EUROCAL'85 éd. B. Caviness
LNCS n° 204, Springer Verlag
p. 289-290

ABOUT A NEW METHOD FOR COMPUTING IN ALGEBRAIC NUMBER FIELDS

Jean DELLA DORA, Claire DICRESCENZO, Dominique DUVAL
Groupe de Calcul Formel
(TIM 3 & Institut Fourier)
Grenoble (France)

Algebraic number fields usually appear in computations as $\mathbb{Q}(\alpha_1, \alpha_2, \ldots, \alpha_\ell)$ where $\alpha_1, \alpha_2, \ldots, \alpha_\ell$ runs over the roots of polynomials $P_1(X) \in \mathbb{Q}[X]$, $P_2(X) \in \mathbb{Q}(\alpha_1)[X], \ldots, P_\ell(X) \in \mathbb{Q}(\alpha_1, \ldots, \alpha_{\ell-1})[X]$, where each $P_i(X)$ can be reducible in $\mathbb{Q}(\alpha_1, \ldots, \alpha_{i-1})[X]$. How can the elementary operations $(=, +, -, *, /)$ be performed in such fields ?

Two kinds of methods are generally used :

- Macsyma works in the ring $\mathbb{Q}[X_1, X_2, \ldots, X_\ell]$ modulo $P_1(X_1), P_2(X_1, X_2), \ldots, P_\ell(X_1, \ldots, X_\ell)$; this fits for $+, -, *$, but neither for $=$ nor for $/$.
- The other method consists in factorizing $P_1(X)$ in $\mathbb{Q}[X]$, then $P_2(X)$ in $\mathbb{Q}(\alpha_1)[X]$ where $\mathbb{Q}(\alpha_1) = \mathbb{Q}[X]/\Pi_1(X)$, for each irreducible factor Π_1 of P_1, and so on, in order to determine all the fields on which one has to compute. But algorithms for factorizing are heavy tools, and they require the knowledge of primitive elements for $\mathbb{Q}(\alpha_1, \alpha_2), \ldots, \mathbb{Q}(\alpha_1, \ldots, \alpha_{\ell-1})$.

We don't use any factorization algorithm nor any primitive element computation. We use the following remark of D. Lazard : for any $P(X)$ and $A(X)$ in $\mathbb{Q}[X]$ with $P(X)$ squarefree, $A(X)$ is zero modulo $\gcd(P, A)$ and is invertible modulo $P/\gcd(P, A)$. Instead of computing in the field $\mathbb{Q}[X]/\Pi(X)$ for each irreducible factor Π of P, we compute in $\mathbb{Q}[X]/P(X)$ (the product of all these fields), but we let the internal representation of the elements of $\mathbb{Q}[X]/P(X)$ evolve during the computations. It allows us to define recursively (for $i = 1$ to ℓ) algorithms for the elementary operations on each field $\mathbb{Q}(\alpha_1, \ldots, \alpha_i)$ by working only in their product, that we denote by $\mathbb{Q}\langle P_1, \ldots, P_i \rangle$ (see [D5]).

We could have used any existing algorithm for gcds computations, but we have experienced a method based on the application of Gauss algorithm for matrices to the Sylvester matrix [EG].

The implementation is done in R-Lisp (the Lisp of Reduce) and would be easily translated in any Lisp. An interface with Reduce is provided, and we are also thinking of a modular version.

Our goal is that the programs have to be written on a field $\mathbb{Q}(\alpha_1,\ldots,\alpha_\ell)$, and then, before beeing applied, have to be automatically translated in order to work on $\mathbb{Q}\langle P_1,\ldots,P_\ell\rangle$.

Example : Apply the program "if $a = 0$ then 0 else $\frac{1}{a}$ " to the image of $A(X) = 2X^4 + X^3 + X^2 + X - 1$ modulo $P(X) = 2X^4 - 3X^3 - 3X - 2$;

- Macsyma : $1/4X^3+X^2+4X+1$ (because a is neither 0 nor invertible mod. P) ;
- factorization method : $P(X) = (X^2+1)(2X+1)(X-2)$ (by some factorization algorithm), the answer is : $0 \bmod(X^2+1)$, $-4/5 \bmod(2X+1)$, $1/45 \bmod(X-2)$;
- our method : $\gcd(A,P) = X^2+1$, $P/X^2+1 = 2X^2 - 3X - 2$, the answer is : $0 \bmod(X^2+1)$, $(74X-143)/225 \bmod(2X^2-3X-2)$.

This method may work with any "computable" field instead of $\mathbb{Q}$. Among its applications, let us just mention :

- solving systems of polynomial equations (in complement to standard bases methods) ;
- locally solving (systems of) linear differential or difference equations ;
- computing on algebraic curves (an essential point for integration) ;
- "absolute" factorization of multivariate polynomials ;
- algebraic codes ...

REFERENCES.

[D5] Jean DELLA-DORA, Claire DICRESCENZO, Dominique DUVAL, Computing in algebraic number fields (to appear).

[EG] Jean DELLA-DORA, Claire DICRESCENZO, Dominique DUVAL, Euclidean results using Gauss algorithm (to appear).