

CoCoA

Computations in Commutative Algebra

CoCoA Team¹

1 What is CoCoA?

CoCoA is a freely available system for computing with multivariate polynomials.

More specifically, CoCoA deals with computations in multivariate polynomial rings over the rationals or modular integers, and on their ideals and modules. The implementation of the ideal/module theoretic operations relies on Gröbner basis theory.

One of CoCoA's most important features is its high-level programming language which allows the user to write his own functions and to guide the system through complicated and involved computations. This language has been designed to be natural and intuitive, so it is easy to learn and is well suited to teaching. The language is interpreted; its syntax is Pascal-like.

The main users of CoCoA are researchers in Commutative Algebra and Algebraic Geometry together with their students. However, computational algebraic techniques are spreading to other fields (for instance numerical analysis, cryptography, statistics, and dynamical systems), and consequently so is the realm of application of CoCoA.

The system includes *complete on-line help* (also available as an *html-manual*). CoCoA is *freely available* software for research and educational purposes which can be obtained by

<http://cocoa.dima.unige.it>

as well as from the mirror site in America

<http://www.reed.edu/mirrors/cocoa>

For convenience, the system offers a textual interface, an Emacs mode, and a graphical user interface common to most platforms.

CoCoA is mentioned in some of the *most widely used text books* in Computational Algebra, and plays a major role in the books *Computational Commutative Algebra* (1 and 2) by Martin Kreuzer and Lorenzo Robbiano.

¹J Abbott, A Bigatti, M Caboara, M Kreuzer, and L Robbiano

2 History, aims and scope of CoCoA

The CoCoA project began in 1987. Initially it comprised just a small program in Pascal running only on Macintoshes, and written simply to enable the authors to experiment with Buchberger’s algorithm for computing Gröbner bases. Within two years it had become widely used in many countries both for research and teaching. Later the program was translated into C, and ported to other platforms. As its range of abilities grew so did its use by researchers and teachers.

The most recent versions benefit from development directed at areas too often neglected in academic environments notwithstanding their unquestionably vital contributions to the usefulness of the program: *e.g.* robustness, natural syntax for (computer-shy) mathematicians, a sophisticated (graphical) user interface, and comprehensive documentation. Naturally, there has also been continual development in more academically “respectable” areas, often spurred on by symbiosis with researchers using CoCoA not only in algebraic geometry but also in other fields such as mathematical analysis, and statistics.

An important purpose of the CoCoA program is to provide a “laboratory” for studying computational commutative algebra: it together with Singular and Macaulay 2 form an elite group of highly specialized systems having as their main forte the capability to calculate Gröbner bases. Although a number of general purpose symbolic computation systems (*e.g.* REDUCE and Maple) do offer the possibility of computing Gröbner bases, their non-specialist nature implies a number of severe compromises which make them far less suitable to act as a laboratory: *e.g.* relatively poor execution speed and limited control over the algorithm parameters.

Aside from computing Gröbner bases CoCoA’s particular strengths include polynomial factorization, exact linear algebra, Hilbert functions, toric ideals, and ideals of zero-dimensional schemes.

3 Mathematical functionality

CoCoA works with multivariate polynomials over the fields of modular integers or rational numbers (exact rationals, not floating point numbers). Naturally, basic arithmetic operations are provided together with more advanced operations such as GCD, composition, and factorization.

The default ring environment is $\mathbb{Q}[x, y, z]$, but one may declare and activate another polynomial ring (via the command `Use`). Data may be transferred from one ring to another by defining and applying a k -algebra homomorphism (via the command `Image`).

Gröbner bases and Orderings

The key tool for effective computations in Commutative Algebra is the concept of Gröbner Basis. Almost every ideal and module theoretic operation is realized via some Gröbner Basis computations: these are effected by Buchberger’s

Algorithm whose worst-case complexity is quite high (in fact, doubly exponential in the number of variables). Therefore an efficient implementation of the Buchberger's Algorithm is the heart of any Commutative Algebra System.

To perform Gröbner basis computations it is necessary to specify a *term-ordering*, i.e. a total ordering which is compatible with multiplication, and such that each indeterminate is bigger than 1. Such orderings are characterized by suitable matrices. Different orderings may lead to different Gröbner bases for the same ideal with quite different computational costs (in space and time). **CoCoA** offers some pre-defined orderings and the ability to specify any matrix-defined term-ordering.

Ideal and module operations

Among the types implemented in **CoCoA** there are ideals and submodules of a free finitely generated module over a polynomial ring. Basic operations on these objects are performed via Gröbner basis computations (*e.g.* intersection, elimination and saturation, radical membership, free resolution of ideals or modules).

4 Technical Information

Extensive information about the **CoCoA** system and its associated research is available at the web site <http://cocoa.dima.unige.it>. You may freely download executable and documentation. You can also find a full list of the platforms on which **CoCoA** runs with instructions for installation, and information regarding the system and associated research.

The **CoCoA** kernel is written in C (about 70,000 lines). It runs on Linux, MacOS X, Microsoft Windows, and other platforms.

The accompanying library of packages, comprising around 17,000 lines of **CoCoA** language code, adds advanced capabilities in other areas such as statistics, integer programming, and invariant theory. The complete list of packages is available at <http://cocoa.dima.unige.it/download/packages>.

5 The future of **CoCoA**

Aside from the normal development, a number of more specific plans are afoot to improve and extend **CoCoA**: the choice of coefficients is to be widened to handle parameters and finite algebraic extensions; and the mathematical core will be made available as a software library written in C++ facilitating integration into other systems.