MATHEMATICS OF COMPUTATION
VOLUME 48, NUMBER 177
JANUARY 1987, PAGES 309–314

On Computing Isomorphisms of Equation Orders

By M. Pohst

Dedicated to Professor D. Shanks on his 70th birthday

Abstract. A number-geometric method for computing isomorphisms of algebraic number fields (respectively, $\mathbb{Z}$-orders of such fields) is developed. Its main advantage is its easy implementation and moderate computation time.

1. Introduction. In this paper we suggest a number-geometric method for deciding whether for algebraic integers ρ, σ the orders $\mathbb{Z}[\rho]$ and $\mathbb{Z}[\sigma]$ are isomorphic. In case the answer is affirmative, an isomorphism is constructed explicitly. The only assumption is that the minimal polynomials $m_\rho(t)$, $m_\sigma(t) \in \mathbb{Z}[t]$ are known.

Usually this problem is attacked by factoring $m_\sigma(t)$ in $\mathbb{Q}(\rho)[t]$ and testing whether $m_\sigma(t)$ has a linear factor in $\mathbb{Z}[\rho][t]$. Though this procedure seems to be somewhat lengthy, it is polynomial time in the input data [3].

For the number-geometric method a polynomial time behavior is not always guaranteed [2]. But usually applications are made to polynomials m_ρ, m_σ of small degree n (say, $n \leqslant 20$) and not too large coefficients, so that O-estimates are not very meaningful anyway. (Actually, the computation time was very moderate in typical examples of Section 3.) Moreover, the number-geometric method is by far easier to implement than the factorization procedure. Hence, it will be of help to all number theoreticians who have no access to a factorization procedure for polynomials over number fields.

2. The Method. Let $F = \mathbb{Q}(\rho)$ be an algebraic number field of degree $n = \deg(m_\rho)$, where the minimal polynomial $m_\rho(t)$ of ρ is in $\mathbb{Z}[t]$. For $\beta \in F$ we denote the corresponding conjugates by $\beta^{(1)}, \ldots, \beta^{(n)}$ and set

$$T_2(\beta) := \sum_{j=1}^{n} |\beta^{(j)}|^2. \tag{1}$$

If we represent β by means of the $\mathbb{Q}$-basis $1, \rho, \ldots, \rho^{n-1}$ of F in the form $\beta = \sum_{i=1}^{n} \beta_i \rho^{i-1}$, then $T_2(\beta)$ becomes a positive-definite quadratic form

$$Q(\mathbf{\beta}) = \mathbf{\beta} A \mathbf{\beta}^t \tag{2}$$

in the coefficients $\beta_1, \ldots, \beta_n$ ($\mathbf{\beta} := (\beta_1, \ldots, \beta_n)$). We note that in case β is integral, the inequality between arithmetic and geometric means yields $T_2(\beta) \geqslant n$ and $T_2(\beta) = n$ if and only if β is a root of unity.

Received February 28, 1986; revised June 3, 1986.
1980 *Mathematics Subject Classification*. Primary 12-04, 12A20.

0025-5718/87 $1.00 + $.25 per page

PROPOSITION 1. *If the order* $\mathbb{Z}[\sigma]$ *is isomorphic to the order* $\mathbb{Z}[\rho]$, *then* $\mathbb{Z}[\rho]$ *contains an element* β *satisfying* $T_2(\beta) = T_2(\sigma)$.

Proof. Let $\psi\colon \mathbb{Z}[\sigma] \to \mathbb{Z}[\rho]$ be an isomorphism and $\psi(\sigma) = \beta$. Then $m_\sigma(\beta)$ is zero, giving $T_2(\beta) = T_2(\sigma)$. □

PROPOSITION 2. *For* $C \in \mathbb{R}^{>0}$ *the order* $\mathbb{Z}[\rho]$ *contains at most finitely many elements* β *subject to* $T_2(\beta) \leqslant C$.

The proof of Proposition 2 is a consequence of the two subsequent algorithms which determine all solutions $\beta \in \mathbb{Z}[\rho]$ satisfying $T_2(\beta) \leqslant C$. We only present the simplest version of the second algorithm, which is sufficient for most applications. For refinements we refer to [2].

We start by computing the coefficient matrix $A = (a_{ij})$ in $\mathbb{R}^{n\times n}$ of the quadratic form (2). Clearly,

$$a_{ij} = \sum_{k=1}^{n} \rho^{(k)^{i-1}}\overline{\rho^{(k)}}^{j-1} \qquad (1 \leqslant i, j \leqslant n). \tag{3}$$

The a_{ij} are computed as floating-point numbers using good approximations for the zeros of m_ρ. We recommend the use of the subroutine ZPOLR of the IMSL library for that purpose. It calculates $\rho^{(1)}, \ldots, \rho^{(n)}$ of a real polynomial of degree less than 100 up to machine precision. The required computation time is practically negligible. (If all roots of $m_\rho(t)$ are known to be real, a Newton-Maehly-type procedure can also be implemented with ease.)

Then the matrix A is decomposed into the product of a lower- and an upper-triangular matrix similar to Cholesky's method, which has the advantage of being fast and numerically stable.

ALGORITHM A.

Input. A positive-definite matrix $A \in \mathbb{R}^{n\times n}$; $Q(\mathbf{x}) := \mathbf{x}A\mathbf{x}^{\mathrm{tr}}$.

Output. An upper-triangular matrix $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ subject to

$$Q(\mathbf{x}) = \sum_{i=1}^{n} a_{ii}\left(x_i + \sum_{j=i+1}^{n} a_{ij}x_j\right)^2.$$

Steps. For $i = 1, 2, \ldots, n-1$ set: $a_{ji} \leftarrow a_{ij}$, $a_{ij} \leftarrow a_{ij}/a_{ii}$ $(i+1 \leqslant j \leqslant n)$ and for $k = i+1, \ldots, n$ set: $a_{kl} \leftarrow a_{kl} - a_{ki}a_{il}$ $(k \leqslant l \leqslant n)$.

We refer to that procedure of transforming a positive-definite quadratic form Q into a sum of squares as *quadratic completion*.

Now all $\mathbf{x} \in \mathbb{Z}^n$ subject to $Q(\mathbf{x}) \leqslant C$ are easily computed by Algorithm B [2].

ALGORITHM B.

Input. Entries a_{ij} $(1 \leqslant i \leqslant j \leqslant n)$ of the output of Algorithm A and a positive constant C.

Output. All $\mathbf{x} \in \mathbb{Z}^n$ subject to $\mathbf{x} \neq 0$ and $Q(\mathbf{x}) \leqslant C$, as well as the values $Q(\mathbf{x})$ for the quadratic form Q of (2).

Step 1. (Initialization) Set $i \leftarrow n$, $T_i \leftarrow C$, $U_i \leftarrow 0$.

Step 2. (Bounds for x_i) Set $S \leftarrow (T_i/a_{ii})^{1/2}$, $UBx_i \leftarrow \lfloor S - U_i \rfloor$, $x_i \leftarrow \lceil -S - U_i \rceil - 1$.

Step 3. (Increase x_i) Set $x_i \leftarrow x_i + 1$. If $x_i \leqslant UBx_i$ go to 5, else to 4.

Step 4. (Increase i) Set $i \leftarrow i + 1$ and go to 3.

Step 5. (Decrease i) If $i = 1$ go to 6; else set

$$i \leftarrow i - 1, \quad U_i \leftarrow \sum_{j=i+1}^{n} a_{ij}x_j, \quad T_i \leftarrow T_{i+1} - a_{i+1,i+1}(x_{i+1} + U_{i+1})^2$$

and go to 2.

Step 6. ($\mathbf{x}$ valid?) For $\mathbf{x} = \mathbf{0}$ terminate; else print $\mathbf{x}$, $-\mathbf{x}$ and $Q(\mathbf{x}) = C - (T_1 - a_{11}(U_1 + x_1)^2)$ and go to 3.

REMARKS. (i) The solution set of Algorithm B can be diminished by eliminating those $\mathbf{x}$ for which all coordinates, except the first one, are zero since the corresponding element of $\mathbb{Z}[\rho]$ is already in $\mathbb{Z}$.

(ii) By a slight modification of Steps 2, 6 the algorithm determines all $\mathbf{x} \in \mathbb{Z}^n$ subject to $Q(\mathbf{x}) = C$. Since this change only affects the inner loop $i = 1$, it does not yield a considerable decrease of the computation time, however.

According to the proof of Proposition 1 there is an isomorphism between $\mathbb{Z}[\rho]$ and $\mathbb{Z}[\sigma]$ if the output of Algorithm B (for $C = T_2(\sigma)$) contains a vector $\mathbf{x}$ such that for

$$\beta := \sum_{i=1}^{n} x_i\rho^{i-1} \tag{4}$$

the characteristic polynomial of β coincides with $m_\sigma(t)$. But the characteristic polynomial $f_\beta(t) = t^n + c_1t^{n-1} + \cdots + c_n \in \mathbb{Z}[t]$ of β is easily computed upon calculating the power sums

$$S_k := \sum_{j=1}^{n} \beta^{(j)^k} \qquad (1 \leqslant k \leqslant n) \tag{5}$$

and applying Newton's relations:

$$c_k = \frac{1}{k}\sum_{i=0}^{k-1} (-1)^{i+k} c_i S_{k-i} \qquad (1 \leqslant k \leqslant n;\ c_0 := 1). \tag{6}$$

Though this method of determining isomorphisms is very easy, we need to discuss some of its details. Clearly, the running time of Algorithm B strongly depends on the size of C. If $T_2(\sigma)$ is large, it is therefore advisable to compute a reduced basis $\omega_1 = 1, \omega_2, \ldots, \omega_n$ of $\mathbb{Z}[\sigma]$ over $\mathbb{Z}$ and to construct epimorphisms of $\mathbb{Z}[\rho]$ onto $\mathbb{Z}[\omega_i]$ $(2 \leqslant i \leqslant n)$ until $\mathbb{Q}(\omega_2, \ldots, \omega_i) = \mathbb{Q}(\sigma)$. Appropriate reduction methods are discussed in [6]; usually LLL-reduction [3] is the best choice, though pair reduction is easier to implement. Obviously, the case of $\mathbb{Q}(\sigma)$ being a primitive extension of $\mathbb{Q}$ is favorable to our method, contrary to other methods.

Finally, we note that the suggested number-geometric method is also suitable for constructing isomorphisms between number fields, or orders of number fields, after slight modifications.

3. Error Analysis. If the minimal polynomial

$$m_\rho(t) = t^n + a_1t^{n-1} + \cdots + a_n \in \mathbb{Z}[t] \tag{7}$$

has nonreal roots, the entries a_{ij} of (3) are algebraic integers which are, in general, not rational. Though the method developed in Section 2 seems to work without problems also in that case, an analysis of round-off errors is certainly required if we

want to prove that no isomorphism exists. (In the preceding section we assumed that this is the case if Algorithm B produces no output.) In the sequel we therefore discuss possible round-off errors for all computations with floating-point numbers.

In the first step we calculate all zeros of $m_\rho(t)$ up to machine precision; in our case 14 digits. Known a priori estimates for the size of those zeros are:

$$|\rho^{(k)}| \leqslant \min\Big(\max\{|a_n|, 1+|a_{n-1}|, \dots, 1+|a_1|\}, \\ 2\max\{|a_i|^{1/i} \mid 1 \leqslant i \leqslant n\}\Big) \tag{8}$$

for all zeros $\rho^{(k)}$ $(1 \leqslant k \leqslant n)$ of $m_\rho(t)$. Hence, for approximate solutions $\tilde{\rho}^{(k)}$ $(\rho^{(k)} = \tilde{\rho}^{(k)} + \varepsilon,\ |\varepsilon| \sim 10^{-14})$ we obtain

$$\rho^{(k)^{i-1}} \sim \tilde{\rho}^{(k)^{i-1}} + (i-1)\tilde{\rho}^{(k)^{i-2}}\varepsilon =: \tilde{\rho}^{(k)^{i-1}} + \varepsilon_{k,i} \qquad (2 \leqslant i \leqslant n-1) \tag{9}$$

and, setting $\varepsilon_{k,1} = 0$ $(1 \leqslant k \leqslant n)$,

$$\begin{aligned} a_{ij} &= \sum_{k=1}^{n} \left(\tilde{\rho}^{(k)^{i-1}} + \varepsilon_{k,i}\right)\left(\overline{\tilde{\rho}^{(k)^{j-1}} + \varepsilon_{k,j}}\right) \\ &\sim \sum_{k=1}^{n} \tilde{\rho}^{(k)^{i-1}}\overline{\tilde{\rho}^{(k)^{j-1}}} + \sum_{k=1}^{n}\left(\varepsilon_{k,i}\overline{\tilde{\rho}^{(k)^{j-1}}} + \overline{\varepsilon_{k,j}}\tilde{\rho}^{(k)^{i-1}}\right) \\ &=: \sum_{k=1}^{n} \tilde{\rho}^{(k)^{i-1}}\overline{\tilde{\rho}^{(k)^{j-1}}} + \delta_{ij}. \end{aligned} \tag{10}$$

We note that $\delta_{ij} = 0$ for $i = 1$ or $j = 1$, and that the worst error generally occurs for $i = j = n$.

Example. By reducing ρ—if necessary—we can often obtain $|\rho^{(k)}| \leqslant 3$ for $n \leqslant 8$ and, therefore, $\varepsilon_{k,n-1}$ of size at most $(n-1)3^{n-2}|\varepsilon|$. Hence, for $n \leqslant 8$ we find $|\delta_{nn}| < 2 \cdot 10^{-6}$.

Remark. The huge increase of potential round-off errors for the powers of $\rho^{(k)}$ requires refined strategies for higher dimensions n. We only mention three possible improvements. The most obvious one is the computation of $\rho^{(k)}$ to higher precision. A little more difficult is the computation of $m_{\rho^i}(t)$ for suitable exponents i (for example, $i = 2, 4, \dots$) and the calculation of the zeros of these polynomials. We note that for

$$m_\rho(t) = \sum_{\substack{i=0 \\ n-i \equiv 0(2)}}^{n} a_i t^{n-i} + \sum_{\substack{i=0 \\ n-i \equiv 1(2)}}^{n} a_i t^{n-i} \qquad (a_0 := 1)$$

we obtain

$$m_{\rho^2}(t) = \left(\sum_{\substack{i=0 \\ n-i \equiv 0(2)}}^{n} a_i t^{(n-i)/2}\right)^2 - \left(\sum_{\substack{i=0 \\ n-i \equiv 1(2)}}^{n} a_i t^{(n-i)/2}\right)^2.$$

This observation also suggests to compute the zeros by Graeffe's method. Another safe way is to work with interval arithmetic.

In analyzing Algorithm A we note that we can obtain $|a_{il}| < a_{ii}$ $(i+1 \leqslant l \leqslant n)$ at each step by a suitable permutation of the coordinates of $\mathbf{x}$. This is because the quadratic form under consideration is positive definite, which implies $a_{ii}a_{ll} > a_{il}^2$

$(1 \leqslant i, l \leqslant n)$. The round-off error in calculating a_{il}/a_{ii} is therefore at most $(|\delta_{ii}| + |\delta_{il}|)/a_{ii}$. Hence the computation of a_{kl} is affected by an error less than $|\delta_{kl}| + |\delta_{ki}| + |\delta_{ii}| + |\delta_{il}|$, hence bounded by $4^{n-1}|\delta_{nn}|$.

Example (continued). Under the assumptions made above, the largest round-off error for any a_{kl} is bounded by $3 \cdot 10^{-2}$.

Knowing upper bounds for the round-off errors of the a_{ij} of the input of Algorithm B, it is now easy to compute correct bounds for each coordinate $x_i \in \mathbb{Z}$, i.e., bounds which are large enough to guarantee that all solutions of the original problem will be among the output of that algorithm.

We note that the estimates given in the example are very rough; usually much better estimates are obtained by calculating upper bounds for the δ_{ij} explicitly.

Finally, the values S_k of (5) are integers and can be easily computed from the (integers) $\mathrm{Tr}(\rho^{i-1})$ $(1 \leqslant i \leqslant n)$ by integral arithmetic, and this also holds for the computation of the numbers c_k of (6). (All these calculations are based on $m_\rho(\rho) = 0$, of course.)

4. Applications and Examples. (i) According to Zimmer [8, pp. 53, 54], Hasse came up with the problem of whether zeros of the polynomials

$$f_1(t) = t^5 - t^3 - 2t^2 - 2t - 1, \qquad f_2(t) = t^5 - 2t^4 + 2t^3 - 3t^2 + 6t - 5,$$

$$f_3(t) = t^5 - t^4 + t^3 + t^2 - 2t + 1$$

generate isomorphic fields. A solution was given by Zassenhaus and Liang [7]. The application of the method of Section 2 for $m_\rho(t) = f_1(t)$ and $C = 11$ (an upper bound for $T_2(\beta)$, β a zero of $f_2(t)$ or $f_3(t)$) yields—up to sign—22 short vectors. Among them, $\sigma_2 = \rho^2 - \rho$ and $\sigma_3 = -\rho^4 + \rho^3 + \rho + 1$ satisfy $f_i(\sigma_i) = 0$ $(i = 2, 3)$.

(ii) In general, there are not many short vectors among the integers of an algebraic number field, even if the discriminant is large. We list the number of pairs of vectors N of the output of Algorithm B, as well as the computation time t in seconds, of three nth root fields:

ρ	$-23^{1/5}$	$-12^{1/9}$	$-5^{1/11}$
C	17.07	15.64	14.74
N	1	1	1
t	0.006	0.009	0.011.

(iii) Searching for algebraic number fields with discriminants of small absolute value usually yields a couple of equations for which it must then be decided whether their zeros generate isomorphic fields. The results below are closely related to the determination of the minimum discriminant for fixed degree $n = s + 2t$, s the number of real, $2t$ the number of complex conjugates. From [5] it is known that an algebraic number field F of degree n and discriminant d_F contains an irrational integer β subject to

$$T_2(\beta) \leqslant \frac{n}{4} + \left(\frac{\gamma_{n-1}^{n-1} |d_F|}{n} \right)^{1/(n-1)}, \tag{11}$$

where γ_n^n denotes Hermite's constant for positive-definite quadratic forms. In our list, N_1 is the number of polynomials $h(t)$ obtained by a minimum discriminant procedure, N_2 the number of polynomials $g(t)$ determined by our method for the input data $m_\rho(t)$ and C. Here, C is either $T_2(\rho)$ or obtained from (7) for the value of the minimum discriminant. N_2 has to be expected to be larger than N_1 because of the missing side condition $0 \leqslant -\mathrm{Tr}(\beta) \leqslant n/2$. In each case, all polynomials $h(t)$ were among the $g(\pm t)$, and therefore the corresponding fields are isomorphic.

The listed computation times refer to the calculation of the zeros of $m_\rho(t)$ (Step 1), Algorithm A (Step 2), and Algorithm B (Step 3).

Examples from [1], [4], [5].

	Coeff. $a_1, \ldots, a_n$		CPU-time in seconds				
$n = s + 2 \cdot t$	of $m_\rho(t)$	C	Step 1	Step 2	Step 3	N_1	N_2
$6 = 6 + 2 \cdot 0$	1, −7, −2, 7, 2, −1	14.01	0.002	0.003	0.002	0	0
$6 = 4 + 2 \cdot 1$	1, −2, −3, −1, 2, 1	11.94	0.002	0.003	0.009	9	11
$6 = 2 + 2 \cdot 2$	2, 0, −3, 0, 2, −1	9.72	0.002	0.003	0.009	5	10
$6 = 0 + 2 \cdot 3$	0, 1, 1, −2, −1, 1	8.10	0.001	0.004	0.015	6	18
$7 = 7 + 2 \cdot 0$	1, −6, −5, 8, 5, −2, −1	21.10	0.003	0.004	0.020	12	21
$7 = 3 + 2 \cdot 2$	0, 0, 0, −4, 0, 3, 1	13.03	0.002	0.004	0.023	19	24

We note that $C = 14.01$ is too small to obtain a generating polynomial in case $n = s = 6$. This is due to the presence of subfields (see [5]). The reason for the discrepancy between N_1, N_2 in the cases $n = 6$, $s = 0$, 2 is that several roots of a polynomial $g(t)$ belong to $\mathbb{Z}[\rho]$.

All computations were carried out on the CDC Cyber 76 of the Rechenzentrum der Universität zu Köln.

Mathematisches Institut der Universität Düsseldorf
Universitätsstrasse 1
4 Düsseldorf, Federal Republic of Germany

1. F. Diaz y Diaz, Private communication to the author.

2. U. Fincke & M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Math. Comp.*, v. 44, 1985, pp. 463–471.

3. A. K. Lenstra, H. W. Lenstra, Jr. & L. Lovász, "Factoring polynomials with rational coefficients," *Math. Ann.*, v. 261, 1982, pp. 515–534.

4. M. Pohst, "The minimum discriminant of seventh degree totally real algebraic number fields," in *Number Theory and Algebra* (H. Zassenhaus, ed.), Academic Press, New York, 1977, pp. 235–240.

5. M. Pohst, "On the computation of number fields of small discriminants including the minimum discriminants of sixth degree fields," *J. Number Theory*, v. 14, 1982, pp. 99–117.

6. M. Pohst & H. Zassenhaus, *Methods and Problems of Computational Algebraic Number Theory*, Cambridge Univ. Press. (To appear.)

7. H. Zassenhaus & J. Liang, "On a problem of Hasse," *Math. Comp.*, v. 23, 1969, pp. 515–519.

8. H. G. Zimmer, *Computational Problems, Methods, and Results in Algebraic Number Theory*, Lecture Notes in Math., vol. 262, Springer-Verlag, Berlin and New York, 1972.