RAIM 2016 : Programme

Mardi 28 juin

Session Arithmétique Flottante - Mioara Joldes (LAAS, CNRS, Toulouse)

- **15h00.** **Claude-Pierre Jeannerod (Aric Team, LIP, ENS-Lyon) :** *Analyse d'algorithmes en arithmétique à virgule flottante.*

  Résumé : L'arithmétique à virgule flottante est le moyen le plus couramment utilisé pour calculer avec des (approximations des) réels sur ordinateur. Si cette arithmétique est par nature inexacte, la norme IEEE qui la régit définit un cadre strict dans lequel il est possible d'analyser et prédire de façon tout à fait rigoureuse le comportement de nombreux algorithmes numériques de base (sommes, produits scalaires, arithmétique complexe, ...). Dans ce cours, nous illustrerons ce principe à l'aide de résultats obtenus très récemment, qui revisitent une partie de l'analyse numérique classique.

- **16h00.** **Valentina Popescu (Aric Team, LIP, ENS-Lyon) :** *A new multiplication algorithm for extended precision using floating-point expansions*

  Résumé : Some important computational problems must use a floating-point (FP) precision several times higher than the hardware implemented available one. These computations critically rely on software libraries for high-precision FP arithmetic. The representation of a high-precision data type crucially influences the corresponding arithmetic algorithms. Recent work showed that algorithms for FP expansions, that is, a representation based on unevaluated sum of standard FP types, benefit from various high-performance support for native FP, such as low latency, high throughput, vectorization, threading, etc. Bailey's QD library and its corresponding Graphics Processing Unit (GPU) version, GQD, are such examples. Despite using native FP arithmetic as the key operations, QD and GQD algorithms are focused on double-double or quad-double representations and do not generalize efficiently or naturally to a flexible number of components in the FP expansion. We introduce here a new multiplication algorithm for FP expansion with flexible precision, up to the order of tens of FP elements in mind. The main feature consists in the partial products being accumulated in a special designed data structure that has the regularity of a fixed-point representation while allowing the computation to be naturally carried out using native FP types. This allows us to easily avoid unnecessary computation and to present rigorous accuracy analysis transparently.

- **16h30.** Pause.

- **17h00.** **Gregoire Lecerf (LIX, Ecole Polytechnique, Palaiseau) :** *Evaluating Straight-Line Programs over Balls.*

  Co-auteur : Joris van der Hoeven.

  Résumé : Interval arithmetic achieves numerical reliability for a wide range of applications, at the price of a performance penalty. For applications to homotopy continuation, one key ingredient is the efficient and reliable evaluation of complex polynomials represented by straight-line programs. This is best achieved using ball arithmetic, a variant of interval arithmetic. In this talk, we describe strategies for reducing the performance penalty of basic operations on balls. We also show how to bound the effect of rounding errors at the global level of evaluating a straight-line program. This allows us to introduce a new and faster "transient" variant of ball arithmetic. Most of our strategies have been implemented inside the Multimix and Justinline libraries of Mathemagix, in C++ and in the Mathemagix language. We have also implemented a dedicated "just in time" compiler for straight-line programs from scratch, supporting SSE and AVX technologies. Overall, we managed to reduce the overhead of ball arithmetic to a small factor between two and four.

- **17h30.** **Antoine Plet (Aric Team, LIP, ENS-Lyon) :** *Développement d'une bibliothèque pour une arithmétique à virgule flottante symbolique.*

  Co-auteurs : Jean-Michel Muller , Claude-Pierre Jeannerod , Nicolas Louvet .

  Résumé : Pour analyser a priori la précision d'un algorithme en arithmétique à virgule flottante, on calcule en général une borne d'erreur uniforme sur la sortie, valide pour la plupart des entrées et paramétrée par la précision p. Pour montrer par la suite que cette borne est précise, une méthode consiste à produire un exemple d'entrée pour laquelle l'erreur commise par l'algorithme approche cette borne, ou même l'atteint. De tels exemples peuvent être donnés sous la forme de nombres à virgule flottante dans un des formats standards IEEE (par exemple, pour p=53), ou bien, sous la forme d'expressions paramétrées par p, qui peuvent être vues comme des nombres à virgule flottante symboliques. Avec de tels exemples d'entrée, un résultat de finesse d'une borne peut être établi pour potentiellement tous les formats raisonnables au lieu d'un seul. Cependant, cela requiert la capacité d'exécuter l'algorithme sur ces données, en particulier, de calculer l'arrondi correct pour la somme, le produit ou la division de deux nombres à virgule flottante symboliques. Au cours de cet exposé, nous verrons comment ces opérations arithmétiques de bas niveau peuvent être exécutées automatiquement. Nous introduisons un modèle pour l'arithmétique à virgule flottante symbolique, accompagné d'algorithmes pour calculer l'arrondi au plus proche d'une somme, d'un produit, de l'opération fma (fused multiply-add) et dans certains cas d'un quotient. Une implantation en Maple sera aussi décrite, et illustrée sur des exemples de la littérature.

- **18h00.** **Vincent Lefèvre (Aric Team, LIP, ENS-Lyon) :** *Correctly Rounded Arbitrary-Precision Floating-Point Summation.*

  Résumé : Je vais présenter un algorithme rapide de sommation en virgule flottante à précision arbitraire. L'arithmétique est celle de GNU MPFR: base 2; pas de dénormalisés; chaque variable (ici, chaque entrée et la sortie) a sa propre précision.

Mercredi 29 juin

Session Calcul Formel - Pascal Giorgi (LIRMM, Université de Montpellier)

- **8h45.** **Clément Pernet (Aric Team, LIP, ENS-Lyon) :** *Calcul algébrique haute performance: comment utiliser l'arithmétique flottante exactement ?*

  Résumé : L'algèbre linéaire exacte est un composant de base du calcul algébrique haute performance, utilisé dans la résolution de problèmes en grande dimension des applications telles que la cryptanalyse algébrique, ou la théorie des nombres effective. En dépit de la diversité des domaines de coefficients à manipuler, les corps finis de la taille d'un mot machine jouent un rôle prépondérant car ils permettent de fournir le plus haut débit de calcul. Nous expliquerons pourquoi et comment les noyaux de calcul efficaces pour l'algèbre linéaire exacte sont construits. Ce faisant nous montrerons pourquoi l'arithmétique flottante reste la meilleure façon d'effectuer du calcul exact, en étudiant l'évolution des micro-architectures des dix dernières années. Nous montrerons ensuite les similarités et spécificités entre le calcul exact et le calcul numérique dans le contexte de la parallélisation de routines d'élimination de Gauss.

- **9h45.** **Vincent Neiger (Aric Team, LIP, ENS-Lyon) :** *Fast computation of shifted normal forms of polynomial matrices using polynomial approximation.*

  Résumé : This talk gives an overview of fast algorithms for performing lattice basis reduction over the polynomials. Given an $m \times m$ polynomial matrix $A(X)$ with entries of degree at most d, this problem asks to find an equivalent matrix R with some type of minimal degree, and can be solved in $O(m^3 d^2)$ operations in the base field using a simple iterative algorithm.

  Given the size $O(m^2 d)$ of the input A and output R, a first goal is to achieve the cost bound $O^{\sim}(m^w d)$, with w the exponent of matrix multiplication. We will observe that a major difficulty towards this is that the unimodular transformation matrix U leading from A to R may have large degrees, and thus cannot be computed in $O^{\sim}(m^w d)$ in general. As a consequence, the fastest known algorithms obtained this cost bound by avoiding the computation of U and resorting to a two-step strategy: they first compute a set of modular polynomial equations which describe the row space of A, and then find R as a reduced basis of solutions for this approximation problem.

  Then, three interesting improvements are algorithms which are in $O^{\sim}(m^w d)$ for d some type of average degree of the entries in A, which support so-called shifts that specify degree weights to be used for the reduction, and which compute

canonical forms such as the Popov form or the Hermite form. We will present a recent algorithm which, relying on a two-step strategy as above and on recent advances on polynomial approximation, achieves these improvements.

- **10h15.** Pause.

- **10h45. Bruno Grenet (LIRMM, Universite de Montpellier) :** *Calcul de racines de polynômes sur un corps fini à l'aide de transformées de Graeffe.*

  Co-auteurs : Grégoire Lecerf et Joris van der Hoeven.

  Résumé : Le calcul des racines de polynômes dans un corps fini est un problème important du calcul formel. Plusieurs algorithmes probabilistes de complexité moyenne polynomiale ont été proposés, mais aucun algorithme déterministe n'est actuellement connu. Nous avons récemment proposé une nouvelle approche pour ce problème, basée sur un outil issu de l'analyse numérique et appelé transformée de Graeffe. Cette approche nous a permis de donner plusieurs algorithmes, déterministe, probabiliste et heuristique. Ces algorithmes sont particulièrement adaptés aux corps finis de type FFT, c'est-à-dire dont le cardinal du groupe multiplicatif est friable, pour lesquels nous obtenons des améliorations de complexité en théorie comme en pratique.

- **11h15. Svyatoslav Covanov (Caramba team, LORIA, Université de Lorraine) :** *Fast integer multiplication using generalized Fermat primes.*

  Résumé : For almost 35 years, Schönhage-Strassen's algorithm has been the fastest algorithm known for multiplying integers, with a time complexity $O(n \times \log n \times \log\log n)$ for multiplying n-bit inputs. In 2007, Fürer proved that there exists $K > 1$ and an algorithm performing this operation in $O(n \times \log n \times K^{(\log^* n)})$. Recent work by Harvey, van der Hoeven, and Lecerf showed that this complexity estimate can be improved in order to get $K = 8$, and, via a conjecture on Mersenne primes, $K = 4$. This presentation will describe an alternative algorithm using generalized Fermat primes for which $K = 4$ conjecturally. This algorithm seems more simple than the algorithm relying on Mersenne primes and can be seen as a fix to an approach proposed by Fürer in 1989 using Fermat primes.

- **11h45. Fredrik Johansson (LFANT team, INRIA Bordeaux sud-ouest, Institut de Mathématiques de Bordeaux) :** *Fast reversion of formal power series*

  Résumé : Composition of formal power series is a fundamental operation in computer algebra. In the 70s, Brent and Kung discovered two composition algorithms with better asymptotic complexity than the classical implementation of Horner's rule. Brent and Kung also observed that Newton iteration allows computing the compositional inverse (reversion) of a power series, with at most a constant factor slowdown compared to composition. In a recently published paper, I show that there is a fast way to compute the reversion directly, without any Newton iteration. The idea is to implement the Lagrange inversion formula using a baby-step giant-step scheme and structured matrix multiplication. The new algorithm is analogous to one of the Brent-Kung composition algorithms, and has the essentially the same asymptotic complexity, but it is faster by a constant factor. It appears to be the best reversion algorithm in practice. In this talk, I discuss the algorithms and their behavior with different coefficient types, including numerical intervals.

Session Reproductibilité, Validation et Vérification - Guillaume Revy (DALI/LIRMM, UPVD)

- **14h30. Christophe Denis (CMLA, ENS Cachan, Université Paris-Saclay, CNRS) :** *Verificarlo: checking floating point accuracy through Monte Carlo Arithmetic.*

  Co-auteurs : Pablo de Oliveira Castro and Eric Petit.

  Résumé : Numerical accuracy of floating point computation is a well studied topic which has not made its way to the end-user in scientific computing. Yet, it has become a critical issue with the recent requirements for code modernization to harness new highly parallel hardware and perform higher resolution computation. To democratize numerical accuracy analysis, it is important to propose tools and methodologies to study large use cases in a reliable and automatic way. In this paper, we propose verificarlo, an extension to the LLVM compiler to automatically use Monte Carlo Arithmetic in a transparent way for the end-user. It supports all the major languages including C, C++, and Fortran. Unlike source-to-source approaches, our implementation captures the influence of compiler optimizations on the numerical accuracy. We illustrate how Monte Carlo Arithmetic using the verificarlo tool outperforms the existing approaches on various use cases and is a step toward automatic numerical analysis.

- **15h00.** **Chemseddine Chohra (DALI-LIRMM, Université de Perpignan) :** *Une Implémentation Reproductible, Précise et Efficace des BLAS.*

  Co-auteurs : Philippe Langlois, David Parello.

  Résumé : Le problème de non-reproductibilité des résultats numériques apparait dans les programmes parallèles à cause de la non-associativité de l'addition flottante. Pour permettre une exploitation efficace des systèmes massivement parallèles, les programmes autorisent le réordonnancement des opérations flottantes dynamiquement. Par conséquent les résultats de calcul peuvent changer d'une exécution à une autre même en ayant les mêmes données en entrée. Nous proposons d'assurer la reproductibilité en étendant la propriété d'arrondi au plus près exigée pour les opérations élémentaires à des séquences de calculs plus larges. Pour cela nous avons introduit notre bibliothèque RARE-BLAS (Reproducible, Accurately Rounded and Efficient BLAS) qui se base sur les algorithmes de sommation les plus performants pour offrir une implémentation des BLAS qui est à la fois précise et efficace. Nous présentons des solutions pour des routines de niveau 1 (asum, dot et nrm2) et niveau 2 (gemv). Les performances de notre bibliothèque sont comparées à la bibliothèque Intel MKL, ainsi que d'autres solutions reproductibles. Le surcoût de notre implémentation parallèle par rapport à MKL est 2 fois plus au pire cas (que cela soit en mémoire partagée ou distribuée), ce qui est acceptable pour la plupart des applications en pratique. Nous montrons aussi des résultats sur l'accélérateur Intel Xeon Phi. Le surcoût dans ce cas est plus important (4 à 6 fois plus), ce qui reste utilisable pour le débogage ou la validation d'un programme.

- **15h30.** **Rafife Nheili (DALI-LIRMM, Université de Perpignan) :** *Recovering numerical reproducibility in hydrodynamic simulations.*

  Co-auteur : Philippe Langlois.

  Résumé : HPC simulations suffer from failures of numerical reproducibility because of floating-point arithmetic peculiarities. Different computing distributions of a parallel computation may yield different numerical results. We are interested in a finite element computation of hydrodynamic simulations within the openTelemac software where parallelism is provided by domain decomposition. One main task in a finite element simulation consists in building one large linear system and to solve it. Here the building step relies on element-by-element storage mode and the solving step applies the conjugated gradient algorithm. The subdomain parallelism is merged within these steps. We study why reproducibility fails in this process and which operations have to be corrected. We detail how to use compensation techniques to compute a numerically reproducible resolution. We illustrate this approach presenting the reproducible version of hydrodynamic simulations for one test cases provided with the openTelemac software suite.

- **16h00.** Pause.

- **16h30.** **Sylvie Boldo (Toccatta (INRIA Saclay Ile-de-France), LRI (Université Paris Sud) :** *Renormalisation d'expansions: une vérification formelle.*

  Co-auteurs : Mioara Joldes, , Jean-Michel Muller , Valentina Popescu.

  Résumé : Les expansions flottantes permettent de calculer avec une précision supérieure et un coût raisonnable. Mais les algorithmes usuels d'addition, multiplication... rendent des expansions qui peuvent se chevaucher. Il est donc nécessaire de pouvoir normaliser ces expansions. Un algorithme efficace dû à Joldes et al. existe, avec une preuve papier convaincante. Ce travail est la vérification formelle de cette preuve papier, avec la découverte de toutes ses lacunes, prévisibles ou non prévisibles.

- **17h15.** **Guillaume Melquiond (Toccatta (INRIA Saclay Ile-de-France), LRI (Université Paris Sud) :** *Formally Verified Approximations of Definite Integrals.*

  Résumé : Finding an elementary form for an antiderivative is often a difficult task, so numerical integration has become a common tool when it comes to making sense of a definite integral. Some of the numerical integration methods can even be made rigorous: not only do they compute an approximation of the integral value but they also bound its inaccuracy. Yet numerical integration is still missing from the toolbox when performing formal proofs in analysis.

  This talk presents an efficient method for automatically computing and proving bounds on some definite integrals inside the Coq formal system. Our approach is not based on traditional quadrature methods such as Newton-Cotes formulas. Instead, it relies on computing and evaluating antiderivatives of rigorous polynomial approximations, combined with an adaptive domain splitting. This work has been integrated to the CoqInterval library.

- **17h45.** Pause.

- **18h00.** Intervention du GDR et assemblée générale.

Jeudi 30 juin

Session Opérateurs - Christoph Lauter (LIP6, UPMC)

- **8h45.** **Florent de Dinechin (Socrate Team, CITI INRIA, INSA-Lyon) :** *Computing the floating-point logarithm with fixed-point operations.*

  Résumé : Elementary functions from the mathematical library input and output floating-point numbers. However it is possible to implement them purely using integer/fixed-point arithmetic. This option was not attractive between 1985 and 2005, because mainstream processor hardware supported 64-bit floating-point, but only 32-bit integers.

  This has changed in recent years, in particular with the generalization of native 64-bit integer support. The purpose of this article is therefore to reevaluate the relevance of computing floating-point functions in fixed-point. For this, several variants of the double-precision logarithm function are implemented and evaluated.

  Formulating the problem as a fixed-point one is easy after the range has been (classically) reduced. Then, 64-bit integers provide slightly more accuracy than 53-bit mantissa, which helps speed up the evaluation. Finally, multi-word arithmetic, critical for accurate implementations, is much faster in fixed-point, and natively supported by recent compilers.

  Thanks to all this, a purely integer implementation of the correctly rounded double-precision logarithm outperforms the previous state of the art, with the worst-case execution time reduced by a factor 5.

  This work also introduces variants of the logarithm that input a floating-point number and output the result in fixed-point. These are shown to be both more accurate and more efficient than the traditional floating-point functions for some applications.

  All this is advertising for the MetaLibm project.

- **10h15.** Pause.

- **10h45.** **Bogdan Pasca (Altera, Toulouse) :** *Single Precision Natural Logarithm Architecture for Hard Floating-Point and DSP-Enabled FPGAs*

  Résumé : In this talk I will present a novel method for implementing the single-precision floating-point (FP) natural logarithm function using the new FP single precision addition and multiplication features of the Altera Arria 10 DSP Block architecture. log(x) is one of the most commonly required functions for emerging datacenter and computing FPGA targets. In this talk I will cover the error analysis, both for the overall function, and each subsection of the architecture, demonstrating that the hard FP (HFP) Blocks, in conjunction with the traditional flexibility and connectivity of the FPGA, can provide a robust and high performance solution. These methods create a highly accurate single precision IEEE-754 function, which is OpenCL conformant. Our methods map directly to almost exclusively embedded structures, and therefore result in significant reduction in logic resources and routing stress compared to current methods, and demonstrate that newly introduced FPGA routing architectures can be leveraged to use almost no soft resources.

- **11h15.** **Volkova Anastasia (Equipe Pequan, LIP6, Université Pierre et Marie Curie) :**. *Towards reliable implementation of digital filters.*

  Résumé : The implementation of a Digital Filter can be considered as a path from a filter specification to a physical implementation (e.g. a DSP device). This process includes generation of a transfer function, which describes the relation between the inputs and outputs of a filter. Further one needs to choose a filter structure to be eventually implemented, i.e. a computational scheme. Finally, there is software and hardware implementation, which is usually done under constraints. However, on each step various issues arise. For example, due to transfer function discretization the roundoff errors in the evaluation of the filter need to be taken into account.

  To unify the above-described process of filter implementation for any LTI filter and to provide a consistent error-analysis of filter design process, we develop an atomatized filter generator. We take into account the computational errors, which

are due to the finite precision implementation, perform rigorous error-analysis of the algorithms and perform various optimizations to meet user-provided criteria, such as output error constraints.

In this talk we focus on determining the Fixed-Point formats for an implementation that is optimal with respect to certain criteria. We consider Linear Time Invariant filters in state-space representation. The computational errors in the intermediate steps of the filter evaluation as well as their accumulation over time are fully taken into account. Our approach is fully rigorous in the way that the output Fixed-Point formats are shown to be free of overflows and that we do not use any (non-exhaustive) filter simulation steps but proofs.

- **11h45.** **De Lassus Saint-Genies Hugues (DALI-LIRMM, Université de Perpignan) :** *Performances de schémas d'évaluation polynomiale sur architectures vectorielles.*

  Co-auteur : Guillaume Revy.

  Résumé : Les fonctions élémentaires sont souvent calculées à l'aide d'approximations polynomiales, dont l'efficacité dépend directement de celle du schéma d'évaluation sous-jacent. Cet article montre que le schéma classiquement utilisé (Horner) est rarement le plus performant. En effet, d'autres schémas exploitent mieux les parallélismes des architectures modernes, en réduisant les dé- pendances de données. Ces résultats ont pour objectif d'être intégrés à un générateur de code performant pour l'évaluation polynomiale dans le cadre de l'approximation de fonctions.

Session Arithmétique et Cryptographie : Christophe Negre (DALI/LIRMM, UPVD)

- **14h00.** **Nicolas Méloni (IMATH , Université de Toulon) :** *Random Digit Representation of Integers.* **Co-auteurs :** Anwar Hasan.

  Résumé : Modular exponentiation, or scalar multiplication, is core to today's main stream public key cryptographic systems. In this article we generalize the classical fractional wNAF method for modular exponentiation - the classical method uses a digit set of the form $\{1,3,\ldots,m\}$ which is extended here to any set of odd integers of the form $\{1,d_2,\ldots,d_n\}$. We propose a general modular exponentiation algorithm based on a generalization of the frac-wNAF recoding and a new precomputation scheme. We also give general formula for the average density of non-zero therms in these representations, prove that there are infinitely many optimal sets for a given number of digits and show that the asymptotic behavior, when those digits are randomly chosen, is very close to the optimal case.

- **14h30.** **Mouhartem Fabrice (Aric Team, LIP, ENS-Lyon) :** *Signatures de Groupes et Réseaux Euclidiens.*

  Co-auteurs : Benoît Libert, Khoa Nguyen.

  Résumé : La signature de groupe est une primitive cryptographique introduite en 1991 par Chaum et van Heyst qui permet à un ensemble d'utilisateurs (le groupe) de produire une signature digitale au nom du groupe. En d'autres termes, un utilisateur va garantir que le message a bien été envoyé par un membre du groupe, et qu'il n'a pas été altéré entre temps. Dans le même temps, tout le monde peut vérifier que la signature est correcte, sans divulguer d'autres informations sur l'identité du signataire que son appartenance au groupe. En revanche, pour prévenir des abus possible sur l'anonymat, une entité est capable à l'aide d'une information secrète de lever l'anonymat d'un utilisateur sur un message malhonnête par exemple, c'est l'autorité d'ouverture. Durant la dernière décennie, la cryptographie reposant sur la difficulté des problèmes sur les réseaux euclidiens a été de plus en plus étudié en permettant des avancées comme le chiffrement totalement homomorphe, dont les solutions alternatives restent anecdotiques. La cryptographie s'appuyant sur les réseaux euclidiens est de plus un candidat à la cryptographie post-quantique, sachant que les solutions actuelles reposant sur l'arithmétique modulaire pourraient être mises à mal par un adversaire disposant d'un calculateur quantique. C'est pourquoi nous avons travaillé à étendre le tableau des solutions existantes sur les réseaux euclidiens en proposant un schéma de signature de groupes autorisant les admissions au groupe de manière interactive et dynamique, ainsi qu'une autre extension permettant le limiter la puissance de l'autorité d'ouverture en rajoutant une autre entité devant donner son accord (par le biais d'un jeton dépendant du message) pour l'ouverture d'une signature.

- **15h00.** **Tania Richmond (IMATH, Université de Toulon) :** *DPA on the Secure Bit Permutation in the McEliece PKC.*

  Co-auteurs : Martin Petrvalský, Miloš Drutarovský, Pierre-Louis Cayrel, Viktor Fischer.

  Résumé : In this talk, we present a differential power analysis attack on the McEliece public-key cryptosystem. We demonstrate that a part of a private key - permutation matrix - can be recovered using the power analysis. We attack a software implementation of a "secure" permutation that was proposed by Strenzke et al. at PQCrypto 2008. The cryptosystem is implemented on a 32-bit ARM based microcontroller and power consumption measurements of the device

provide us leakage. In addition, we outline a novel countermeasure against the introduced attack. The countermeasure uses properties of linear codes and does not require large amount of random bits which can be profitable for low-cost embedded devices.

- **15h30. Jean-Marc Robert (DALI-LIRMM, Université de Perpignan) :** *Enhanced Digital Signature with Radix R and RNS Digits Exponent Representation*.

  Co-auteur : Thomas Plantard.

  Résumé : Digital Signature involve modular exponentiation (or elliptic curve scalar multiplication of a point), of a public and known base (a base point, respectively) by a random one-time exponent (a scalar, respectively). In order to speed-up this operation, well-known methods take advantage of the memorization of powers of the exponent (of the multiples of the base point, respectively). However, due to the cost of the memory, to its small size and to the latency of access, previous research sought for minimization of the storage. In this paper, taking into account the modern processor features and the growing size of the cache memory, we improve the compromise storage/efficiency, by using a radix R and RNS Digits scalar/exponent representation. The complexity is smaller in comparison with classical binary representation, especially as the representation size grows for equivalent amount of storage, or the storage is lower for equivalent complexities. We then propose algorithms for modular exponentiation and elliptic curve scalar multiplication. The implementation performances show significant memory saving or speed-up.