Brief Announcement: An Implementation Report for Parallel Triangular Decompositions

Marc Moreno Maza and Yuzhen Xie
ORCCA, University of Western Ontario (UWO)
London, Ontario, Canada
moreno@orcca.on.ca, yxie@orcca.on.ca

Categories and Subject Descriptors:
I.1.2 [**Computing Methodologies**]: Symbolic and Algebraic Manipulation – *Algebraic Algorithms*;
D.1.3 [**Software**]: Programming Techniques – *Parallel Programming*

General Terms:
Algorithms, Design, Experimentation, Performance.

Keywords: Polynomial System Solver, Parallel Execution.

ABSTRACT

Since the discovery of Gröbner bases, the algorithmic advances in Commutative Algebra have made possible to tackle many classical problems in Algebraic Geometry that were previously out of reach. However, algorithmic progress is still desirable, for instance when solving symbolically a large system of algebraic non-linear equations. For such a system, in particular if its solution set consists of geometric components of different dimension (points, curves, surfaces, etc) it is necessary to combine Gröbner bases with decomposition techniques, such as triangular decompositions. Ideally, one would like each of the different components to be produced by an independent processor, or set of processors. In practice, the input polynomial system, which is hiding those components, requires some transformations in order to split the computations into sub-systems and, then, lead to the desired components. The efficiency of this approach depends on its ability to detect and exploit geometrical information during the solving process.

Our work addresses two questions: How to discover geometrical information, at an early stage of the solving process, that would be favorable to parallel execution? How to ensure load balancing among the processors? We answer these questions in the context of triangular decompositions [2] which are a popular way of solving polynomial systems symbolically. These methods tend to split the input polynomial system into subsystems and, therefore, are natural candidate for parallel implementation. However, the only such method which has been parallelized so far is the Characteristic Set Method of Wu [5], as reported in [1, 6]. This approach suffers from several limitations. For instance, the solving of the second component cannot start before that of the first one is completed; this is a limitation in view of coarse-grain parallelization.

In [4] an algorithm, called Triade, for *TRIAngular DEcompositions*, provides a good management of the intermediate computations for triangular decompositions. It is also a natural candidate for coarse-grain parallel implementation based on geometrical considerations; indeed the number of working processors can depend on the intrinsic difficulty of the system to solve. However, several challenges remain to be considered. First, load balancing is very difficult to control due to irregular tasks. Even worse: for some input polynomial systems, especially with integer coefficients, resource consuming tasks may not be necessarily executed concurrently. Second, data communication overhead can be very heavy due to large intermediate results.

In order to achieve load balancing we rely on the following facts. For an input polynomial system, the Triade algorithm generates the intermediate or output components by decreasing order of dimension. As a consequence, expensive tasks (those in lower dimension) can be processed concurrently. In addition, when solving a (non-trivial) polynomial system modulo a prime integer, the number of these tasks is sufficient for expecting a good speed-up in a parallel execution. The case of polynomial systems with integer coefficients can also benefit from these features by using the modular techniques introduced in [3].

We have developed a parallel scheme for the Triade algorithm, aiming at minimizing data communication overhead. Tasks are scheduled and updated by a *process manager*. Individual tasks are solved "lazily" by *process workers*. However, each *process worker* keeps track of enough information such that it can continue the solving of some of these tasks, when needed.

We have realized a preliminary implementation on a shared memory multiprocessor. The experimental results show a satisfactory speed-up for some well-known problems.

SPAA'06, July 30–August 2, 2006, Cambridge, Massachusetts, USA.
ACM 1-59593-452-9/06/0007.

1. REFERENCES

[1] I. A. Ajwa. *Parallel Algorithms and Implementations for the Gröbner Bases Algorithm and the Characteristic Set Method.* PhD thesis, Kent State University, Kent, Ohio, 1998.

[2] P. Aubry, D. Lazard and M. Moreno Maza. On the Theories of Triangular Sets. *J. Symb. Comp.*, 28:105–124, 1999.

[3] X. Dahan, M. Moreno Maza, É. Schost, W. Wu, and Y. Xie. Lifting techniques for triangular decompositions. In *ISSAC'05*, pages 108–115. ACM Press, 2005.

[4] M. Moreno Maza. On triangular decompositions of algebraic varieties. MEGA-2000 Conference, Bath, 2000.

[5] W. T. Wu. A zero structure theorem for polynomial equations solving. *MM Research Preprints*, 1:2–12, 1987.

[6] Y.W. Wu, G.W. Yang, H. Yang, W.M. Zheng, and D.D. Lin. A distributed computing model for wu's method. *Journal of Software (in Chinese)*, 16(3), 2005.