Master 2 internship Project
Multimodular algorithms in low and mixed precision: residue number systems on AI hardware

We are seeking a top graduate candidate for a 5-month research internship at the interface of computer algebra, computer arithmetic, numerical linear algebra and high-performance computing. This internship is particularly appropriate for students willing to pursue a Ph.D. after obtaining their Master degree.

Scientific context. Driven by the needs of artificial intelligence, the computing power of modern processors is increasingly concentrated in **matrix multiply-accumulate units** (such as NVIDIA Tensor Cores) operating on **low-precision formats**: FP16, BF16, INT8, FP8 [13] and, most recently, block-scaled FP4. On such hardware, the throughput of these units exceeds that of the traditional FP64 pipelines by one to several orders of magnitude, and the gap keeps widening from one GPU generation to the next. Moreover, even the relative performance of the low-precision formats changes over time: on some recent architectures, INT8 throughput has been reduced in favour of floating-point formats [21]. Scientific computing, traditionally relying on double precision, must therefore adapt: this is the purpose of **mixed-precision algorithms** [1,12], which use low precision wherever possible and high precision only where it is needed, and, more generally, of **approximate computing**, where the working precision is tuned to the accuracy actually required by the application.

A particularly successful line of work is the **emulation** of a high-precision matrix product (DGEMM) by a sequence of low-precision ones. The original Ozaki scheme [17], now called *Ozaki-I*, splits each input matrix into a sum of slices with few significant bits, so that all pairwise products of slices can be computed *exactly* on low-precision units; it has been implemented on FP16, INT8 and FP8 Tensor Cores [14,16,20,22]. Its drawback is that the number of low-precision products grows *quadratically* with the number of slices, hence with the target precision.

The more recent *Ozaki-II* scheme [18] replaces slicing by a technique well known in computer algebra: the **Chinese remainder theorem (CRT)** and **residue number systems (RNS)** [6,7]. Given $A \in \mathbb{R}^{m\times k}$ and $B \in \mathbb{R}^{k\times n}$, it proceeds as follows:

- scale the rows of A and the columns of B by powers of two and round them to integer matrices A', B'; this is the *only* step introducing an error;
- choose pairwise coprime moduli $p_1, \ldots, p_s$ whose product exceeds twice a bound on the entries of $A'B'$, and reduce A', B' modulo each p_i, with entries small enough to be represented in the low-precision format;
- compute the s products $A'B' \bmod p_i$ *exactly* on the low-precision units, the accumulation being performed in a wider format;
- recover $A'B'$ by CRT reconstruction, then undo the scaling.

The number of low-precision products is now *linear* in the target precision, and the scheme behaves like the **multimodular algorithms** that have long been at the core of exact linear algebra over $\mathbb{Z}$ and $\mathbb{Q}$ [5,7,19]. Note, however, that the roles are reversed: in computer algebra one uses FP64 units to compute exactly modulo word-size primes, whereas here one uses INT8, FP8 or FP4 units to compute approximately in floating-point arithmetic.

Problem, state of the art, objectives. Three very recent works illustrate how fast the field is moving, and how algebraic ideas can lead to significant practical savings.

- Ozaki-II was first designed for INT8 units and cannot be directly adapted to FP8, whose significand is too short to hold the residues. In [21], Uchino, Ozaki and Imamura introduce a new technique making Ozaki-II run on FP8 units, at a significantly lower cost than the FP8 version of Ozaki-I. A related approach targets FP4 units by encoding integers as base-13 digits representable in the E2M1 format [9].
- Complex matrix products are classically computed with 4 real products, or 3 with the well-known 3M (Karatsuba-like) method [11]. In [3], Caday shows that, for matrices with *integer* real and imaginary parts, 2 real products suffice, up to quadratic pre- and post-processing. Combined with Ozaki-II, this yields a complex floating-point product in roughly twice the time of a real one.
- The AWE method [8] asks how many FP4 products are needed for a single exact integer matrix product. It splits integers into limbs with freely chosen integer weights, stores linear combinations of these limbs, and reconstructs the exact product as a weighted sum of FP4 products, using the headroom of the accumulator to reduce the number of products.

These results share a common structure, familiar to computer algebraists: **evaluation–interpolation**. Splitting into limbs, Karatsuba and Toom–Cook schemes, CRT over $\mathbb{Z}$ or over rings of algebraic integers such as $\mathbb{Z}[i]$, are all ways of mapping one large bilinear computation onto several small independent ones, and back. Yet the existing low-precision schemes are mostly designed case by case, for one given format and one given operation, and their costs (number of products, cost of the conversions to and from the residue representation, accuracy) are not compared in a unified framework.

The objectives of this internship are the following:

- **model and compare** the existing schemes (Ozaki-I, Ozaki-II and their INT8, FP8 and FP4 variants, 2M, AWE) within a unified evaluation–interpolation framework, parametrized by the input format, the accumulator width, the inner dimension k and the target precision; and derive, when possible, *lower bounds* on the number of low-precision products needed to reach a given accuracy;
- **bring computer algebra techniques** into these schemes: choice of moduli with special structure, moduli in extension rings (as in the 2M algorithm), mixed-radix and balanced representations, and fast RNS conversions expressed themselves as matrix products [4], so that the pre- and post-processing steps can also run on the low-precision units;
- **analyse the accuracy** [10,15] of these schemes in an approximate computing perspective: since CRT reconstruction is exact, the error comes from the initial scaling, which suggests adaptive strategies choosing the number of moduli according to the requested precision (FP32, FP64 or beyond) and to the dynamic range of the inputs.

The methodology and expected main tasks of this internship are as follows:

- after getting introduced to floating-point arithmetic and to mixed-precision algorithms [12,15], and to multimodular algorithms [7, Ch. 5], study the Ozaki schemes [17,18] and the recent works [2,3,8,21];
- design and analyse (cost and accuracy) new variants, starting with the combination of the above techniques, e.g. 2M or AWE-like encodings on FP8 residues, and with RNS conversions performed by matrix products;
- prototype the algorithms and validate them experimentally, first by simulating the low-precision formats, then, depending on the candidate's interests, on actual GPU hardware;
- **study perspectives for these new results**: beyond matrix multiplication, identify which higher-level operations (linear system solving with iterative refinement, factorizations, polynomial and structured matrix products) would benefit from these techniques, and at which precision.

Software implementations. The computer algebra community has developed a number of high-quality software libraries that provide efficient implementations of fundamental algorithms, several of which rely on multimodular techniques and floating-point units: FFLAS-FFPACK [5] and LinBox for exact linear algebra, NTL and Flint for integer and polynomial arithmetic. Several of these libraries have been recognized with the ACM/SIGSAM Richard D. Jenks Memorial Prize (https://www.sigsam.org/awards/jenks/awardees/). On the

numerical side, open-source implementations of the Ozaki schemes on GPUs are available (e.g. `ozIMMU` and `GEMMul8`), built on top of vendor libraries such as cuBLAS and CUTLASS.

In this project, we will rely on SageMath, Python or Julia for prototyping and for simulating low-precision formats, in conjunction with C/C++ libraries such as `Flint` and `FFLAS-FFPACK`, and with CUDA for GPU experiments.

Scientific environment. This internship will be co-supervised by Jérémy Berthomieu, Stef Graillat, and Théo Mary. It will take place at Sorbonne Université, in the computer science lab LIP6, which is located on the Pierre-et-Marie-Curie campus, at the heart of Paris. The intern will be welcome in both the PolSyS and PEQUAN teams, which which develops and implements fast computer algebra and numerical algorithms with high-performance computing techniques. The intern will work in a kind and international environment gathering PhD students and Post-Docs representing several nationalities, and animated with several working groups and a monthly seminar. All computing facilities will be provided, including access to GPUs equipped with low-precision matrix units.

How to apply. Interested applicants should send a full CV, a letter of motivation and the grades obtained during the last two years to

jeremy.berthomieu@lip6.fr, stef.graillat@lip6.fr, theo.mary@lip6.fr.

References

[1] A. Abdelfattah et al. "A survey of numerical linear algebra methods utilizing mixed-precision arithmetic". In: *The International Journal of High Performance Computing Applications* (2021).

[2] J. Berthomieu, S. Graillat, D. Lesnoff, and T. Mary. *Multiword matrix multiplication over large finite fields in floating-point arithmetic*. 2026. HAL: hal-04917201.

[3] P. Caday. *The 2M multiplication algorithm for complex matrices*. 2026. arXiv: 2609.05419.

[4] J. Doliskani, P. Giorgi, R. Lebreton, and É. Schost. "Simultaneous conversions with the residue number system using linear algebra". In: *ACM Transactions on Mathematical Software* (2018).

[5] J.-G. Dumas, P. Giorgi, and C. Pernet. "Dense linear algebra over word-size prime fields: the FFLAS and FFPACK packages". In: *ACM Transactions on Mathematical Software* (2008).

[6] H. L. Garner. "The residue number system". In: *IRE Transactions on Electronic Computers* (1959).

[7] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. 3rd ed. Cambridge University Press, 2013.

[8] S.-i. Hayashi, D. Mukunoki, T. Hoshino, and T. Katagiri. *AWE: adaptive weight encoding for exact integer matrix products with fewer GEMMs on FP4 Tensor Cores*. 2026. arXiv: 2609.24519.

[9] S.-i. Hayashi, D. Mukunoki, T. Hoshino, and T. Katagiri. *DGEMM with Ozaki Scheme I/II on FP4 Tensor Cores: a base-13 E2M1 limb representation*. 2026. arXiv: 2608.06812.

[10] N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. 2nd ed. SIAM, 2002.

[11] N. J. Higham. "Stability of a method for multiplying complex matrices with three real matrix multiplications". In: *SIAM Journal on Matrix Analysis and Applications* (1992).

[12] N. J. Higham and T. Mary. "Mixed precision algorithms in numerical linear algebra". In: *Acta Numerica* (2022).

[13] P. Micikevicius et al. *FP8 formats for deep learning*. 2022. arXiv: 2209.05433.

[14] D. Mukunoki, K. Ozaki, T. Ogita, and T. Imamura. "DGEMM using Tensor Cores, and its accurate and reproducible versions". In: *High Performance Computing (ISC 2020)*. 2020.

[15] J.-M. Muller, N. Brunie, F. de Dinechin, C.-P. Jeannerod, M. Joldes, V. Lefèvre, G. Melquiond, N. Revol, and S. Torres. *Handbook of Floating-Point Arithmetic*. 2nd ed. Birkhäuser, 2018.

[16] H. Ootomo, K. Ozaki, and R. Yokota. *DGEMM on integer matrix multiplication unit*. 2023. arXiv: 2306.11975.

[17] K. Ozaki, T. Ogita, S. Oishi, and S. M. Rump. "Error-free transformations of matrix multiplication by using fast routines of matrix multiplication and its applications". In: *Numerical Algorithms* (2012).

[18] K. Ozaki, Y. Uchino, and T. Imamura. *Ozaki Scheme II: a GEMM-oriented emulation of floating-point matrix multiplication using an integer modular technique*. 2025. arXiv: 2504.08009.

[19] C. Pernet. "Exact linear algebra algorithmic: theory and practice". In: *Proceedings of ISSAC 2015*. 2015.

[20] A. Schwarz et al. *Guaranteed DGEMM accuracy while using reduced precision Tensor Cores through extensions of the Ozaki scheme*. 2025. arXiv: 2511.13778.

[21] Y. Uchino, K. Ozaki, and T. Imamura. *Double-precision matrix multiplication emulation via Ozaki-II scheme with FP8 quantization*. 2026. arXiv: 2603.10634.

[22] Y. Uchino, K. Ozaki, and T. Imamura. *Performance enhancement of the Ozaki scheme on integer matrix multiplication unit*. 2024. arXiv: 2409.13313.