AGENCE NATIONALE DE LA RECHERCHE ANR	**Programme BLANC**	Réservé à l'organisme gestionnaire du programme N° de dossier : ANR-08-XXXX-00 Date de révision :
	Document scientifique associé	**Edition 2008**

Acronyme/short title	**FOST**
Titre du projet *(en français)*	Preuves formelles de programmes de calcul scientifique
Titre du projet/Proposal title *(en anglais)*	Formal prOofs of Scientific compuTing programs

Les pages seront numérotées et l'acronyme du projet devra figurer sur toutes les pages du document en pied de page.
Un sommaire du document est bienvenu

S'il s'agit d'un projet déposé dans le cadre d'un accord de coopération internationale*, préciser avec quelle agence étrangère :
☐ National Natural Science Foundation of China (NSFC)
☐ Japan Society for the Promotion of Science (JSPS)
☐ Japanese Science and Technology Agency (JST)
☐ National Science Council of Taiwan (NSC)

**** Veuillez vous reporter aux modalités de soumission particulières pour chaque agence sur le site de l'ANR.***

1. *Programme scientifique / Technical and scientific description of the proposal*

1.1 Problème posé/Rationale

FOST aims at developing and applying methods to formally prove the soundness of programs used in numerical analysis. In particular, we are interested in programs which often appear in the resolution of critical problems and in increasing their safety level. Many critical programs come from numerical analysis, but few people have ever tried to apply formal methods to this kind of programs.
One reason is that formal methods were too immature to handle such problems. Formal method tools and in particular formal proof systems are becoming mature and are now able to deal with the real numbers and with floating-point numbers, which makes it possible to apply these systems to numerical analysis programs.
Another main reason is that the two communities are apart and seldom talk one to the other. The numerical analysis community is used to prove with pen-and-paper that the method error is bounded. The floating-point error is usually discarded as it is very different from what they are used to bound.

They either do not know about or are afraid of formal methods, which are considered too far away from their usage. The formal method community is now able to handle real numbers and mathematical properties. Proving these properties is often tedious and hardly rewarding. Therefore, numerical analysis is discarded. As there is no library of mathematical formally proved facts, there is no program verification about numerical analysis.

There are two main issues in order to prove the correctness of a numerical analysis program:

- the method error must be bounded. This means that the difference between what is computed and what should be computed must be under control. For example, infinite series are truncated, continuous spaces are discretized, and so on.
- the floating-point error must be bounded. This means that the difference between what the program would compute if all the computations were exact and the effective result with floating-point roundings and particularities.

These two issues are linked: for example, the floating-point error depends on the value of the variables that are given by their theoretical value.

FOST aims at developing new methods to bound the global error of a numerical program. These methods will be very generic in order to prove a large range of numerical analysis programs.

Moreover, FOST aims at providing reusable methods that are understandable by non-specialists of formal methods.

FOST is the association of several members with varied and appropriate competences (computing scientists, experts in proof assistant design, in proofs of programs, in real and floating-point numbers) and with a significant experience in their respective domains.

1.2 *Contexte et enjeux du projet/Background, objective, issues and hypothesis*

State of the art:

Since the 90s, computer systems are handling more and more of our day-to-day life (transports, banking, medicine, meteorology, and so on). Programs are taking decisions for us and the quality of these programs have been raised. A certification, or at least a high level of guarantee, are needed for critical systems. In the transport field in France, the Meteor subway has been guaranteed using a formal method called the B method [Abrial]. As for numerical problem, some disasters such as the explosion of the first flight of Ariane 5 or the Pentium bug, have shown that this is a particularly critical topic.

Another example of critical numerical program is the management of nuclear power plants which computes eigenvalues that should be smaller than 1.

Formal methods have been applied to floating-point arithmetic by various industrial or academic groups worldwide. The goal is to give certificate of conformance to specifications or correctness of applications, even when computer arithmetic is used.

In France, the teams Arénaire (LIP/France), DALI (ELIAUS/France) and Fluctuat (CEA-Saclay/France) are interested in computer arithmetic [Brise-Muller ,Fluctuat] and in formal methods applied to floating-point arithmetic [DRT01,Bol04] using the proof assistant Coq [Coq06].

Worldwide, the interaction formal proof/computer arithmetic concerns for example the National

Institute of Aerospace (NIA) linked to the NASA Langley (USA) [Munoz-Dowek-Car], AMD [Russinoff] and INTEL [Harrison].

As for numerical programs, the method error is bounded in many articles, but we do not know of any work that formally proves this kind of bound as the mathematics involved are hardly formalized.

Issues:

The issue here is to study numerical analysis programs. Their particularity is that the method error is usually well-known and proved in pen-and-paper proofs. Getting a formal proof of correctness requires several steps:

- many mathematical properties have to be formalized and proved, e.g. Fourier transform. These properties are well-known, but their formalization may be long and tedious, while hardly rewarding.
- the proof of the mathematical error relies on the previous libraries. It may reveal gaps in the proof and will probably have to go into the very depths of the pen-and-paper proof.
- the floating-point proof is unknown: this aspect is usually discarded. Nevertheless, for this aspect, there is already a formal library of known facts about floating-point arithmetic that FOST can reuse.

This technique will first be applied to one given program. It will then be generalized in order to be applicable to a large category of numerical programs. The last point is the diffusion of our results: it is important to provide methods that can be used by non-specialists.

[Abrial] J.-R. Abrial ;The B-book : Assigning programms to meanings ; Cambridge University Press ; 1996.

[Brise-Muller] N. Brisebarre and J.-M. Muller; Correct Rounding of Algebraic Functions; Theoretical Informatics and Applications; Vol. 41, pages 71-83; jan-march 2007.

[Coq06] The Coq Development Team ; The Coq Proof Assistant Reference Manual Version~8.1. INRIA.

[DRT01] Marc Daumas, Laurence Rideau, and Laurent Théry. A generic library of floating-point numbers and its application to exact computing. In Richard J. Boulton and Paul B. Jackson, editors. *Theorem Proving in Higher Order Logics: 14th International Conference, TPHOLs 2001*, volume 2152 of *LNCS*. Springer-Verlag, pages 169-184. 2001.

[Fluctuat] Eric Goubault and Sylvie Putot; Static Analysis of Numerical Algorithms; Proceedings of Static Analysis Symposium SAS'06; Seoul, LNCS volume 4134, pp. 18-34; 2006.

[Harrison] John Harrison; Floating-Point Verification using Theorem Proving; Proceedings of SFM 2006, the 6th International School on Formal Methods for the Design of Computer, Communication, and Software Systems. Springer LNCS 2965, pp. 211-242, 2006.

[Munoz-Dowek-Car] César Muñoz, Gilles Dowek, and Victor Carreño; Formal Analysis of the Operational Concept for the Small Aircraft Transportation System; Rigorous Engineering of Fault-Tolerant Systems; 2006.

[Russinoff] David Russinoff; A Mathematical Approach to RTL Verification; Computer Aided Verification; Berlin, LNCS volume 4590; 2007.

1.3 *Objectifs et caractère ambitieux/novateur du projet/Specific aims, highlight of the originality and novelty of the project*

The originality of FOST mainly lies in the meeting of two separated communities:

- computing scientists use computers to provide an answer and bound their error with pen-and-paper proofs. They do not go into formal proof as it is considered too complex. They usually discard floating-point errors.
- Formal methods provide the correctness of systems. Formal proofs may concern mathematical properties [MMPHD01] or correctness of programs. They do not go into numerical analysis as it is considered too complex and would require a large library of mathematical facts. This library would be long to develop and poorly rewarded.

The novelty in this approach is that we put together people from different fields in order to provide a high level of guarantee of real-life programs. FOST therefore needs people from different fields to work together.

The scientific locks we plan to unfasten are:

- study a real program computing the gradient. This operator is well-known and often used. This is a good case study for our approach.
- study a large range of numerical programs. Some methods and results obtained previously can be applied (or not) to other programs. It is interesting to know which general methods we develop can be applied to a class of numerical applications.
- provide a toolbox for computing scientists. Formal methods are rather obscure for many mathematicians and we need a lot of pedagogy to bring them to formal tools.

FOST will also deal with two very different kinds of bounds: numerical programs are correct if both their method error and floating-point error are bounded. Therefore, we need to develop methods for bounding both errors. The methods will probably be very different and require expertise in two different domains.

- The method error mostly amongst to mathematical demonstrations. We will therefore need to formally prove many mathematical results and go into the very details of the pen-and-paper proofs. The consequence will be a library of mathematical facts.
- The floating-point error requires a deep study of the program. We will look into manual and automatic methods to be able to bound the final error due to roundings at each operation. The fact that these errors may compensate is to be taken into account for the final error to be reasonably small.

This double aspect of mathematical and computer arithmetic requirements is to be composed with the formal methods requirement of very detailed proofs. This is indeed ambitious as many competences are to be mixed in order to reach our goal.

1.4 Description des travaux : programme scientifique/*For each specific aim: a proposed work plan should be described (including preliminary data, work packages and deliverables)*

1. Gradient

   When coefficients in a partial differential equation are unknown, they can be estimated from some measurement of the solution. This parameter estimation problem is usually put in terms of the minimization of a least square function. The adjoint state technique is an efficient tool for computing the analytic gradient of this function, that can then be used as input by an optimization code. The singular value decomposition is a powerful tool for the deterministic sensitivity analysis. It can quantify the number of parameters that can be estimated from the solution measurements. This allows us to choose a parametrization for the sought coefficients, and also to create measurement experiments. These techniques are for example used in seismic in order to probe the interior of Earth with pulse emissions. In this exact case, this consists in minimizing the function $J(p) = ½ ||d – F(p)||^2$ where d stands for the experimental measure and F(p) the theoretic function. In order to compute this minimization, we find all the values such as grad(J(p))=0, where grad is the gradient. The gradient is therefore a typical numerical analysis program we intend to formally prove.

   This task is divided into two sub-tasks:

   - The first sub-task will bound the method error, based on the literature proofs. This requires to be able to formally define this mathematical quantity and to prove various lemmas about it. The fact that the mathematical ideal value and what the computer tries to compute are linked are far from trivial and requires a detailed mathematical that will be formally proved.
   - The second sub-task will bound the floating-point error. There is no literature to help us here as this problem is usually discarded. A naïve bounding of the error may produce a exponential error bound. We will therefore have to apply more subtle methods in order to exhibit the error compensations of this very program.

   The method is the following one. We take a real program written in C for example that computes the gradient. We produce a very detailed pen-and-paper proof of the bounding of the method error (or of its nullity) and of the bounding of the floating-point error. At this point, we may use any mathematical results, and any complicated technique. We then annotate the program with the results of the pen-and-paper proof: for example, each value cannot be further than a given quantity (e.g. 2^(-50)) of the ideal mathematical value. The Why platform then generates proof obligations [FilMar07]. Some of them may be solved by automatic means, but most of them will have to be proved by hand. Depending on the pen-and-paper proof, the formal proof may be long and technical. They may need mathematical definitions which are not yet formalized or mathematical results which are not yet formally proved. When the formal tools have agreed with all the proof obligations, the program is proved correct with respect to its given specification.

2. Generalization

   This task is also subdivided into two sub-tasks as we are interested in bounding both the method error or the floating-point error:

   - From the results of the literature about the method error of numerical schemes and the results of the CerPAN ANR about the formal proof of the classical three-point numerical scheme for the one-dimensional acoustic wave equation, we want to deduce generic methods and results that can be applied to a larger class of problems.

     For example, we will increase the order of the numerical scheme, or increase the space dimension or even consider other equations (elastic waves, Maxwell equations...). We will also address the simpler cases of Ordinary Differential Equations (ODE) and in particular study how our generic results could be applied to the classical 4th-order Runge-Kutta method (RK4). FOST will show the limits of CerPAN's approach and what genericity can be obtained from the already developed methods.

     Another generalization in order to bound the method error is that, if the dimensions are independent, the same results could be applied several times. This would be very useful but requires some technicality in the formal proofs.

   - The generalization on the floating-point error is based on the analytical error. This method is a full determination of the floating-point error, with its signed value. We can then take full advantage of this in order to subtract the errors and therefore exhibit the error compensations. the drawback is that we have to explicit an exact analytical expression of the error (and not an overestimation) and this expression does not have to be simple. FOST will determine if this technique can be applied to a large class of programs and under which conditions. Moreover, it will try to help to find out the analytical expression (automatically or not). The existence of a simple analytical expression is also under question: the convergence of the numeric scheme may be a sufficient condition for a convenient expression of the analytical error.

3. Toolbox

   This task represents several ways to make formal methods accessible to non-specialists. The computing scientists are indeed not familiar with formal tools, and some pedagogy is needed to help them towards formal guarantees.

   This toolbox can be thought as a comprehensive manual. It contains some documentation for annotating programs, methods to try to prove them and several real-life examples that are fully proved and explained step by step. We do not hope to convert computing scientists to formal methods. A first step would be a simple specification of their program that would bring more coherence. A part of the annotations could be proved using the Gappa tool [Mel06] that produces a formal proof by using interval arithmetic methods. Nevertheless, as soon as loops are involved, Gappa is usually not powerful enough to give interesting results. Nevertheless, partial formal proofs could be done automatically (such as array access or floating-point overflow).

Some tools can also be designed to evaluate the floating-point error. For example, if we compute with specific values, both with exact computations and with floating-point computations, we can get the floating-point error. This is some kind of program testing concerning floating-point arithmetic. It does not provide any proof, but it can help the user to figure out reasonable floating-point errors.
Formal proofs should also be simplified by the creation of ad hoc tactics, designed for our kind of goals and that handle real numbers or floating-point number easily and automatically solve typical proof obligations. This means adapting the formal tools especially Coq to make it more intelligent about floating-point and method errors.

[Mel06] G. Melquiond ; De l'arithmétique d'intervalles à la certification de programmes. Thèse de doctorat, Ecole Normale Supérieure de Lyon; 2006.

1.5 Résultats escomptés et retombées attendues/*Expected results and potential impact*

The expected results of FOST are:

- a gallery of formally proved numerical analysis programs;
- a library of mathematical facts to base upon in order to bound the method error;
- several generic methods and tools for formally proving numerical analysis programs;
  - for the method error;
  - for the floating-point error;
- a toolbox for computing scientists to add confidence in their programs.

1.6 Organisation du projet/*Project flow*

Tâche/Tasks	Partenaire s/Partners			Année 1 Year 1		Année 2 Year 2		Année 3 Year 3	
	1	2	3	6	12	18	24	30	36
Task 1.1 Gradient – method error, P3		▨	▨	▨	▨	▨	▨		
Task 1.2 Gradient – floating-point error, P1	▨	▨		▨	▨	▨	▨		
Task 2.1 Generalization – method error, P3		▨	▨	▨	▨	▨	▨	▨	
Task 2.2 Generalization – floating-point error, P1	▨	▨		▨	▨	▨	▨	▨	
Task 3 Toolbox, P2	▨	▨	▨				▨	▨	▨
Rapports d'avancement semestriel Progress report/expenses				☺	☺	☺	☺	☺	☺
Rapport final /Final report									☆

☺ : Rapport d'avancement semestriel/6 month-progress report
☆: Rapport de synthèse et récapitulatif des dépenses/Final report and expenses summary

TABLEAU des LIVRABLES et des JALONS (le cas échéant)/*Deliverables and milestones*			
Tâche Task	Intitulé et nature des livrables et des jalons/ *Title and substance of the deliverables and milestones*	Date de fourniture ***nombre de mois à compter de T0*** ***Delivery date, in months starting from T0***	Partenaire responsable du livrable/jalon *Partner in charge of the deliverable/ milestone*
1.			
	Technical report	18	1
	Formal proof	24	1
	Publications	24	1
2.			
	Simplest formal proofs	12	1
	More complex formal proofs	24	1
	Publications	30	1
3.			
	Manual	36	2
	Toolbox	36	2

1.7 Organisation du partenariat/*Consortium organisation*

1.7.1 Pertinence des partenaires/*Consortium relevance*

– Partner 1 is the developer of the Why platform. This tool will be used in order to prove the numerical C programs as they are written. This expertise in program certification is needed for specific proofs such as proofs of valid pointer access, valid function calls, and so on. Partner 1 has also a large expertise in floating-point arithmetic that is needed for the rounding error bounds. Lastly, partner 1 has also developed an annotation language for floating-point properties that will be used in this work.

[BolFil07] Sylvie Boldo and Jean-Christophe Filliâtre. **Formal Verification of Floating-Point Programs**. In Peter Kornerup and Jean-Michel Muller, editors, *Proceedings of the 18th IEEE Symposium on Computer Arithmetic*, pages 187-194, Montpellier, France, June 2007.

[FilMar07] Jean-Christophe Filliâtre and Claude Marché. **The Why/Krakatoa/Caduceus platform for deductive program verification**. In Werner Damm and Holger Hermanns, editors, *19th International Conference on Computer Aided Verification*, Lecture Notes in Computer Science, Berlin, Germany, July 2007. Springer-Verlag.

[Bol04] Sylvie Boldo. **Preuves formelles en arithmétiques à virgule flottante.** PhD thesis, École Normale Supérieure de Lyon, November 2004.

– Partner 2 has expertise in the field of scientific computing, and more precisely in inverse problems and deterministic sensitivity analysis with respect to distributed coefficients in Partial Derivative Equations (PDE) or Ordinary Differential Equations (ODE). For instance, his work on seismic waveform inversion was based on the 2D acoustic wave equation. One of the key issues is the implementation of the exact derivative of the forward model to study. More recently, Partner 2 developed interest in the safety of programming for scientific computing applications, and in particular in the use of functional programming and formal proof tools.

[BACW08] H. Ben Ameur, F. Clément and P. Weis, **The Multi-Dimensional Refinement Indicators Algorithm for Optimal Parameterization.** *Journal of Inverse and Ill-Posed Problems*, **16**, pp. 107-126. 2008.

[C+07] A. Cartalade, P. Montarnal, M. Filippi, C. Mugler, M. Lamoureux, J.-M. Martinez, F. Clément, Y. Wileveau, D. Coelho and E. Tevissen, **Application of inverse modeling methods to thermal and diffusion experiments at Mont Terri Rock laboratory**, *Physics and Chemistry of the Earth, Parts A/B/C*, **32**, pp. 491-506. 2007.

[C+06] F. Clément, V. Martin, A. Vodicka, R. Di Cosmo and P. Weis, **Domain Decomposition and Skeleton Programming with OCamlP3l**, *Parallel Computing*, **32**, pp. 539-550. 2006.

– Partner 3 is working on interaction between formal proofs and numerical analysis for several years. In particular, partner 3 is the instigator of the CerPAN project. Partner 3 has expertise in the field of real numbers (development of real numbers standard library of Coq) as well as formalization of numerical problems into theorem prover. She has also expertise to develop automatic decision procedures for real numbers. All of this expertises are necessary for this work.

[DDMMCalcul06] David Delahaye and Micaela Mayero. **Quantifier Elimination over Algebraically Closed Fields in a Proof Assistant using a Computer Algebra System.** In *Proceedings of Calculemus 2005*, volume 151(1) of ENTCS, pages 57-73, 2006.

[MMTPHOL02] Micaela Mayero. **Using Theorem Proving for Numerical Analysis. Correctness Proof of al Automatic Differentiation Algorithm.** In *Proceedings of TPHOLs2002*, volume 2410, page 246. Springer-Verlag LNCS, 2002.

[MMPHD01] Micaela Mayero. **Formalisation et automatisation de preuves en analyses réelle et numérique.** PhD thesis, Université Paris VI, décembre 2001.

1.7.2 Complémentarité et synergie des partenaires/*Added value of the consortium*

The three partners represent various fields that are all needed to fulfil the objectives. The complementarity is very high as the partners are experts in numerical analysis (P2), formal methods (P3), program verification and floating-point arithmetic (P1) as well.

Moreover, the represented fields are not used to work together. Floating-point arithmetic and formal certification have a quite recent history while numerical analysis and formal certification are miles away. To show that complementarity is a key fact for this project, we apply FOST to two different CSD (Comité scientifique disciplinaire – disciplinary scientific committee).

1.7.3 Qualification du coordinateur du projet et des partenaires/*Principal investigator and partners : résumé and CV*

See at the end of this document the CVs of Sylvie Boldo and Micaela Mayero and the implications of the partners in other proposals.

1.8 Access to large facilities

None.

1.9 Stratégie de valorisation et de protection des résultats/*Data management, data sharing, intellectual property strategy, and exploitation of project results*

The data involved are:

- numerical programs
- formal proofs
- a toolbox
- publications

  We intend to make all of them available. In particular, the annotated programs that will be formally proved correct will be made available to the community with their error bounds.

FICHES BUDGÉTAIRES - Blanc

Fiche Partenaire 1

Nom Complet du partenaire	Catégorie de partenaire	Base de calcul pour l'assiette de l'aide
INRIA / Centre de Recherche Saclay - Île-de-France	Organismes de recherche+Fondation de recherche	Coût marginal

Données financières (montant HT en € incluant la TVA non récupérable) 1

EQUIPEMENTS (€)	Personnels						Prestations de service externe (€)	Missions (€)	Autres dépenses (€)	Dépenses justifiées sur facturation interne (€)	Totaux (€)
	permanents		non permanents à financer par l'ANR		Autres non permanents						
	personne. mois	Coût (€)	*personne. mois*	Coût (€)	*personne. mois*	Coût (€)					
	25.20	105,696	*6.00*	8,000				21,000	21,000		155,696
Montant maximum des frais de gestion/ frais de structure (€)				2,000			<--Frais de gestion / frais de structure demandés (€)-->				2,000
Uniquement pour laboratoire d'organisme public de recherche ou fondation financé au coût marginal, indiquer le taux d'environnement								100.0%	Frais d'environnement (€)		113,696
										Coût complet (€)	**271,392**
										Coût éligible pour le calcul de l'aide : Assiette (€)	**50,000**
100.00%							Taux d'aide demandée)-->	100.0%		***Aide demandée (€)***	***50,000***

Fiche Partenaire 2

Nom Complet du partenaire	Catégorie de partenaire	Base de calcul pour l'assiette de l'aide
INRIA / Paris- Rocquencourt	Organismes de recherche+Fondation de recherche	Coût marginal

Données financières (montant HT en € incluant la TVA non récupérable) 1

EQUIPEMENTS (€)	Personnels						Prestations de service externe (€)	Missions (€)	Autres dépenses (€)	Dépenses justifiées sur facturation interne (€)	Totaux (€)
	permanents		non permanents à financer par l'ANR		Autres non permanents						
	personne. mois	Coût (€)	*personne. mois*	Coût (€)	*personne. mois*	Coût (€)					
	5.40	29,435	*3.00*	4,000				6,000	12,000		51,435
Montant maximum des frais de gestion/ frais de structure (€)				880			<--Frais de gestion / frais de structure demandés (€)-->				880
Uniquement pour laboratoire d'organisme public de recherche ou fondation financé au coût marginal, indiquer le taux d'environnement								100.0%	Frais d'environnement (€)		33,435
										Coût complet (€)	**85,750**
										Coût éligible pour le calcul de l'aide : Assiette (€)	**22,000**
100.00%							Taux d'aide demandée)-->	100.0%		***Aide demandée (€)***	***22,000***

Fiche Partenaire 3

Nom Complet du partenaire	Catégorie de partenaire	Base de calcul pour l'assiette de l'aide
Laboratoire d'Informatique de Paris Nord	Organismes de recherche+Fondation de recherche	Coût marginal

Données financières (montant HT en € incluant la TVA non récupérable) 1

EQUIPEMENTS (€)	Personnels						Prestations de service externe (€)	Missions (€)	Autres dépenses (€)	Dépenses justifiées sur facturation interne (€)	Totaux (€)
	permanents		non permanents à financer par l'ANR		Autres non permanents						
	personne. mois	Coût (€)	*personne. mois*	Coût (€)	*personne. mois*	Coût (€)					
	14.40	29,003	*6.00*	8,000				14,000	18,000		69,003
Montant maximum des frais de gestion/ frais de structure (€)				1,600			<--Frais de gestion / frais de structure demandés (€)-->				1,600
Uniquement pour laboratoire d'organisme public de recherche ou fondation financé au coût marginal, indiquer le taux d'environnement								80.0%	Frais d'environnement (€)		29,602
										Coût complet (€)	**100,205**
										Coût éligible pour le calcul de l'aide : Assiette (€)	**40,000**
100.00%							Taux d'aide demandée)-->	100.0%		***Aide demandée (€)***	***40,000***

Récapitulatif des données financières

	EQUIPEMENTS (€)	Personnels						Prestations de service externe (€)	Missions (€)	Autres dépenses (€)	Dépenses justifiées sur facturation interne (€)	Totaux (€)
		permanents		non permanents à financer par l'ANR		Autres non permanents						
		personne. mois	Coût (€)	*personne. mois*	Coût (€)	*personne. mois*	Coût (€)					
Partenaire1	-	25	105,696	6	8,000	-	-	-	21,000	21,000	-	155,696
Partenaire2	-	5	29,435	3	4,000	-	-	-	6,000	12,000	-	51,435
Partenaire3	-	14	29,003	6	8,000	-	-	-	14,000	18,000	-	69,003
Partenaire4	-	-	-	-	-	-	-	-	-	-	-	-
Partenaire5	-	-	-	-	-	-	-	-	-	-	-	-
Partenaire6	-	-	-	-	-	-	-	-	-	-	-	-
Partenaire7	-	-	-	-	-	-	-	-	-	-	-	-
	-	**45.00**	**164,134**	***15.00***	**20,000**	**-**	**-**	**-**	**41,000**	**51,000**	**-**	**276,134**
											Frais de gestion / frais de structure demandés (€)-->	4,480
											Frais d'environnement (€)	176,733
											Coût complet (€)	**457,347**
											Coût éligible pour le calcul de l'aide : Assiette (€)	**112,000**
											Aide demandée (€)	***112,000***

2 Justification scientifique des moyens demandés/*Requested budget : detailed financial plan*

The equipment budget represents the purchase of desktop and laptop computers, books and softwares. Each of them costs less than 4000 euros HT. Formal methods and formal proof checking require up-to-date material in terms of speed and memory.

The manpower represents trainees allowances. The trainee might be “licence” (L3), “master 2 recherche” or “master 2 pro”.

Travels are a large part of our need. This represents our participation to international conferences (such as ARITH), short-terms visits or invitations for collaborations both national and international with teams working in topics linked to the FOST project.

Annexes

Description des partenaires/*Partners informations* (cf. § 1.7.1)

The project-team ProVal, from INRIA Saclay – Île-de-France, provides methods and tools that can be integrated into the software development cycle and that make it possible to produce code that is proven to be correct with respect to its expected behaviour. When sophisticated tools are used for analyzing safety-critical code, their reliability is an important question: in an industrial setting, there is often a certification process. This certification is based on an informal satisfaction of development rules. The team believes decision procedures, compilers or verification condition generators should not act as black boxes but should be themselves specified and proved or should produce evidence of the correctness of their output. This choice is influential in the design of our tools and is also a good challenge for its own tools. The team develops a generic program proof environment (the Why platform), that is able to generate proof demands that can be delegated to automatic or interactive provers. Dedicated environments to prove C programs (Caduceus) and Java programs (Krakatoa) annotated with formulas describing the expected behaviour, are constructed on top of this tool. Jean-Christophe Filliâtre is the main developer of the Why tool. Sylvie Boldo is specialist in floating-point arithmetic.

The project-team Estime, from INRIA Paris – Rocquencourt, concerns on the numeric modelization of heterogeneous media. All these problems are governed by complex physics phenomena and appropriate techniques must be used to model them numerically and to optimize them. The aim of the Estime project is to design such techniques that are efficient and accurate. A first example of such a medium is the subsurface. On one hand, we construct images of its structure through seismic inversion. On the other hand, we build numerical models of various flow in porous media: contaminant transport for environment studies, or oil displacement in petroleum engineering. The core of a nuclear power plant is another example of a heterogeneous medium. In this case, we study its neutron properties. François Clément is mostly interested in the inverse problems and the deterministic sensibility analysis, and also in the flow in porous media for the deep underground waste disposal and the dynamic of underground waters.

The LRC team, from LIPN, focuses on three main topics: linear logic and various applications to computer science; systems specification and assisted modelisation with applications to dynamic and distributed systems, as well as to databases; algebraic combinatorics and conception of tools for the design and analysis of algorithms in mathematical physics. Micaela Mayero mostly cares about the second topic, and the interactions with the first one. Her competences concern problems specifications in various domains such as numerical analysis, number theory, distributed systems (such as Petri nets). This requires a large knowledge on formal methods (she developed a formal library about real numbers) and on their automations.

Biographies/***Résumés and CV*** (cf. § 1.7.3) *(1 page maximum par personne)*

Sylvie Boldo (coordinator)

Ms Sylvie Boldo is 29 years old and is permanent junior researcher (CR2) at the INRIA Saclay – Île de France in the project-team ProVal.

Education/positions:

1996-1998 “Classes préparatoires“ at the Lycée Louis-le-Grand
1998-2002 Student/Civil servant at the “École Normale Supérieure de Lyon“
2002-2005 PhD student (“Allocataire Couplée“) at the “École Normale Supérieure de Lyon“
under the supervision of Marc Daumas about “Preuves formelles en arithmétiques à virgule flottante”: Formal Proofs about Floating-Point Arithmetics.
2005 Visit at the NIA (National Institute of Aerospace) in Hampton, VA

5 publications:

[BolDauLi08] Sylvie Boldo, Marc Daumas, and Ren-Cang Li. **Formally Verified Argument Reduction with a Fused-Multiply-Add**. *IEEE Transactions on Computers*, 2008. Minor revision.

[BolMel08] Sylvie Boldo and Guillaume Melquiond. **Emulation of FMA and correctly-rounded sums: proved algorithms using rounding to odd**. *IEEE Transactions on Computers*, April 2008.

[BolFil07] Sylvie Boldo and Jean-Christophe Filliâtre. **Formal Verification of Floating-Point Programs.** In Peter Kornerup and Jean-Michel Muller, editors, *Proceedings of the 18th IEEE Symposium on Computer Arithmetic*, pages 187-194, Montpellier, France, June 2007.

[Bol06a] Sylvie Boldo. **Pitfalls of a Full Floating-Point Proof: Example on the Formal Proof of the Veltkamp/Dekker Algorithms**. In *Proceedings of the third International Joint Conference on Automated Reasoning (IJCAR)*, pages 52-66, Seattle, USA, August 2006.

[BolMul05] Sylvie Boldo and Jean-Michel Muller. **Some Functions Computable with a Fused-mac**. In Paolo Montuschi and Eric Schwarz, editors, *Proceedings of the 17th Symposium on Computer Arithmetic*, pages 52-58, Cape Cod, USA, 2005.

Former coordinator experience: none.

Micaela Mayero

Ms Micaela Mayero (36 years old) is associate professor ("Maître de Conférences") at the University of Paris 13.

Education/positions:
Mathematics: Licence, Maîtrise, DEA of numerical analysis (1992-1994)
Computer science: Licence, Maîtrise, DEA of "Sémantique Preuves et Programmation" (1995-1997)

PhD thesis (97-2001): "Formalisation et automatisation de preuves en analyses réelle et numérique" at the University of Paris 6 under the supervision of Gilles Dowek.

Postdoc (november 2001 - september 2002) at Chalmers University -Göteborg, Sweden.

Temporary position ("ATER") at the University of Paris 7 (october 2002 - august 2003).

Engineer-researcher at CEA-Saclay (Laboratoire de Sûreté des Logiciels (LSL)) (november 2003 - june 2004).

5 publications:

[DelMay06] David Delahaye and Micaela Mayero. **Quantifier Elimination over Algebraically Closed Fields in a Proof Assistant using a Computer Algebra System.** In *Proceedings of Calculemus 2005*, volume 151(1) of ENTCS, pages 57-73, 2006.

[DelMay05] David Delahaye and Micaela Mayero. **Dealing with Algebraic Expressions over a Field in Coq using Maple.** In *Journal of Symbolic Computation: special issue on the integration of automated reasoning and computer algebra systems*, 39:569-592, 2005.

[May02] Micaela Mayero. **Using Theorem Proving for Numerical Analysis. Correctness Proof of al Automatic Differentiation Algorithm.** In *Proceedings of TPHOLs2002*, volume 2410, page 246. Springer-Verlag LNCS, 2002.

[DelMay01] David Delahaye and Micaela Mayero. **Field: une procédure de décision pour les nombres réels en Coq.** In *JFLA 2001*, Pontarlier. INRIA, Janvier 2001.

[May00] Micaela Mayero. **The Three Gap Theorem (steinhauss conjecture).** In *Proceedings of TYPES'99*, volume 1956, pages 162-173. Springer-Verlag LNCS, 2000.

Implication des personnes dans d'autres contrats/*Partner's involvement in other projects* (cf. § 1.7.3)

All the participants were participants of the ANR CerPAN (ANR-05-BLAN-0281-04). FOST uses the results of CerPAN as CerPAN has shown that numerical programs can be formally proved, but also that this requires some effort. It has shown where the difficulties are. CerPAN also created the annotation language that will be used for the assertions.

Partenaire	Nom de la personne participant au projet	Personne. Mois	Intitulé de l'appel à projets Source de financement Montant attribué	Titre du projet	Nom* du coordinateur	Date début - Date fin
Partner	Name of the person involved in the project	Man.month	Name call for proposals Other fundings from different organisms Allocated budgets	Proposal title	Name Principal Inverstigator	Start- End of the project
N°1	Sylvie Boldo	18	ANR Blanc	CerPAN	Micaela Mayero	12/05 12/08
N°1	Jean-Christophe Filliâtre	7.2	ANR Blanc	CerPAN	Micaela Mayero	12/05 12/08
N°1	Jean-Christophe Filliâtre	7.2	RNTL	CAT	Benjamin Monate	06/06 06/09
N°2	François Clément	3.6	ANR Blanc	CerPAN	Micaela Mayero	12/05 12/08
N°3	Micaela Mayero	18	ANR Blanc	CerPAN	Micaela Mayero	12/05 12/08

Demandes de contrats en cours d'évaluation/Other proposals under evaluation

COMPAR is orthogonal to the FOST proposal. COMPAR aims at automatically generating programs that are accurate, fast and certified. This requires a very high level of automatization as the formal proofs must be generated at the same time as the program. The numerical analysis programs we care about in FOST are too complicated for such an approach. Moreover, COMPAR do not handle the method error at all.

Partenaire	Nom de la personne participant au projet	Personne. Mois	Intitulé de l'appel à projets Source de financement Montant demandé	Titre du projet	Nom* du coordinateur
Partner	Name of the person involved in the project	Man.month	Name call for proposals Other fundings from different organisms Expected grants	Proposal title	Name Principal Inverstigator
N°1	Sylvie Boldo	7.2	ERC Advanced Investigator Grant	COMPAR Computer Arithmetic: towards accurate, fast and certified numerical programs	Jean-Michel Muller
N°3	Micaela Mayero	6	ANR Blanc	ComplICE: Complexité Implicite, Concurrence et Extraction	Patrick Baillot
N°3	Micaela Mayero	15	ANR JC	CoVAMP: Combiner Vérification Automatique de Modèles et Preuves de théorèmes	Kais Klai