layers.

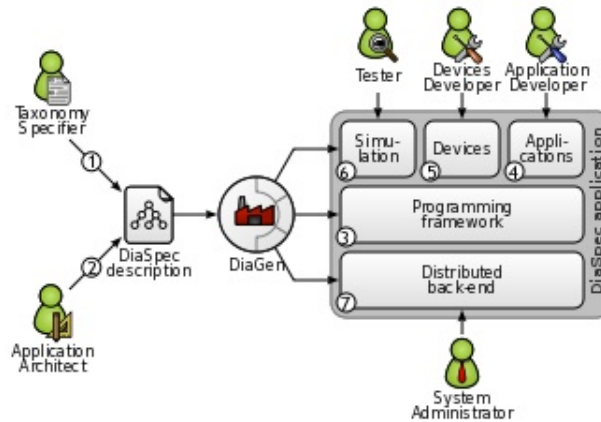


Figure 2. DIASUITE Development Cycle

The Taxonomy Layer. It allows the declaration of entities that are relevant to the target application area. An entity consists of sensing capabilities, producing data, and actuating capabilities, providing actions. Accordingly, an entity description declares a data source for each one of its sensing capabilities. As well, an actuating capability corresponds to a set of method declarations. An entity declaration also includes attributes, characterizing properties of entity instances. Entity declarations are organized hierarchically allowing entity classes to inherit attributes, sources and actions. A taxonomy allows separation of concerns in that the expert can focus on the concerns of cataloging area-specific entities. The entity developer is concerned about mapping a taxonomical description into an actual entity, and the application developer concentrates on the application logic.

The Architecture Layer. It is based on an architectural pattern commonly used in the pervasive computing domain [35]. It consists of context components fueled by sensing entities. These components process gathered data to make them amenable to the application needs. Context data are then passed to controller components that trigger actions on entities. Using an architecture description enables the key components of an application to be identified, allowing their implementation to evolve with the requirements (*e.g.*, varying light management implementations in a controller component to optimize energy consumption).

DIAGEN. It is the DIASPEC compiler that performs both static and runtime verifications over DIASPEC declarations and produces a dedicated programming framework that guides and eases the implementation of components. The generated framework is independent of the underlying distributed technology. As of today, DIAGEN supports multiple targets: Local, RMI, SIP, Web Services and OSGI.

6.2. DiaSuiteBox: an Open Orchestration Platform

Participants: Charles Consel, Julien Durand, Adrien Carteron, Milan Kabac.

The DiaSuiteBox platform runs an open-ended set of applications leveraging a range of appliances and web services. Our solution consists of a dedicated development environment, a certifying application store, and a lightweight runtime platform. This solution is based on the DIASUITE project.

The DiaSuiteBox platform can be embedded in a small plug-computer or deployed in the cloud. Thanks to the application store and the developer community, the platform is fed by a full offer of new innovative applications. During the submission process, an application is automatically analyzed and checked in order to be certified. The user is ensured the behavior of its applications are innocuous and correct with respect to the provided information. Finally, DiaSuiteBox provides an extensible software architecture. This allows the easily

connect new device technologies to the platform. For example, the support for new wireless communication technologies such as Zigbee, Z-Wave or Sigfox can be easily added to the DiaSuiteBox platform.

More details can be found on the web page <http://diasuitebox.inria.fr>.

The iQSpot startup uses DiaSuiteBox as a software platform to ease the management of Smart Buildings. In this project, the DiaSuiteBox platform is first used to host building management functionalities such as lighting management, heating/ventilating/air conditioning management, energy efficiency monitoring. It is also used to host software drivers that allow the building management functionalities to interact with the connected devices deployed in buildings. These devices can use wired communication technologies such as LonWorks, BACNet or KNX, as well as wireless communication technologies such as Z-Wave or Zigbee.

6.3. DiaSwarm: Orchestrating masses of objects

Participants: Charles Consel [correspondent], Milan Kabac, Eugène Volanschi.

DiaSwarm provides a design language dedicated to the domain of orchestrating masses of objects. The language provides high-level, declarative constructs that allow a developer to deal with masses of objects at design time, prior to programming the application. The DiaSwarm compiler generates programming frameworks, which provide high-level support to the developer, while ensuring that programming is driven by design.

DiaSwarm consists of two main components.

- DiaSwarm Studio. It is available as an Eclipse plugin. The tool provides of a domain-specific language and a compiler dedicated to the domain of orchestrating masses of sensors.
- Runtime environment. This is a library allowing you to execute DiaSwarm applications locally. The runtime environment comprises the Apache Hadoop framework v. 1.2.1 allowing you to run MapReduce jobs locally without the need to install the framework separately.

More details can be found on a dedicated web page <http://phoenix.inria.fr/software/diaswarm>. In particular, an example application is provided: it is dedicated to the management of parking lots in a city. It uses presence sensors to monitor the availability of parking spaces through magnetic field variations. The application defines a few context components, which transform sensed data to determine the availability of parking lots and average occupancy of parking lots in 24 hr. Furthermore, the application suggests parking lots to drivers entering the city. Please note that the availability of parking places is computed every 30 seconds. The average occupancy of parking lots is computed every 2 minutes.

6.4. School+ Apps: Assistive tablet applications for school Inclusion

Participants: Charles Consel [correspondent], H el ene Sauz eon, Charles Fage.

School+ is a package of 7 applications. Three applications are assistive applications, guiding the child doing specific tasks. Three others are training applications made as serious games, addressing specific skills. The last application is a meta-application, comprising a link to the three training applications, with an access to statistics of their usage. For each application, data are separated from the design, meaning that every element of each application (pictures, texts, settings, etc.) can be changed at any time. Each application records a log file containing all the interactions performed by the child.

6.4.1. Assistive applications

Routines application. This application shows a list of tasks, with a short description. After clicking the starting button, a specific slideshow is shown; it decomposes a task into steps. For each step, a text and a picture can be displayed. Thumbnail of previous and next steps are also displayed. This application guides the child through classroom situations: entering classroom, taking school materials out of a backpack, writing notes, handling agenda, leaving the classroom.

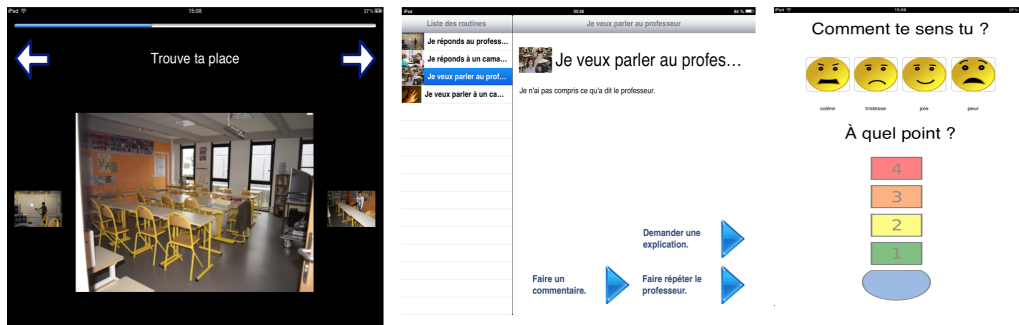


Figure 3. Assistive applications

Communication application. With the same design, the assistance provided by this application targets to communicating situations inside the classroom. The application covers four scenarios addressing two interaction situations (initiating and answering the interaction) and two types of interlocutors (professor and classmate). For each scenario, different slideshows guide the child, depending on the goal of the interaction.

Emotion Regulation application. This application aims to assist the child to self-regulate his/her emotions. Four simplified emoticons are proposed to the child to choose from: anger, sadness, joy and fear. Then, (s)he selects a level of intensity via a thermometer with a scale from 1 to 4. In response, the application delivers different multimedia contents according to the level selected to help the child regulate his/her emotions. Typically, a text (breathing instructions) are shown at level 1, pictures at level 2, a video at level 3 and another text at level 4.

6.4.2. Training applications

These three applications are serious games with increasing levels of difficulties, reachable after a ratio of good answers has been attained.

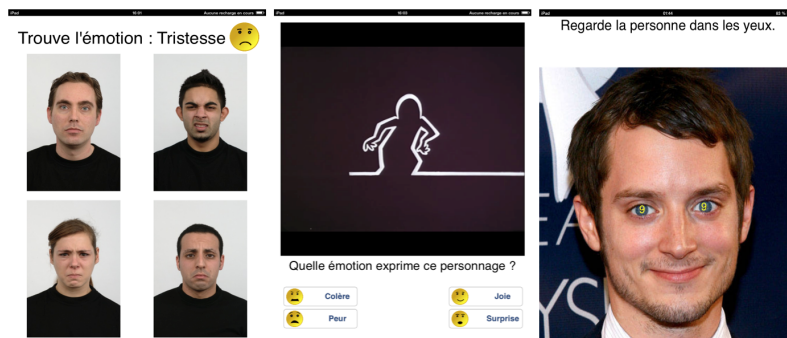


Figure 4. Training applications

Emotion Recognition application with pictures. In this application, the child is instructed to identify a specific emotion among 4 pictures showing different people exhibiting an emotion. Seven emotions are involved in this application: joy, sadness, fear, anger, surprise, disgust and neutral. The emotion to be recognized is displayed together with its simplified emoticon. The type of pictures changes with the difficulty

level: level 1 contains pictures of unfamiliar people and level 2 contains pictures of friends and relatives of the child.

Emotion Recognition application with videos. In this application, the child is presented with a fragment of an animated cartoon. At some point, the video stops and the child is asked to identify the emotion of the character. Four emotions are involved in this application: joy, sadness, fear and anger. Videos are slowed down, with a speed percentage that can be changed at each level. Videos change with difficulty level: level 1 contains videos of a very basic cartoon (only one cartoon character drawn by basic form un-textured), level 2 contains a video of more sophisticated cartoons and level 3 contains movies with actors.

Attention Training. In this application, the child is presented a picture of a face and asked to make eye contact with it. Second, a symbol appears briefly in the eyes of the character. Third, the child is asked to identify the symbol shown in the previously displayed picture, to make sure he kept eye contact. The speed at which the symbol appears and disappears is changed according to the difficulty level. Types of pictures also change with the level : level 1 contains pictures of faces and level 2 contains pictures of classroom situations.

6.5. HomeAssist: A Platform for Assistive Living

Participants: Charles Consel [correspondent], Adrien Carteron, Julien Durand, Lucile Dupuy, H el ene Sauz eon.

The HomeAssist platform proposes a systemic approach to introducing an assistive technological platform for older people. To do so, we formed a trans-disciplinary team that allows (1) to identify the user needs from a gerontological and psychological viewpoint; (2) to propose assistive applications designed by human factors and HCI experts, in collaboration with caregivers and users; (3) to develop and test applications by software engineers; (4) to conduct a field study for assessing the benefits of the platform and assistive applications, in collaboration with caregivers, by deploying the system at the actual home of older adults.

The HomeAssist platform is implemented on top of the DiaSuiteBox platform, using a suite of tools, namely DiaSuite, that have been designed, developed and tested by our research group at Inria. The DiaSuite tools include a dedicated integrated development environment that enables applications to be developed quickly and safely. This technology has been successfully applied to a variety of domains where environments consist of networked objects that need to be orchestrated.

6.5.1. Applications

HomeAssist offers an online catalog of applications. Using this catalog, the user and the caregiver determine what and how activities should be assisted by selecting the appropriate assistive applications and configuring them with respect to the user’s requirements and preferences. The resulting set of applications forms a personalized assistive support. Additionally, to respond to evolving needs, our platform allows to stop/remove applications easily and to install new ones from the online catalog.

This platform proposes many applications in three domains of everyday life.

- Daily activities: including activity monitoring, light path, and a reminder.
- Home or personal safety: including entrance monitoring, stove monitoring, and warning if no movements are detected after a certain amount of time.
- Communications and social activities: including collaborative games, videoconference, information about local events, TV programming, *etc.*

For video presentations of HomeAssist, see the following:

- <http://videotheque.inria.fr/videotheque/media/23705>. Title: “DiaSuiteBox”, 2013.
- <http://videotheque.inria.fr/videotheque/media/29998>. Title: “DomAssist : L’assistance num erique   la personne”, 2014.

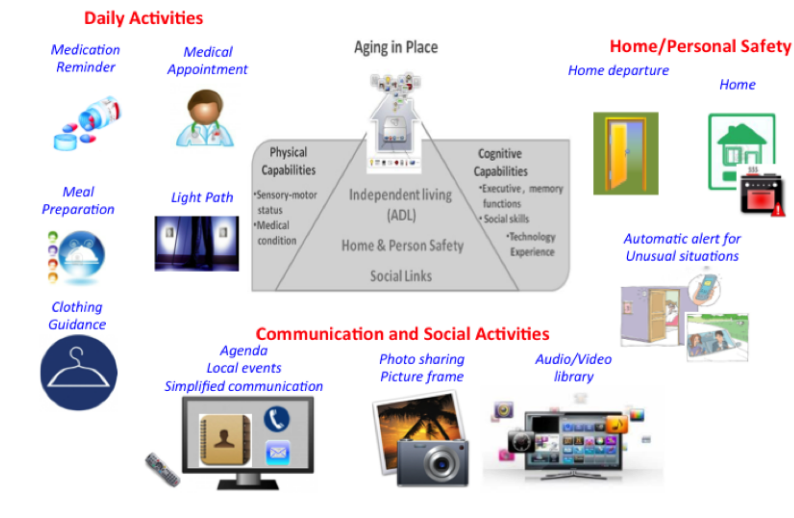


Figure 5. The HomeAssist platform and applications

6.5.2. Devices

Several entities have been identified to deliver an assistive support. These entities include (1) technological devices: wireless sensors (motion detectors, contact sensors and smart electric switches), and two tablets, and (2) software services (agenda, address book, mail agent, and photo agent) to monitor everyday activities and propose assistive applications. Sensors are placed in relevant rooms in the house: kitchen, bedroom, bathroom, and around the entrance.



Figure 6. HomeAssist devices

6.5.3. Experimental validation

A field study is currently being conducted with elderly people. The major purpose of this study is to identify the benefits of using HomeAssist for this population in an ecological framework. We selected 24 elderly people with different levels of autonomy (GIR scores). The HomeAssist technology has been installed in their house during 9 months. Twenty-four non-equipped older adults were also selected to participate to the study, as control participants.

The expected impact of HomeAssist reflects the trans-disciplinary nature of the project. We aim to deliver results in the domain of (1) elderly care, (2) ergonomics and human factors, and (3) pervasive computing.

The major expected results are that HomeAssist (1) prolongs ageing in place, improves well-being of the users, and improves the efficiency of the caregiving environment; (2) is a cognitively low-cost assistive technology, and is well accepted and perceived as useful and usable by the users; (3) is technologically robust, and is a validated assistive platform.

The preliminary results of this field study show that the platform is well adopted (highly accessible and usable) by the older users and their families or caregivers.

Additionally, these results support our claim that DiaSuiteBox is effective in terms of:

1. autonomy and well-being of older adults, comparing pre- vs. post-deployment, as well as not-equipped, control participants;
2. gains pre- vs. post deployment in management and load of caregiving tasks, and
3. longitudinal efficacy (gains unchanged at 6 vs. 9 months of use).

Most notably, this pilot project also uses a systemic approach to assistive living as illustrated by (1) its funding sources involving key stakeholders of the caregiving domain, including a territorial agency of Gironde dedicated to ageing, as well as the regional branch of the national public retirement organization (*i.e.*, CARSAT of Aquitaine), (2) their partners operating caregiving services (UDCCAS Gironde), and (3) The regional Chamber of Trades and Crafts to assist the older adults in installing and positioning the devices.

Final results are expected in March 2016.

6.5.4. External Partners

The HomeAssist platform is being developed with support from the following partners:

- Équipe “Handicap et Système Nerveux” (EA 4136), Bordeaux University
- Chaire TSA, Université du Québec Trois-Rivières
- CRIUGM, Université de Montréal
- UDCCAS Gironde
- CARSAT Aquitaine
- Conseil Général 33
- Conseil Régional d’Aquitaine

PLEIADE Team

5. New Software and Platforms

5.1. Magus

KEYWORDS: Bioinformatics - Genomic sequence - Knowledge database

SCIENTIFIC DESCRIPTION

MAGUS can be used on small installations with a web server and a relational database on a single machine, or scaled out in clusters or elastic clouds using Apache Cassandra for NoSQL data storage and Apache Hadoop for Map-Reduce.

FUNCTIONAL DESCRIPTION

The MAGUS genome annotation system integrates genome sequences and sequences features, in silico analyses, and views of external data resources into a familiar user interface requiring only a Web navigator. MAGUS implements annotation workflows and enforces curation standards to guarantee consistency and integrity. As a novel feature the system provides a workflow for simultaneous annotation of related genomes through the use of protein families identified by in silico analyses this has resulted in a three-fold increase in curation speed, compared to one-at-a-time curation of individual genes. This allows us to maintain standards of high-quality manual annotation while efficiently using the time of volunteer curators.

- Participants: Florian Lajus, David Sherman, Natalia Golenetskaya, Pascal Durrens and Xavier Calcas
- Partners: Université de Bordeaux - CNRS - INRA
- Contact: David James Sherman
- URL: <http://magus.gforge.inria.fr>

5.2. Mimoza

KEYWORDS: Systems Biology - Bioinformatics - Biotechnology

FUNCTIONAL DESCRIPTION

Mimoza uses metabolic model generalization and cartographic paradigms to allow human experts to explore a metabolic model in a hierarchical manner. The software creates a zoomable representation of a model submitted by the user in SBML format. The most general view represents the compartments of the model, the next view shows the visualization of generalized versions of reactions and metabolites in each compartment, and the most detailed view visualizes the initial model with the generalization-based layout (where similar metabolites and reactions are placed next to each other). The zoomable representation is implemented using the Leaflet JavaScript library for mobile-friendly interactive maps. Users can click on reactions and compounds to see the information about their annotations. The resulting map can be explored on-line, or downloaded in a COMBINE archive.

- Participants: Anna Zhukova and David James Sherman
- Contact: David James Sherman
- URL: <http://mimoza.bordeaux.inria.fr/>

5.3. Pantograph

KEYWORDS: Systems Biology - Bioinformatics - Genomics - Gene regulatory networks

FUNCTIONAL DESCRIPTION

Pantograph is a software toolbox to reconstruct, curate and validate genome-scale metabolic models. It uses existing metabolic models as templates, to start its reconstructions process, to which new, species-specific reactions are added. Pantograph uses an iterative approach to improve reconstructed models, facilitating manual curation and comparisons between reconstructed model's predictions and experimental evidence.

Pantograph uses a consensus procedure to infer relationships between metabolic models, based on several sources of orthology between genomes. This allows for a very detailed rewriting of reaction's genome associations between template models and the model you want to reconstruct.

- Participants: Nicolas Loira, Anna Zhukova, David James Sherman and Pascal Durrens
- Partner: University of Chile
- Contact: Nicolas Loira
- URL: <http://pathtastic.gforge.inria.fr/>

5.4. biorica

KEYWORDS: Systems Biology - Bioinformatics - Hierarchical models - Hybrid models - Stochastic models
FUNCTIONAL DESCRIPTION

BioRica is used to mathematically describe the behavior of complex biological systems.

It is a software platform that permits simulation of biological systems on the basis of their description. It allows one to reuse existing biological models and to combine them into more complex models.

- Partner: University of Chile
- Contact: David Sherman
- URL: <http://biorica.gforge.inria.fr/>

5.5. Platforms

5.5.1. Plafrim

Plafrim (<http://plafrim.fr>) is an essential instrument for PLEIADE. We use it for developing software data analysis methods and evaluating them at real world scale. The platform combines considerable computing power with excellent support, both in terms of the quality of the interactions with the local staff and of the ease of large-scale data transfer between Plafrim and PLEIADE's data storage infrastructure. Plafrim facilitates collaboration between team members who are not in the Bordeaux Sud-Ouest building, and furthermore allows us to share best practices and tools with other teams from the Center.

5.5.2. Inria forge and Inria continuous integration

The Inria forge (<http://gforge.inria.fr>) provides a secure collaboration platform for software project administration and source code management, and Inria's continuous integration platform (<http://ci.inria.fr>) provides a cloud-based service for automatic compilation and testing of software systems. PLEIADE uses these two services extensively for agile software development. The continuous integration platform allows us to verify the correct operation of our methods in different operating system and deployment environments.

POSET Team

6. New Software and Platforms

6.1. i-score

The *i-score* software, whose first definition dates back to 2005 [1], [6], aims at offering graphical views, aka *scores*, of interactive system. It has already been used to experiment various graphical user interface proposals to define time constraints between processes. It has been the subject of Jaime Arias's PhD [11]. Several formalizations of its semantics have been proposed [36], [20], [21], [19]. In 2016, especially within the ADT project "Tuilage", *i-score* independent modules should be identified and integrated as possible GUIs for the *T-calculus*.

6.2. T-calculus

Sketched in [9], the *T-calculus* is a Domain Specific Language⁰ to provide simple and robust high-level description mechanisms of reactive systems. It will offer a programmatic view of the tile modeling paradigm [3], [8]. Its definition has been refined a number of times (see e.g. [9], [7] and [30], [18]). A prototype implementation of its reactive kernel has eventually been achieved in Haskell on top of the Euterpea libraries by the end of 2015. Its consolidation and further developments are now scheduled for 2016, especially within the ADT project "Tuilage". Graphical representation of tiles should give rise to a robust correspondance between programmatic and graphical representations of reactive systems.

⁰See [40] for an early note by Hudak about the notion of Domain Specific Language, and see [39], [42] for application of this notion is computer music.

POTIOC Project-Team

6. New Software and Platforms

6.1. OpenViBE

KEYWORDS: Neurosciences - Interaction - Virtual reality - Health - Real time - Neurofeedback - Brain-Computer Interface - EEG - 3D interaction

FUNCTIONAL DESCRIPTION

OpenViBE is a software platform for real-time neurosciences (that is, for real-time processing of brain signals). It can be used to acquire, filter, process, classify and visualize brain signals in real time from various signal sources. OpenViBE is free and open source software. It works on Windows and Linux operating systems.

- Participants: Yann Renard, Anatole Lécuyer, Fabien Lotte, Bruno Renier, Vincent Delannoy, Laurent Bonnet, Baptiste Payan, Jozef Legény, Jussi Tapio Lindgren, Alison Cellard, Loïc Mahé, Guillaume Serriere, Marsel Mano, Maureen Clerc Gallagher, Théodore Papadopoulo, Laurent Bougrain, Jérémy Frey, Nathanael Foy
- Partners: INSERM - CEA-List - GIPSA-Lab
- Contact: Anatole Lécuyer
- URL: <http://openvibe.inria.fr>

In 2015, the first stable version of the OpenViBE software, OpenViBE version 1.0.0, was released. OpenViBE 1.0.0 features lots of fixes for stability and usability. There has been a significant effort in cleanup and removal of unused components and dead code. This version introduces more tools for communicating with other software, added support for some new, emerging acquisition systems, such as OpenBCI and new signal processing algorithms such as Wavelet decomposition and artifact removal boxes. Link: <http://openvibe.inria.fr/openvibe-1-0-0-has-been-released/>

6.2. Platforms

6.2.1. AMI

Augmented Michelson Interferometer

SCIENTIFIC DESCRIPTION

We have developed a hybrid platform that merges physical and virtual elements for teaching optics. This work is described in more details in Section 7.6

- Participants: David Furio, Martin Hachet, Patrick Reuter and Bruno Bousquet
- Partners: Université de Bordeaux - LaBRI
- Contact: Martin Hachet
- URL: <https://team.inria.fr/potioc/fr/2015/06/30/hobit-hybrid-optical-bench-for-innovative-teaching>

REALOPT Project-Team

6. New Software and Platforms

6.1. BaPCod : a generic Branch-And-Price Code

KEYWORDS: Column Generation - Branch-and-Price - Branch-and-Cut - Mixed Integer Programming - Mathematical Optimization - Benders Decomposition - Dantzig-Wolfe Decomposition - Extended Formulation

FUNCTIONAL DESCRIPTION: BaPCod is a prototype code that solves Mixed Integer Programs (MIP) by application of reformulation and decomposition techniques. The reformulated problem is solved using a branch-and-price-and-cut (column generation) algorithm, Benders approaches, or network flow algorithms.

- Participants: Francois Vanderbeck, Ruslan Sadykov, Issam Tahiri, Artur Alves Pessoa, Boris Detienne, François Clautiaux, Pierre Pesneau, Eduardo Uchoa Barboza and Michael Poss
- Partners: Université de Bordeaux - CNRS - IPB - Universidade Federal Fluminense
- Contact: Francois Vanderbeck
- URL: https://realopt.bordeaux.inria.fr/?page_id=2

SISTM Project-Team

6. New Software and Platforms

6.1. New Software

6.1.1. *sgPLS*

Sparse Group Partial Least Square

KEYWORD: Bioinformatics

FUNCTIONAL DESCRIPTION

The Sparse Group Partial Least Square package (sgPLS) provides sparse, group, and sparse group versions of partial least square regression models.

- Contact: Benoît Liquet
- URL: <https://cran.r-project.org/web/packages/sgPLS/index.html>

6.2. Upgraded Software

6.2.1. *TcGSA*

Time-course Gene Set Analysis

KEYWORDS: Bioinformatics - Genomics

FUNCTIONAL DESCRIPTION

An R package for the gene set analysis of longitudinal gene expression data sets. Available on the CRAN website, this package implements a Time-course Gene Set Analysis method and provides useful plotting functions facilitating the interpretation of the results.

- Contact: Boris Hejblum
- URL: <https://cran.r-project.org/web/packages/TcGSA/index.html>

6.2.2. *VSURF*

Variable Selection Using Random Forests

KEYWORD: Bioinformatics

FUNCTIONAL DESCRIPTION

An R package for Variable Selection Using Random Forests. Available on CRAN, this package performs an automatic (meaning completely data-driven) variable selection procedure. Originally designed to deal with high dimensional data, it can also be applied to standard datasets.

- Contact: Robin Genuer
- URL: <http://cran.r-project.org/web/packages/VSURF/index.html>

STORM Team

6. New Software and Platforms

6.1. Chameleon

KEYWORDS: HPC - Dense linear algebra - Task-based algorithm - Runtime system - Task scheduling

SCIENTIFIC DESCRIPTION

Chameleon is part of the MORSE (Matrices Over Runtime Systems @ Exascale) project. The overall objective is to develop robust linear algebra libraries relying on innovative runtime systems that can fully benefit from the potential of those future large-scale complex machines.

We expect advances in three directions based first on strong and closed interactions between the runtime and numerical linear algebra communities. This initial activity will then naturally expand to more focused but still joint research in both fields.

1. Fine interaction between linear algebra and runtime systems. On parallel machines, HPC applications need to take care of data movement and consistency, which can be either explicitly managed at the level of the application itself or delegated to a runtime system. We adopt the latter approach in order to better keep up with hardware trends whose complexity is growing exponentially. One major task in this project is to define a proper interface between HPC applications and runtime systems in order to maximize productivity and expressivity. As mentioned in the next section, a widely used approach consists in abstracting the application as a DAG that the runtime system is in charge of scheduling. Scheduling such a DAG over a set of heterogeneous processing units introduces a lot of new challenges, such as predicting accurately the execution time of each type of task over each kind of unit, minimizing data transfers between memory banks, performing data prefetching, etc. Expected advances: In a nutshell, a new runtime system API will be designed to allow applications to provide scheduling hints to the runtime system and to get real-time feedback about the consequences of scheduling decisions.

2. Runtime systems. A runtime environment is an intermediate layer between the system and the application. It provides low-level functionality not provided by the system (such as scheduling or management of the heterogeneity) and high-level features (such as performance portability). In the framework of this proposal, we will work on the scalability of runtime environment. To achieve scalability it is required to avoid all centralization. Here, the main problem is the scheduling of the tasks. In many task-based runtime environments the scheduler is centralized and becomes a bottleneck as soon as too many cores are involved. It is therefore required to distribute the scheduling decision or to compute a data distribution that impose the mapping of task using, for instance the so-called “owner-compute” rule. Expected advances: We will design runtime systems that enable an efficient and scalable use of thousands of distributed multicore nodes enhanced with accelerators.

3. Linear algebra. Because of its central position in HPC and of the well understood structure of its algorithms, dense linear algebra has often pioneered new challenges that HPC had to face. Again, dense linear algebra has been in the vanguard of the new era of petascale computing with the design of new algorithms that can efficiently run on a multicore node with GPU accelerators. These algorithms are called “communication-avoiding” since they have been redesigned to limit the amount of communication between processing units (and between the different levels of memory hierarchy). They are expressed through Direct Acyclic Graphs (DAG) of fine-grained tasks that are dynamically scheduled. Expected advances: First, we plan to investigate the impact of these principles in the case of sparse applications (whose algorithms are slightly more complicated but often rely on dense kernels). Furthermore, both in the dense and sparse cases, the scalability on thousands of nodes is still limited, new numerical approaches need to be found. We will specifically design sparse hybrid direct/iterative methods that represent a promising approach.

Overall end point. The overall goal of the MORSE associate team is to enable advanced numerical algorithms to be executed on a scalable unified runtime system for exploiting the full potential of future exascale machines.

FUNCTIONAL DESCRIPTION

Chameleon is a dense linear algebra software relying on sequential task-based algorithms where sub-tasks of the overall algorithms are submitted to a Runtime system. A Runtime system such as StarPU is able to manage automatically data transfers between not shared memory area (CPUs-GPUs, distributed nodes). This kind of implementation paradigm allows to design high performing linear algebra algorithms on very different type of architecture: laptop, many-core nodes, CPUs-GPUs, multiple nodes. For example, Chameleon is able to perform a Cholesky factorization (double-precision) at 80 TFlop/s on a dense matrix of order 400 000 (e.i. 4 min).

- Participants: Marc Sergent, Suraj Kumar, Samuel Thibault, Andra Hugo, Terry Cojean, Nathalie Furmento
- Partners: Innovative Computing Laboratory (ICL) - King Abdullha University of Science and Technology - University of Colorado Denver
- Contact: Emmanuel Agullo
- URL: <https://project.inria.fr/chameleon/>

6.2. KLANG-OMP

The Klang-Omp OpenMP Compiler

KEYWORDS: Compilers - Task scheduling - OpenMP - Source-to-source compiler - Data parallelism

FUNCTIONAL DESCRIPTION

The Klang-Omp software is a source-to-source OpenMP compiler for languages C and C++. The Klang-Omp compiler translates OpenMP directives and constructs into API calls from the StarPU runtime system or the XKaapi runtime system. The Klang-Omp compiler is virtually fully compliant with OpenMP 3.0 constructs. The Klang-Omp compiler supports OpenMP 4.0 dependent tasks and accelerated targets.

- Participants: Olivier Aumage, Nathalie Furmento, Samuel Pitoiset and Samuel Thibault
- Contact: Olivier Aumage
- URL: <http://kstar.gforge.inria.fr/#!index.md>

6.3. KaStORS

The KaStORS OpenMP Benchmark Suite

KEYWORDS: Benchmarking - HPC - Task-based algorithm - Task scheduling - OpenMP - Data parallelism

FUNCTIONAL DESCRIPTION

The KaStORS benchmarks suite has been designed to evaluate implementations of the OpenMP dependent task paradigm, introduced as part of the OpenMP 4.0 specification.

- Participants: Olivier Aumage, François Broquedis, Pierrick Brunet, Nathalie Furmento, Thierry Gautier, Samuel Thibault and Philippe Virouleau
- Contact: Thierry Gautier
- URL: <http://kastors.gforge.inria.fr/#!index.md>

6.4. MAQAO

SCIENTIFIC DESCRIPTION

MAQAO relies on binary codes for Intel x86 and ARM architectures. For x86 architecture, it can insert probes for instrumentation directly inside the binary. There is no need to recompile. The static/dynamic approach of MAQAO analysis is the main originality of the tool, combining performance model with values collected through instrumentation.

MAQAO has a static performance model for x86 and ARM architectures. This model analyzes performance of the codes on the architectures and provides some feed-back hints on how to improve these codes, in particular for vector instructions.

The dynamic collection of data in MAQAO enables the analysis of thread interactions, such as false sharing, amount of data reuse, runtime scheduling policy, ...

FUNCTIONAL DESCRIPTION

MAQAO is a performance tuning tool for OpenMP parallel applications. It relies on the static analysis of binary codes and the collection of dynamic information (such as memory traces). It provides hints to the user about performance bottlenecks and possible workarounds.

- Participants: Denis Barthou, Olivier Aumage, Christopher Haine and James Tombi A Mba
- Contact: Denis Barthou
- URL: <https://gforge.inria.fr/projects/maqao>

6.5. P-EDGE

An Efficient and Portable Library for Error Correction Code Design and Optimization

KEYWORDS: Code generation - Error Correction Code

FUNCTIONAL DESCRIPTION

The P-Edge library joins genericity techniques together with code generation capabilities to enable implementing efficient and portable error correction codes. The genericity offered allows to easily experiment with a large panel of algorithmic variants.

- Authors: Adrien Cassagne, Olivier Aumage, Bertrand Le Gal, Camille Leroux and Denis Barthou
- Partner: IMS
- Contact: Adrien Cassagne

6.6. StarPU

The StarPU Runtime System

KEYWORDS: HPC - Scheduling - GPU - Multicore - Performance

SCIENTIFIC DESCRIPTION

Traditional processors have reached architectural limits which heterogeneous multicore designs and hardware specialization (e.g. coprocessors, accelerators, ...) intend to address. However, exploiting such machines introduces numerous challenging issues at all levels, ranging from programming models and compilers to the design of scalable hardware solutions. The design of efficient runtime systems for these architectures is a critical issue. StarPU typically makes it much easier for high performance libraries or compiler environments to exploit heterogeneous multicore machines possibly equipped with GPGPUs or Cell processors: rather than handling low-level issues, programmers may concentrate on algorithmic concerns. Portability is obtained by the means of a unified abstraction of the machine. StarPU offers a unified offloadable task abstraction named "codelet". Rather than rewriting the entire code, programmers can encapsulate existing functions within codelets. In case a codelet may run on heterogeneous architectures, it is possible to specify one function for each architectures (e.g. one function for CUDA and one function for CPUs). StarPU takes care to schedule and execute those codelets as efficiently as possible over the entire machine. In order to relieve programmers from the burden of explicit data transfers, a high-level data management library enforces memory coherency over the machine: before a codelet starts (e.g. on an accelerator), all its data are transparently made available on the compute resource. Given its expressive interface and portable scheduling policies, StarPU obtains portable performances by efficiently (and easily) using all computing resources at the same time. StarPU also takes advantage of the heterogeneous nature of a machine, for instance by using scheduling strategies based on auto-tuned performance models.

StarPU is a task programming library for hybrid architectures

The application provides algorithms and constraints: - CPU/GPU implementations of tasks - A graph of tasks, using either the StarPU's high level GCC plugin pragmas or StarPU's rich C API

StarPU handles run-time concerns - Task dependencies - Optimized heterogeneous scheduling - Optimized data transfers and replication between main memory and discrete memories - Optimized cluster communications

Rather than handling low-level scheduling and optimizing issues, programmers can concentrate on algorithmic concerns!

FUNCTIONAL DESCRIPTION

StarPU is a runtime system that offers support for heterogeneous multicore machines. While many efforts are devoted to design efficient computation kernels for those architectures (e.g. to implement BLAS kernels on GPUs), StarPU not only takes care of offloading such kernels (and implementing data coherency across the machine), but it also makes sure the kernels are executed as efficiently as possible.

- Participants: Samuel Thibault, Nathalie Furmento, Jérôme Clet-Ortega, Pierre-André Wacrenier, Terry Cojean, Andra Hugo, Raymond Namyst, Olivier Aumage, Marc Sergent and Samuel Pitoiset.
- Contact: Raymond Namyst
- URL: <http://starpu.gforge.inria.fr/>

6.7. hwloc

Hardware Locality

KEYWORDS: HPC - Topology - Open MPI - Affinities - GPU

FUNCTIONAL DESCRIPTION

Hardware Locality (hwloc) is a library and set of tools aiming at discovering and exposing the topology of machines, including processors, cores, threads, shared caches, NUMA memory nodes and I/O devices.

It builds a widely-portable abstraction of these resources and exposes it to the application so as to help them adapt their behavior to the hardware characteristics.

hwloc targets many types of high-performance computing applications, from thread scheduling to placement of MPI processes. Most existing MPI implementations, several resource managers and task schedulers, and multiple other parallel libraries already use hwloc .

- Participants: Brice Goglin and Samuel Thibault
- Contact: Brice Goglin
- URL: <http://www.open-mpi.org/projects/hwloc/>

TADAAM Team

5. New Software and Platforms

5.1. Hardware Locality

KEYWORDS: Topology - Locality

FUNCTIONAL DESCRIPTION

Hardware Locality (HWLOC) is a library and set of tools aiming at discovering and exposing the topology of machines, including processors, cores, threads, shared caches, NUMA memory nodes and I/O devices.

It builds a widely-portable abstraction of these resources and exposes it to the application so as to help them adapt their behavior to the hardware characteristics. HWLOC also offers monitoring abilities to identify application bottlenecks [12]. Moreover it focuses on modeling the network topology by embedding the NETLOC subproject in its future releases.

HWLOC targets many types of high-performance computing applications [1], [2], from thread scheduling to placement of MPI processes. Most existing MPI implementations, many resource managers, task schedulers and parallel libraries already use HWLOC.

HWLOC is developed in collaboration within the OPEN MPI consortium. The core development is carried out by Brice GOGLIN and other members of TADAAM team-project, with external contribution from many academic and industrial partners.

HWLOC is composed of 100,000 lines of C.

- Participants: Brice Goglin, Nicolas Denoyelle, Cyril Bordage
- Contact: Brice Goglin
- URL: <http://www.open-mpi.org/projects/hwloc/>

5.2. NewMadeleine

KEYWORDS: High-performance computing - MPI communication

FUNCTIONAL DESCRIPTION

NewMadeleine is the fourth incarnation of the Madeleine communication library. The new architecture aims at enabling the use of a much wider range of communication flow optimization techniques. Its design is entirely modular: drivers and optimization strategies are dynamically loadable software components, allowing experimentations with multiple approaches or on multiple issues with regard to processing communication flows.

The optimizing scheduler SchedOpt targets applications with irregular, multi-flow communication schemes such as found in the increasingly common application conglomerates made of multiple programming environments and coupled pieces of code, for instance. SchedOpt itself is easily extensible through the concepts of optimization strategies (what to optimize for, what the optimization goal is) expressed in terms of tactics (how to optimize to reach the optimization goal). Tactics themselves are made of basic communication flows operations such as packet merging or reordering.

The communication library is fully multi-threaded through its close integration with PIOMan. It manages concurrent communication operations from multiple libraries and from multiple threads. Its MPI implementation Mad-MPI fully supports the MPI_THREAD_MULTIPLE multi-threading level.

- Participants: Alexandre Denis
- Contact: Alexandre Denis
- URL: <http://pm2.gforge.inria.fr/newmadeleine/>

5.3. PaMPA

KEYWORDS: parallel numerical solvers, unstructured heterogeneous meshes, dynamic load balancing, subdomain decomposition, parallel remeshing

FUNCTIONAL DESCRIPTION PAMPA (“Parallel Mesh Partitioning and Adaptation”) is a library dedicated to the management of distributed meshes. Its purpose is to relieve solver writers from the tedious and error-prone task of writing again and again service routines for mesh handling, data communication and exchange, remeshing, and data redistribution. It is based on a distributed data structure that represents meshes as a set of entities (elements, faces, edges, nodes, etc.), linked by relations (that is, computation dependencies).

PAMPA interfaces with SCOTCH for mesh redistribution, and with MMG3D for parallel remeshing of tetrahedral elements. Other sequential remeshers can be plugged-in, in order to handle other types of elements.

Version 1.0 of PAMPA allows users to declare distributed meshes, to declare values attached to the entities of the meshes (e.g. temperature attached to elements, pressures to the faces, etc.), to exchange values between overlapping entities located at the boundaries of subdomains assigned to different processors, to iterate over the relations of entities (e.g. iterate over the faces of elements), to remesh in parallel the areas of a mesh that need to be remeshed, and to redistribute evenly the remeshed mesh across the processors of the parallel architecture. PAMPA is already used as the data structure manager for the AeroSol solver developed by teams CARDAMOM and CAGIRE.

Version 2.0 of PAMPA features parallel remeshing using any sequential remesher. At the time being, PAMPA is interfaced with the MMG3D tetrahedral remesher designed within team CARDAMOM. Its coupling with Tetgen is in progress. Meshes above one billion elements are generated with a quality similar to that offered by the sequential remesher alone. More than 1 billion of elements are obtained on a cluster with 600 processors in less than 20 minutes. Tests show a quite good weak scalability up to 600 processors, limited by cluster size. Extensive scalability testing will be performed during year 2016. An collaboration with Dassault Aviation demonstrated the use of PAMPA with their meshes during solver computations.

- Participants: Cédric Lachat, François Pellegrini, Cécile Dobrzynski.
- Contact: Cédric Lachat, François Pellegrini
- URL: project.inria.fr/pampa

5.4. Scotch

KEYWORDS: parallel graph partitioning, parallel static mapping, parallel sparse matrix block ordering, graph repartitioning, fixed vertices, mesh partitioning.

FUNCTIONAL DESCRIPTION

SCOTCH is a software package for parallel and sequential sparse matrix ordering, parallel and sequential graph partitioning, as well as sequential static mapping and remapping, without and with fixed vertices, and mesh and hypergraph partitioning.

The initial purpose of SCOTCH was to compute high-quality static mappings of valued graphs representing parallel computations, onto target architectures of arbitrary topologies. Taking into account the topology and heterogeneity of the target architecture, in terms of processor speed and link bandwidth, allows SCOTCH to provide partitions that maximize communication locality.

This feature, which was meant for the NUMA machines of the 1980’s, has not been widely used in the past because high performance computers in the 1990’s were UMA architectures, thanks to hardware advances. As today’s high-end architectures are NUMA again, the mapping feature is regaining popularity.

The SCOTCH package consists of two libraries: the sequential SCOTCH library, and the parallel PT-SCOTCH library (for “Parallel Threaded SCOTCH”) that operates according to the distributed memory paradigm, using MPI. SCOTCH was the first full 64-bit implementation of a general purpose graph partitioner.

Version 6.0, released on December 2012, corresponding to the 20th anniversary of SCOTCH, offers many new features: static mapping with fixed vertices, static remapping, and static remapping with fixed vertices. Several critical algorithms of the formerly strictly sequential SCOTCH library can now run in a multi-threaded way. All of these features, which exist only in the sequential version, will be ported to the parallel PT-SCOTCH library in the upcoming 6.1 major release.

A recent work on branch 6.0 enables SCOTCH to map onto (possibly disconnected) parts of a regular architecture, thanks to the new sub target architecture. This feature aims at allowing programs to distribute their data so as to maximize locality, according to the assignment of nodes performed by the batch scheduler, which do not always represent a regular, nor a connected, part of a parallel machine.

SCOTCH has been integrated into numerous third-party software, which indirectly contribute to its diffusion.

- Participants: François Pellegrini, Cédric Lachat, Astrid Casadei.
- Contact: François Pellegrini
- URL: <https://gforge.inria.fr/projects/scotch/>

5.5. TreeMatch

KEYWORDS: High-performance computing. Process Placement

FUNCTIONAL DESCRIPTION

TreeMatch is a library for performing process placement based on the topology of the machine and the communication pattern of the application. TreeMatch provides a permutation of the processes to the processors/cores in order to minimize the communication cost of the application. Important features are : the number of processors can be greater than the number of applications processes ; it assumes that the topology is a tree and does not require valuation of the topology (e.g. communication speeds) ; it implements different placement algorithms that are switched according to the input size. Some core algorithms are parallel to speed-up the execution. TreeMatch is integrated into various software such as the Charm++ programming environment as well as in both major open-source MPI implementations: Open MPI.

- Participants: Emmanuel Jeannot, Guillaume Mercier, François Tessier.
- Contact: Emmanuel Jeannot
- URL: <http://treematch.gforge.inria.fr>

5.6. Platforms

5.6.1. Platform: The Daltons

The Daltons are a set of machines available for the team members and the STORM team members enabling fast prototyping and benchmarking of our solutions, algorithms and software. It features up-to-date hardware (e.g. latest infiniband or GPU card) with the most recent processors.