Overcoming the memory bottleneck in differential algebra

Supervisors:

François Boulier	Francois.Boulier@univ-lille1.fr	http://cristal.univ-lille.fr/~boulier
Pierre Fortin	Pierre.Fortin@univ-lille1.fr	http://lip6.fr/Pierre.Fortin
François Lemaire	Francois.Lemaire@univ-lille1.fr	http://cristal.univ-lille.fr/~lemaire

Hosting laboratory:
CRIStAL (http://www.cristal.univ-lille.fr/CFHP/), Université de Lille

Hosting team:
CFHP team (*Calcul Formel et Haute Performance* - Computer Algebra and High Performance Computing, http://www.cristal.univ-lille.fr/CFHP/)

Duration: 5 or 6 months, starting from February or March 2018

Funding: standard pay of Université de Lille

Keywords: Rosenfeld-Gröbner algorithm, differential algebra, memory constraint, algorithmic, out-of-core algorithms

Scientific context. Poor blood flow in the human brain may trigger strokes which result in cell death. These damages can lead to long-term neurological or motor deficits which constitute a major health issue. In order to better understand and prevent such strokes, strong scientific progress is required concerning the fundamental role of the neuron-astrocyte interaction. Current mathematical models of this interaction are too simple, which has led the CFHP team to investigate new biological models based on differential and integro-differential algebras [1].

In order to tackle these models, the CFHP team has developed computer algebra software applications. Among these, the BLAD[1] software aims at solving systems of differential polynomial equations. BLAD relies on the Rosenfeld-Gröbner algorithm [2] (denoted RG below) which simplifies such systems thanks to differential elimination methods based on GCD (greatest common divisor) computations.

However, depending on the system to solve, the memory requirements can greatly increase: this is referred to as the expression swell. For example, some systems we target cannot be solved on a 512 GB server: they indeed require GCD computations among huge polynomials whose (compressed) storage on disk require up to 70MB.

[1] See: http://www.lifl.fr/~boulier/BLAD/

Objectives. In order to overcome this memory bottleneck, and depending on the student interests and affinities, we plan to investigate the following research directions. These directions mainly trade memory savings with extra computations.

- Instead of fully developing the (sparse) polynomials in memory, we can rely on partial factorizations of these polynomials and on suitable data structures. This will enable to save memory, but will also require to adapt the operations performed on these polynomials (evaluations, multiplications, GCD computations ...) so that they can process polynomial parts.
- It is also possible to change the strategy used in the reduction algorithm and in the RG algorithm: new strategies taking into account the remaining memory space can here be designed.
- Finally, out-of-core algorithms (arised from high performance computing) can also be used here. These out-of-core algorithms consist in storing, during the computation, parts of the data on disk in order to save RAM memory. The control of these expensive load and store operations from/to the disk, along with the overlap of these operations with computations (thanks to asynchronous I/O, see e.g. [3]), can result in efficient implementations of these algorithms.

In practice, we will rely on two HPC servers: a first one with two 18-core CPUs and 128GB of memory, and a second one with two 10-core CPUs and 512GB of memory.

In the end, this work will enable to solve biological models never solved before, and to investigate new and more difficult systems and models.

Required skills: Interest in scientific computing, strong motivation and reasoning are mainly required, as well as C programming. No skill in computer algebra is required.

This internship may be followed by a Ph.D. thesis in the same research area.

References:

[1] F. Boulier, F. Lemaire, M. Rosenkranz, R. Ushirobira and N. Verdière. On Symbolic Approaches to Integro-Differential Equations. Algebraic and Symbolic Computation Methods in Dynamical Systems, Advances in Delays and Dynamics, Springer, 2017.

[2] F. Boulier, D. Lazard, F. Ollivier, M. Petitot. Computing representations for radicals of finitely generated differential ideals. Journal of AAECC, vol. 20(1):73-121, 2009.

[3] I. Said, P. Fortin, J.L. Lamotte and H. Calandra. Leveraging the Accelerated Processing Units for seismic imaging : a performance and power efficiency comparison against CPUs and GPUs. International Journal of High Performance Computing Applications (to appear).