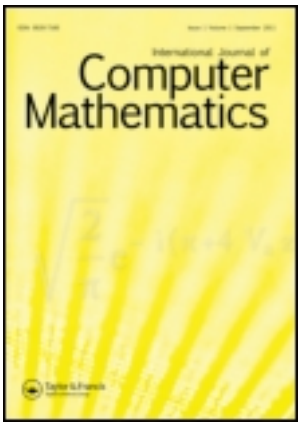


International Journal of Computer Mathematics

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/gcom20

# A complete algorithm to find exact minimal polynomial by approximations

Xiaolin Qin [a b c], Yong Feng [a], Jingwei Chen [a] & Jingzhong Zhang [a]

[a] Laboratory for Automated Reasoning and Programming, Chengdu Institute of Computer Applications, CAS, Chengdu, 610041, China

[b] Department of Mathematics, Sichuan University, Chengdu, 610064, China

[c] Graduate School of the Chinese Academy of Sciences, Beijing, 100049, China

Accepted author version posted online: 02 Aug 2012.Version of record first published: 29 Aug 2012.

To cite this article: Xiaolin Qin , Yong Feng , Jingwei Chen & Jingzhong Zhang (2012): A complete algorithm to find exact minimal polynomial by approximations, International Journal of Computer Mathematics, 89:17, 2333-2344

To link to this article: http://dx.doi.org/10.1080/00207160.2012.716199

*International Journal of Computer Mathematics*
Vol. 89, No. 17, November 2012, 2333–2344

Taylor & Francis
Taylor & Francis Group

# A complete algorithm to find exact minimal polynomial by approximations

Xiaolin Qin[a,b,c]*, Yong Feng[a], Jingwei Chen[a] and Jingzhong Zhang[a]

[a]*Laboratory for Automated Reasoning and Programming, Chengdu Institute of Computer Applications, CAS, Chengdu 610041, China;* [b]*Department of Mathematics, Sichuan University, Chengdu 610064, China;* [c]*Graduate School of the Chinese Academy of Sciences, Beijing 100049, China*

(*Received 5 January 2011; revised version received 3 January 2012; accepted 20 July 2012*)

Based on an improved parameterized integer relation construction method, a complete algorithm is proposed for finding an exact minimal polynomial from its approximate root. It relies on a study of the error controlling for its approximation. We provide a sufficient condition on the precision of the approximation, depending only on the degree and the height of its minimal polynomial. Our result is superior to the existent error controlling on obtaining an exact rational or algebraic number from its approximation. Moreover, some applications are presented and compared with the subsistent methods.

**Keywords:** error controllable algorithm; symbolic–numerical computation; integer relation construction; minimal polynomial

*2010 AMS Subject Classifications*: 41A52; 65D99; 68W30



## 1. Introduction

Symbolic–numerical computation is a novel method for solving large-scale problems, which applies both numerical and symbolic methods in its algorithms and provides a new perspective of them. Recently, the exact computation by intermediate of floating-point arithmetic has been an active area of solving the problem of intermediate expression swell in [6,8,9,15,18,20,36]. It can be considered as an error controllable algorithm in [24,27,33]. Of course, it is also a challenge in symbolic and scientific computing [14,29,31,32]. The aim of this article is to provide a rigorous and efficient algorithm to reconstruct an exact minimal polynomial from its approximate root.

Consider the interesting question: suppose we are given an approximate root of an unknown minimal polynomial with integral coefficients, and two bounds on the degree and size of the coefficients of the minimal polynomial. Is it possible to infer the exact minimal polynomial? The question was raised by Manuel Blum in Theoretical Cryptography [17, p. 243]. Moreover, in the reference [34, p. 119], the authors indicated that symbolic computation is substituted

*Corresponding author. Email: qinxl@casit.ac.cn

ISSN 0020-7160 print/ISSN 1029-0265 online

http://dx.doi.org/10.1080/00207160.2012.716199
http://www.tandfonline.com

for numerical computation in order to reduce memory consumption, which is a practical and interesting numerical method. Kannan *et al.* (KLL) answered the question in [16,17]. However, their technique is based on the Lenstra-Lenstra-Lovasz (LLL) lattice reduction algorithm, which is quite unstable in numerical computations [4]. In [13], Just *et al*. presented an algorithm for finding an integer relation on $n$ real numbers using the LLL-lattice basis reduction technique, which needed the high precision. The built-in function *PolynomialTools:-MinimalPolynomial()* in *Maple*, which finds the minimal polynomial for an approximate root, was implemented using the same technique (see *Maple*'s help). Fieker and Friedrichs [11] used integral LLL-reduction to reconstruct a solution by modulo a suitable ideal. Their approach did not involve how to obtain exact minimal polynomial by approximation. Moreover, Chèze and Galligo (CG) applied Number Theory techniques and provided sharp bounds to obtain exact absolute polynomial factorization from an approximate factor in [5,6]. However, their results rely on finding a primitive element, the procedure of which is very complicated. In this article, we propose an efficient approach to remedy these drawbacks.

In this article, a new algorithm is presented for finding an exact minimal polynomial from its approximation. It is based on the improved parameterized integer relation construction algorithm PSLQ($\tau$) [12], whose stability admits an efficient implementation with lower run times on average than the existing algorithms, and can be used to prove that relation bounds obtained from computer runs using it are numerically accurate. Based on the PSLQ, one can find algebraic relations, such as [1–4], whereas these literatures did not involve the minimal polynomial finding in detail. The other function *identify* in *Maple*, which finds a closed form for a decimal approximation of a number, was implemented using the integer relation construction algorithm. However, the choice of *Digits* of approximate value is fairly arbitrary [3]. In contrast, we fully analyse numerical behaviour of an approximate to exact value and give the number of *Digits* of approximate value, which are required to exact results. The work is regarded as further research of [35]. Solving the problem can be described as follows.

Given an approximate value $\tilde{\alpha}$ at arbitrary accuracy of an unknown algebraic number, and obtaining the upper bound degree $n$ of the algebraic number and an upper bound $N$ of its height on minimal polynomial in advance, the problem will be solved in two steps. First, we discuss how much the error $\varepsilon$ can be, so that we can reconstruct the algebraic number $\alpha$ from its approximation $\tilde{\alpha}$ with $|\alpha - \tilde{\alpha}| < \varepsilon$. Of course, $\varepsilon$ is a function in $n$ and $N$. Second, we give an algorithm to compute the minimal polynomial of the algebraic number.

Our method can be generalized to transcendental numbers of the forms $\sin^{-1}(\alpha)$, $\log(\alpha)$, etc., where $\alpha$ is algebraic. We also propose a simple polynomial-time algorithm to factor multivariate polynomials with rational coefficients, and provide a natural, efficient technique to the minimal polynomial representation. The basic idea is taken from [17]. However, the efficiency of our method is improved greatly.

This article is the final journal version of [25], which adds to the main contents as follows: re-proves the main results, analyses the complexity of our algorithm, compares with four different algorithms, and extends the applications of our method.

Our main contributions in this article are the following. Based on the remarkable parameterized integer relation construction algorithm, we have completely solved the problem of reconstruction of the general real algebraic numbers, including the unique conditions and the upper bound of error controlling, and propose an algorithm to obtain exact minimal polynomial from its approximate value. The algorithm is efficient and numerically stable. Moreover, some applications are presented.

The rest of this article is organized as follows. Section 2 illustrates the improved parameterized integer relation construction algorithm. Section 3 discusses how to reconstruct minimal polynomial (RMP), and gives some applications and small examples in detail. Section 4 gives some experimental results. The final section concludes this article.

## 2. Preliminaries

In this section, we first give some notations, and a brief introduction on integer relation problems. After that an improved parameterized integer relation construction algorithm is also reviewed.

### 2.1 *Notations*

Throughout this article, $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$ denote the set of the integers, rationals, and reals, respectively. $\mathbb{O}(\mathbb{R}^n)$ is the corresponding system of ordinary reals, $U(n,\mathbb{Z})$ the group of unitary matrices over $\mathbb{Z}$, $\mathrm{GL}(n,\mathbb{Z})$ the group of unimodular matrices with entries in the $\mathbb{Z}$. For $c \in \mathbb{R}$, $\lfloor c \rceil = \lfloor c + \frac{1}{2} \rfloor$. The ring of polynomials with integral coefficients will be denoted as $\mathbb{Z}[x]$. The *content* of a polynomial $p(x)$ in $\mathbb{Z}[x]$ is the greatest common divisor of its coefficients. A polynomial in $\mathbb{Z}[x]$ is *primitive* if its content is 1. A polynomial $p(x)$ has degree $d$ if $p(x) = \sum_{i=0}^{d} p_i x^i$ with $p_d \neq 0$. The *length* $|p|$ of $p(x) = \sum_{i=0}^{d} p_i x^i$ is the Euclidean length of the vector $(p_0, p_1, \ldots, p_d)$; the *height* $|p|_\infty$ of $p(x)$ is the $L_\infty$-norm of the vector $(p_0, p_1, \ldots, p_d)$, so $|p|_\infty = \max_{0 \le i \le d} |p_i|$. An *algebraic number* is a root of a polynomial with integral coefficients. The minimal polynomial of an algebraic number $\alpha$ is the irreducible polynomial in $\mathbb{Z}[x]$ satisfied by $\alpha$. The minimal polynomial is unique up to units in $\mathbb{Z}$. The *degree* and *height* of an algebraic number are the degree and height of its minimal polynomial, respectively.

### 2.2 *Integer relation algorithm*

There exists an integer relation amongst the numbers $x_1, x_2, \ldots, x_n$ if there are integers $a_1, a_2, \ldots, a_n$, not all zero, such that $\sum_{i=1}^{n} a_i x_i = 0$. For the vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]^{\mathrm{T}}$, the nonzero vector $\mathbf{a} = [a_1, a_2, \ldots, a_n] \in \mathbb{Z}^n$ is an integer relation for $\mathbf{x}$ if $\mathbf{a} \cdot \mathbf{x} = 0$.

In order to introduce the integer relation algorithm, we recall some useful definitions and theorems [4,12]:

DEFINITION 2.1 *For simplicity, defined by the vector $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^{\mathrm{T}} \in \mathbb{R}^n$ for the rest of this article, and all the bold variates are represented to the vectors. Define $\boldsymbol{x}^{\perp}$ to be the set of all vectors in $\mathbb{R}^n$ orthogonal to $\boldsymbol{x}$.*

DEFINITION 2.2 *($M_{\boldsymbol{x}}$) Assume $\boldsymbol{x}$ has norm $|\boldsymbol{x}| = 1$. Let $\mathbb{O}(\mathbb{R}^n) \cap \boldsymbol{x}^{\perp}$ be the discrete lattice of integral relations for $\boldsymbol{x}$. Define $M_{\boldsymbol{x}} > 0$ to be the smallest norm of any relation for $\boldsymbol{x}$ in this lattice.*

DEFINITION 2.3 *($H_{\boldsymbol{x}}$) Assume $\boldsymbol{x}$ has norm $|\boldsymbol{x}| = 1$. Furthermore, suppose that no coordinate entry of $\boldsymbol{x}$ is zero, that is, $x_j \neq 0$ for $1 \le j \le n$(otherwise $\boldsymbol{x}$ has an immediate and obvious integral relation). For $1 \le j \le n$ define the partial sums*

$$s_j^2 = \sum_{j \le k \le n} x_k^2.$$

*Given such a unit vector $\boldsymbol{x}$, define the $n \times (n-1)$ lower trapezoidal matrix $H_{\boldsymbol{x}} = (h_{i,j})$ by*

$$h_{i,j} = \begin{cases} 0 & \text{if } 1 \le i < j \le n-1, \\ \dfrac{s_{i+1}}{s_i} & \text{if } 1 \le i = j \le n-1, \\ \dfrac{-x_i x_j}{(s_j s_{j+1})} & \text{if } 1 \le j < i \le n. \end{cases}$$

*Note that $h_{i,j}$ is scale invariant.*

THEOREM 2.4 *Let $\boldsymbol{x} \neq 0 \in \mathbb{R}^n$. Suppose that for any relation $\boldsymbol{m}$ of $\boldsymbol{x}$ and for any matrix $A \in \mathrm{GL}(n, \mathbb{Z})$, there exists a unitary matrix $Q \in U(n-1, \mathbb{Z})$ such that $H = AH_{\boldsymbol{x}}Q$ is lower trapezoidal and all of the diagonal elements of $H$ satisfy $h_{j,j} \neq 0$. Then,*

$$\frac{1}{\max_{1 \leq j \leq n-1} |h_{j,j}|} = \min_{1 \leq j \leq n-1} \frac{1}{|h_{j,j}|} \leq |\boldsymbol{m}|.$$

*Proof* See Theorem 1 of [12]. ■

*Remark 2.5* The inequality of Theorem 2.4 offers an increasing lower bound on the size of any possible relation. Theorem 2.4 can be used with any algorithm that produces $\mathrm{GL}(n, \mathbb{Z})$ matrices. Any $\mathrm{GL}(n, \mathbb{Z})$ matrix $A$ whatsoever can be put into Theorem 2.4.

THEOREM 2.6 *Assume real numbers, $n \geq 2, \tau > 1, \gamma > 2/\sqrt{3}$, and that $0 \neq \boldsymbol{x} \in \mathbb{R}^n$ have integer relations. Let $M_{\boldsymbol{x}}$ be the least norm of relations for $\boldsymbol{x}$. Then,* PSLQ($\tau$) *will find some integer relation for $\boldsymbol{x}$ in no more than*

$$\binom{n}{2} \frac{\log(\gamma^{n-1} M_{\boldsymbol{x}})}{\log \tau}$$

*iterations.*

*Proof* See Theorem 2 of [12]. ■

THEOREM 2.7 *Let $M_{\boldsymbol{x}}$ be the smallest possible norm of any relation for $\boldsymbol{x}$. Let $\boldsymbol{m}$ be any relation found by* PSLQ($\tau$). *For all $\gamma > 2/\sqrt{3}$*

$$|\boldsymbol{m}| \leq \gamma^{n-2} M_{\boldsymbol{x}}.$$

*Proof* See Theorem 3 of [12]. ■

*Remark 2.8* For $n = 2$, Theorem 2.7 proves that any relation $0 \neq \mathbf{m} \in \mathbb{O}(\mathbb{R}^2)$ found has norm $|\mathbf{m}| = M_{\mathbf{x}}$. In other words, PSLQ($\tau$) finds a shortest relation. For real numbers this corresponds to the case of the Euclidean algorithm.

Theorem 2.4 suggests a strategy to construct a relation finding algorithm. The key step is to find a way to reduce the norm of the matrix $H_{\mathbf{x}}$ by multiplication by some unimodular A on the left. The following modified Hermite reduction can achieve it.

ALGORITHM 2.1 Modified Hermite reduction

***Input:*** *a lower trapezoidal matrix $H_{\mathbf{x}} = (h_{i,j}) \in \mathbb{R}^{n \times (n-1)}$ with $h_{j,j} \neq 0$.*
***Output:*** *a reducing matrix $D$ of $H_{\mathbf{x}}$.*

```
(1) D := I_n
(2) for i from 2 to n do
(3)     for j from i − 1 by −1 to 1 do
(4)         q := ⌊h_{i,j}/h_{j,j}⌉
(5)         for k from 1 to j do
(6)             h_{i,k} := h_{i,k} − q h_{j,k}
(7)         for k from 1 to n do
(8)             d_{i,k} := d_{i,k} − q d_{j,k}
(9) return the n × n matrix D.
```

If Algorithm 2.1 outputs $D$ for an $n \times (n-1)$ matrix $H_{\mathbf{x}}$, we say that $DH_{\mathbf{x}}$ is the modified Hermite reduction of $H_{\mathbf{x}}$ and that $D$ is the reducing matrix of $H_{\mathbf{x}}$. This reduction develops the left multiplying modified Hermite reducing matrix $D$.

Hermite reduction is also presented (see [12], Definition 3), and is equivalent to modified Hermite reduction for a lower triangular matrix $H_{\mathbf{x}}$ with $h_{j,j} \neq 0$ (see [12, Lemma 3]). Both equivalent reductions have the following properties (see [12, Lemma 4]):

(i) The reducing matrix $D \in \mathrm{GL}(n, \mathbb{Z})$.
(ii) For all $k > i$, the modified Hermite reduced matrix $H' = (h'_{i,j}) = DH_{\mathbf{x}}$ satisfies $|h'_{k,i}| \leq |h'_{i,i}|/2 = |h_{i,i}|/2$.

Based on Algorithm 2.1 and the theorems above, and the existence of a parameter $\gamma > 2/\sqrt{3}$, an algorithm for obtaining the integer relation can be designed as follows:

ALGORITHM 2.2 Parameterized integer relation construction

***Input:*** $(x_1, x_2, \ldots, x_n) = \boldsymbol{x} \in \mathbb{R}^n$, *and the parameter* $\gamma > 2/\sqrt{3}$.
***Output:*** *either output an integer relation for* $\boldsymbol{x}$ *or give a lower bound* $N$ *on.*

(1) *Initiation. Compute the hyperplane matrix* $H_{\mathbf{x}}$ *by Definition 2.3, set* $H := H_{\mathbf{x}}, A := I_n, B := I_n$.
(2) *Reduction. Call Algorithm 2.1 to reduce* $H$ *producing the reducing matrix* $D \in \mathrm{GL}(n, \mathbb{Z})$. *Set* $\mathbf{x} := \mathbf{x}D^{-1}, H := DH, A := DA, B := BD^{-1}$.
(3) ***loop***
(4) *Exchange. Let* $H = (h_{i,j})$. *Choose an integer* $r$ *such that* $\gamma^r |h_{r,r}| \geq \gamma^i |h_{i,i}|$ *for all* $1 \leq i \leq n-1$. *Define the permutation matrix* $R$ *to be the identity matrix with the* $r$ *and* $r+1$ *rows exchanged. Update* $\mathbf{x} := \mathbf{x}R, H := RH, A := RA, B := BR$.
(5) *Corner. Let*

$$\alpha := h_{r,r}, \quad \beta := h_{r+1,r},$$
$$\lambda := h_{r+1,r+1}, \quad \delta := \sqrt{\beta^2 + \lambda^2}.$$

*Let* $Q := I_{n-1}$. *If* $r < n-1$, *then change the submatrix of* $Q$ *to consist of the* $r$*th and* $(r+1)$*th rows of columns* $r$ *and* $r+1$ *be* $\begin{pmatrix} \beta/\delta & -\lambda/\delta \\ \lambda/\delta & \beta/\delta \end{pmatrix}$.
*Update* $H := HQ$.
(6) *Reduction. Call Algorithm 2.1 to reduce* $H$ *producing* $D$. *Update* $\mathbf{x} := \mathbf{x}D^{-1}, H := DH, A := DA, B := BD^{-1}$.
(7) *Compute* $N := 1/\max_{1 \leq j \leq n-1} |h_{j,j}|$. *Then there exists no integer relation whose Euclidean norm is less than* $N$.
(8) ***if*** $\mathbf{x}_j = 0$ *for some* $1 \leq j \leq n$, *or* $h_{n-1,n-1} = 0$ ***then***
(9) ***return*** *the corresponding integer relation for* $\mathbf{x}$.
(10) ***end loop***

*Remark 2.9* The basic ideas of Algorithms 2.1, 2.2 are from the literature [12]. However, they only gave the definition styles. In this article, we give the equivalent styles and improve the Algorithm 2.2 for introducing a lower bound $N$.

By Algorithm 2.2, we can find the integer relation $\mathbf{m}$ of the vector $\mathbf{x} = (1, \tilde{\alpha}, \tilde{\alpha}^2, \ldots, \tilde{\alpha}^n)$ by error controlling. So, we get a nonzero polynomial of degree $n$, which denotes $G(x)$ for the rest of this article, that is,

$$G(x) = \mathbf{m} \cdot (1, x, x^2, \ldots, x^n)^{\mathrm{T}}. \tag{1}$$

Our main task is to show that polynomial (1) is uniquely determined under assumptions, and discuss the controlling error in Algorithm 2.2 in the next section.

## 3. RMP from its approximation

In this section, we will solve the following problem. Given a floating number $\tilde{\alpha}$, which is an approximation of unknown algebraic number $\alpha$, how do we obtain its exact minimal polynomial? At first, we state some lemmas as follows:

LEMMA 3.1 *Let $f$ be a nonzero polynomial in $\mathbb{Z}[x]$ of degree $n$. If $\varepsilon = \max_{1\le i\le n} |\alpha^i - \tilde{\alpha}^i|$[1], then*

$$|f(\alpha) - f(\tilde{\alpha})| \le \varepsilon \cdot n \cdot |f|_\infty. \tag{2}$$

*Proof* Clear. ■

LEMMA 3.2 *Let $h$ and $g$ be two nonzero polynomials in $\mathbb{Z}[x]$ of degree $n$ and $m$, respectively, and let $\alpha \in \mathbb{R}$ be a zero of $h$ with $|\alpha| \le 1$. If $h$ is irreducible and $g(\alpha) \ne 0$, then*

$$|g(\alpha)| \ge n^{-1} \cdot |h|^{-m} \cdot |g|^{1-n}. \tag{3}$$

*Proof* See Proposition (1.6) of [17]. Without loss of generality suppose that $|\alpha| \le 1$. If $|\alpha| > 1$, the lemma also holds. We only need a simple transformation by substituting $x$ with $1/x$, then do $x^n \cdot h(1/x)$ for a polynomial, the height of which is constant. ■

*Remark 3.3* From Lemma 3.2, we can know the following facts: if $|g(\alpha)| < n^{-1} \cdot |h|^{-m} \cdot |g|^{1-n}$, then $g(\alpha) = 0$.

COROLLARY 3.4 *With the previous notations, if $h$ is irreducible and $g(\alpha) \ne 0$, then*

$$|g(\alpha)| \ge n^{-1} \cdot (n+1)^{-m/2} \cdot (m+1)^{(1-n)/2} \cdot |h|_\infty^{-m} \cdot |g|_\infty^{1-n}. \tag{4}$$

*Proof* First notice that $|f|^2 \le (n+1) \cdot |f|_\infty^2$ holds for any polynomial $f$ of degree at most $n > 0$, so $|f| \le \sqrt{n+1} \cdot |f|_\infty$. From Lemma 3.2, we get

$$|g(\alpha)| \ge n^{-1} \cdot (n+1)^{-m/2} \cdot (m+1)^{(1-n)/2} \cdot |h|_\infty^{-m} \cdot |g|_\infty^{1-n}.$$

This proves Corollary 3.4. ■

LEMMA 3.5 *Let $\tilde{\alpha}$ be an approximate value to an unknown algebraic number $\alpha$ with degree $n > 0$ and $N$ be the upper bound on the height of minimal polynomial of $\alpha$. For any $G(x)$ in $\mathbb{Z}[x]$ with degree $n$, if*

$$|G(\tilde{\alpha})| < n^{-1} \cdot (n+1)^{-n+1/2} \cdot |G|_\infty^{-n} \cdot N^{1-n} - n \cdot \varepsilon \cdot |G|_\infty,$$

*then*

$$|G(\alpha)| < n^{-1} \cdot (n+1)^{-n+1/2} \cdot |G|_\infty^{-n} \cdot N^{1-n}.$$

*Proof* Let $\alpha \in \mathbb{R}$ where $|\alpha| \le 1$. From Lemma 3.1, we notice that $|G(\alpha) - G(\tilde{\alpha})| \le \varepsilon \cdot n \cdot |G|_\infty$, and together with

$$|G(\alpha)| - |G(\tilde{\alpha})| \le |G(\alpha) - G(\tilde{\alpha})|,$$

we get,

$$|G(\alpha)| \le |G(\tilde{\alpha})| + n \cdot \varepsilon \cdot |G|_\infty. \tag{5}$$

From the assumption of the theorem, since

$$|G(\tilde{\alpha})| < n^{-1} \cdot (n+1)^{-n+1/2} \cdot |G|_\infty^{-n} \cdot N^{1-n} - n \cdot \varepsilon \cdot |G|_\infty, \tag{6}$$

combined with Equation (5), we have proved Lemma 3.5. ■

COROLLARY 3.6 *With the previous notations, for any $G(x)$ in $\mathbb{Z}[x]$ with degree $n$, if $|G(\alpha)| < n^{-1} \cdot (n+1)^{-n+1/2} \cdot |G|_\infty^{-n} \cdot N^{1-n}$, then*

$$G(\alpha) = 0. \tag{7}$$

*The primitive part of polynomial $G(x)$ is the minimal polynomial $g(x)$ of algebraic number $\alpha$.*

*Proof by contradiction* The proof of Corollary 3.6 is from the fact, there is a gap for a univariate polynomial which has been assigned constant to variable, that is, there is the lower bound on the known polynomial. Let $\alpha \in \mathbb{R}$ where $|\alpha| \leq 1$. According to Lemma 3.2, we can get that if $G(\alpha) \neq 0$, then

$$|G(\alpha)| \geq n^{-1} \cdot (n+1)^{-n+1/2} \cdot |G|_\infty^{-n} \cdot N^{1-n}.$$

From the assumption of the corollary, we have

$$|G(\alpha)| < n^{-1} \cdot (n+1)^{-n+1/2} \cdot |G|_\infty^{-n} \cdot N^{1-n}.$$

However, it leads to a contradiction. So, $G(\alpha) = 0$.

Let $G(x) = \sum_{i=0}^{n} a_i x^i$, which is constructed by the parameterized integer relation construction algorithm from the vector $\mathrm{x} = (1, \alpha, \alpha^2, \ldots, \alpha^n)$. Since algebraic number $\alpha$ with degree $n > 0$, according to the definition of minimal polynomial, then the primitive polynomial of $G(x)$, denoted by $pp(G(x))$. Hence, $pp(G(x))$ is just irreducible and equal to $g(x)$. Of course, it is unique.

This proves Corollary 3.6. ■

### 3.1 *Obtaining minimal polynomial by approximation*

If $\alpha$ is a real number, then by definition $\alpha$ is algebraic if and only if, for some $n$, the vector

$$(1, \alpha, \alpha^2, \ldots, \alpha^n) \tag{8}$$

has an integer relation. Integer relation algorithm can be employed to search for minimal polynomial in a straightforward way by simply feeding it the vector (8) as its input. Let $\tilde{\alpha}$ be an approximate value belonging to an unknown algebraic number $\alpha$, considering the vector $\mathbf{v} = (1, \tilde{\alpha}, \tilde{\alpha}^2, \ldots, \tilde{\alpha}^n)$, how to obtain the exact minimal polynomial from its approximate root? We have the same technique to answer the question from the following theorem.

THEOREM 3.7 *Let $\tilde{\alpha}$ be an approximate value belonging to an unknown algebraic number $\alpha$ of degree $n > 0$. If*

$$\varepsilon = |\alpha - \tilde{\alpha}| < \frac{1}{(n^2(n+1)^{n-1/2}N^{2n})}, \tag{9}$$

*where $N$ is the upper bound on the height of its minimal polynomial, then $G(\alpha) = 0$, and the primitive part of $G(x)$ is its minimal polynomial.*

*Proof* The key states of the proof of Theorem 3.7 is to obtain error controlling $\varepsilon$ relationship with the degree $n$ and height $N$. Let $\alpha \in \mathbb{R}$ where $|\alpha| \leq 1$. From Corollary 3.6, it is obvious that

$$G(\alpha) = 0,$$

if and only if

$$|G(\alpha)| < n^{-1} \cdot (n+1)^{-n+1/2} \cdot |G|_\infty^{-n} \cdot N^{1-n}. \tag{10}$$

Under the assumption of the theorem, we get the upper bound of degree $n$ and an approximate value $\tilde{\alpha}$ belonging to an unknown algebraic number $\alpha$.

For substituting the approximate value $\tilde{\alpha}$ in $G(x)$, denoted by $G(\tilde{\alpha})$, there are two cases:

Case 1: $G(\tilde{\alpha}) \neq 0$, $|G(\tilde{\alpha})| > 0$. Here, the $G(\tilde{\alpha})$ is the constant, $|G(\tilde{\alpha})|$ represents its absolute value. We have the inequality (6) of Lemma 3.5 holds, that is,

$$0 < n^{-1} \cdot (n+1)^{-n+1/2} \cdot |G|_{\infty}^{-n} \cdot N^{1-n} - n \cdot \varepsilon \cdot |G|_{\infty}. \tag{11}$$

Clearly, the inequality (11) satisfies from the condition (9). This proves the Case 1.

Case 2: $G(\tilde{\alpha}) = 0$. From Lemma 3.1, we have $|G(\alpha) - G(\tilde{\alpha})| < n \cdot \varepsilon \cdot |G|_{\infty}$, hence $|G(\alpha)| < n \cdot \varepsilon \cdot |G|_{\infty}$. In order to satisfy condition (10), we only need the following inequality holds,

$$n \cdot \varepsilon \cdot |G|_{\infty} < n^{-1} \cdot (n+1)^{-n+1/2} \cdot |G|_{\infty}^{-n} \cdot N^{1-n}. \tag{12}$$

From Theorem 2.7, and Algorithm 2.2 in Step 7, $|G|_{\infty}$ is not more than $N$. Hence, we replace $|G|_{\infty}$ by $N$. So, the correctness of the inequality (12) follows from (9). This proves Theorem 3.7. ■

*Remark 3.8* We consider that $\varepsilon = |\alpha - \tilde{\alpha}|$ is the general case. For $n = 1$, Theorem 3.7 proves that $|\alpha - \tilde{\alpha}| < 1/(\sqrt{2}N^2)$ needed to obtain the exact value in $\mathbb{Q}$, which is superior to $|\alpha - \tilde{\alpha}| < 1/(2N^2)$ by continued fraction in reference [35]. For $n = 2$, the result of Theorem 3.7 is consistent with Theorem 5 of our original works in [25].

It is easiest to appreciate the theorem by seeing how it justifies the following algorithm for obtaining minimal polynomials from its approximation:

ALGORITHM 3.1 Reconstructing minimal polynomial

***Input:*** *a floating number $\tilde{\alpha}$ to $\alpha$ satisfying (9), upper bounds n and N on the degree and height.*
***Output:*** *$g(x)$, the minimal polynomial of $\alpha$.*

*(1)* ***while*** $2 \leq i \leq n$ ***do***
*(2)* $\mathbf{x} := (1, \tilde{\alpha}, \ldots, \tilde{\alpha}^i)$
*(3)* *Call Algorithm 2.2 with $\gamma > 2/\sqrt{3}$ producing an integer relation $\mathbf{g_i} = (g_0, g_1, \ldots, g_i)$ for $\mathbf{x}$*
$g(x) :=$ *the primitive part of* $\sum_{j=0}^{i} g_j x^j$
*(4)* ***if*** *height*$(g(x)) > \gamma^{n-2}\sqrt{n+1}N$ ***then***
*(5)* $i := i + 1$
*(6)* ***else return*** $g(x)$
*(7)* ***end while***

*Remark 3.9* The termination of Algorithm 3.1 is from Theorem 2.7. According to Theorem 2.7, we can know that the norm of obtained relation for $\mathbf{x}$ is less than $\gamma^{n-2}\sqrt{n+1}N$. Otherwise, it continues with the next iteration of the loop until the condition is met.

THEOREM 3.10 *Algorithm 3.1 works correctly as specified and uses $\mathcal{O}(n^4 + n^3 \log N)$ arithmetic operations on floating-point numbers having $O(n(\log n + \log N))$ binary bit operations, where n and N are the degree and height of its minimal polynomial, respectively.*

*Proof* The proof of Algorithm 3.1 on the arithmetic operations refers to the Corollary 2 of [12]. Correctness follows from Theorem 3.7. From Equation (9), at most $\lfloor \lg(n^2(n+1)^{n-1/2}N^{2n}) \rceil$ correct decimal digits are needed to guarantee the output is correct. So, the cost of the algorithm is $\log \varepsilon \in O(n(\log n + \log N))$ binary bit operations obviously. ■

Table 1 gives a comparison of the digits and complexity of four different minimal polynomial finding algorithms in the worst case. Since the algorithm in [5] needs to recognize a primitive

Table 1. Comparison of different minimal polynomial finding algorithms.

| | Digits | Complexity |
|---|---|---|
| Just [13] | $\mathcal{O}(n^2 + n^2 \log N)$ | $\mathcal{O}(n^8 \log n + n^8 \log N)$ |
| KLL [17] | $\mathcal{O}(n^2 + n \log N)$ | $\mathcal{O}(n^5 + n^4 \log N)$ |
| CG [5] | $\mathcal{O}\left(\log\left(\sum_{k=1}^{n-1}\binom{n}{k}\max_{j=k+1,\ldots,n}\left(\binom{n-k}{j-k}(2^{2n}N)^{j-k}\right)\right)\right)$ | — |
| RMP | $\mathcal{O}(n(\log n + \log N))$ | $\mathcal{O}(n^4 + n^3 \log N)$ |

element, we do not compare the complexity with it. It seems that a lower complexity can be achieved by using some new type LLL algorithms, such as H-LLL [22] and $L^2$ [23], but when we apply these new algorithms to find the minimal polynomial, we have to choose $\varepsilon$ as in a similar formula with Equation (9). Thus, multiple precision arithmetic is inevitable.

### 3.2 *Some applications*

In this section, we discuss some applications to the practicalities. The method of obtaining exact minimal polynomial from an approximate root can be extended to the set of complex numbers and many applications in computer algebra and science.

This yields a simple factorization algorithm for multivariate polynomials with rational coefficients: we can reduce a multivariate polynomial to a bivariate polynomial using the Hilbert irreducibility theorem, the basic idea of which was described in [9], and then convert a bivariate polynomial to a univariate polynomial by substituting an algebraic number of high degree for one variate in [7] or a transcendental number in [30]. After this substitution, we can get an approximate root of the univariate polynomial and use our algorithm to find the irreducible polynomial satisfied by the approximate root, which must then be a factor of the given polynomial. It can find the bivariate polynomial's factors, from which the factors of the original multivariate polynomial can be recovered using Hensel lifting. This is repeated until all the factors are found.

The other application yields an efficient method of converting the rational approximation representation to the minimal polynomial representation of an algebraic number. For more details refer to [26].

We also discuss some applications to some transcendental numbers by using an improved parameterized integer relation construction method. The form of these transcendental numbers is $\sin^{-1}(\alpha)$, $\cos^{-1}(\alpha)$, $\log(\alpha)$, etc., where $\alpha$ is an algebraic number. Moreover, a large number of results were found by using integer relation detection algorithm in the course of research on multiple sums and quantum field theory in [10].

Suppose $\beta$ is the principle value of $\sin^{-1}(\alpha)$ for some unknown $\alpha$, which is, however, known to the algebraic of degree and height at most $n$ and $N$, respectively. We consider inferring the minimal polynomial of $\alpha$ from an approximation $\tilde{\beta}$ to $\beta$ in the deterministic polynomial time. We show that if $|\beta - \tilde{\beta}|$ is at most $\varepsilon = 1/(n^2(n+1)^{n-1/2}N^{2n})$, this can be done. The specific technique is similar to the method in [17].

Thus, in polynomial time, we can compute from $\tilde{\beta}$ an approximation $\tilde{\alpha}$ to an unknown algebraic number $\alpha$ such that $|\alpha - \tilde{\alpha}| \leq \varepsilon$, with $\varepsilon$ as above. Now Theorem 3.7 guarantees that we can find the minimal polynomial of $\alpha$ in polynomial time.

### 3.3 *Some small examples in detail*

The first example illuminates how to obtain an exact minimal polynomial by its approximate root. Example 3.12 uses a simple example to test our algorithm for factoring primitive polynomials with integral coefficients.

*Example 3.11* Let a known floating number $\tilde{\alpha}$ belonging to some algebraic number $\alpha$ of degree $n = 4$, where $\tilde{\alpha} = 3.14626436994198$, we also know an upper bound of height on its minimal polynomial $N = 10$. According to Theorem 3.7, we can get the error $\varepsilon = 1/(n^2(n+1)^{n-1/2}N^{2n}) = 1/(4^2 \cdot 5^{7/2} \cdot 10^8) \approx 2.2 \times 10^{-12}$. Calling Algorithm 3.1, if only the floating number $\tilde{\alpha}$, such that $|\alpha - \tilde{\alpha}| < \varepsilon$, then we can get its minimal polynomial $g(x) = x^4 - 10x^2 + 1$.

*Example 3.12* This example is an application in factoring primitive polynomials over integral coefficients. For convenience and space-saving purposes, we choose a very simple and primitive polynomial as follows:

$$p = x^9 - 3x^8 + x^7 + 2x^5 - 9x^4 + 7x^3 + 10x^2 - 7x + 1.$$

We see the upper bound of coefficients on polynomial $p$ is 10, which has relation with an upper bound of coefficients of the factors on the primitive polynomial $p$ by Landau-Mignotte bound [21,28]. Taking $N = 5$, $n = 2$ yields $\varepsilon = 1/(2^2 \cdot (2+1)^{2-1/2} \cdot 5^4) = 1/(7500 \cdot \sqrt{3}) \approx 8.0 \times 10^{-5}$. Then, we compute the approximate root on $x$ with *Maple*, and get via [fsolve($p = 0, x$)]: $S = [2.618033989, 1.250523220, -0.9223475138, 0.3819660113, 0.2192284350]$.

According to Theorem 3.7, let $\tilde{\alpha} = 2.618033989$ be an approximate value belonging to some quadratic algebraic number $\alpha$. Calling Algorithm 3.1 yields as follows:

$$p_1 = x^2 - 3x + 1.$$

And then, we use the polynomial division to get

$$p_2 = x^7 + 2x^3 - 3x^2 - 4x + 1.$$

Based on the Eisenstein's Criterion [19], the $p_2$ is irreducible in $\mathbb{Z}[x]$. So, the $p_1$ and $p_2$ are the factors of primitive polynomial $p$.

## 4. Experimental results

Our algorithms have been implemented as a software package RMP in *Maple*. The following examples run in the same platform of *Maple* 13 under Windows and AMD Athlon(tm) 2.70 GHz, 2.00 GB of main memory. Figure 1 proposes the *Digits* of approximate values to compare our method with Just [13], KLL [17], and CG [5].

In Figure 1, we present many examples to compare our new method against three different algorithms. For each example, we construct the irreducible polynomial with random integral coefficients in the range $-100 \leq coeffs \leq 100$. Here, all the results are obtained under the condition that $\gamma = 2/\sqrt{3} + 10^{-15}$ with the parameterized integer relation construction algorithm.

From Figure 1, we have the observations as follows:

- Just generally works for algebraic number with degrees not higher than 10 within reasonable digits.
- The *Digits* of RMP is far less than the LLL-lattice basis reduction technique, such as Just and KLL. The *Digits* of RMP is slightly more than that of CG. However, their method must check whether there is a primitive element. Moreover, in the further work, we would like to consider improving the error controlling.

In addition, we also construct the irreducible polynomial with random degree in the range $2 \leq degree \leq 30$. For each example, they have the similar results such as Figure 1.

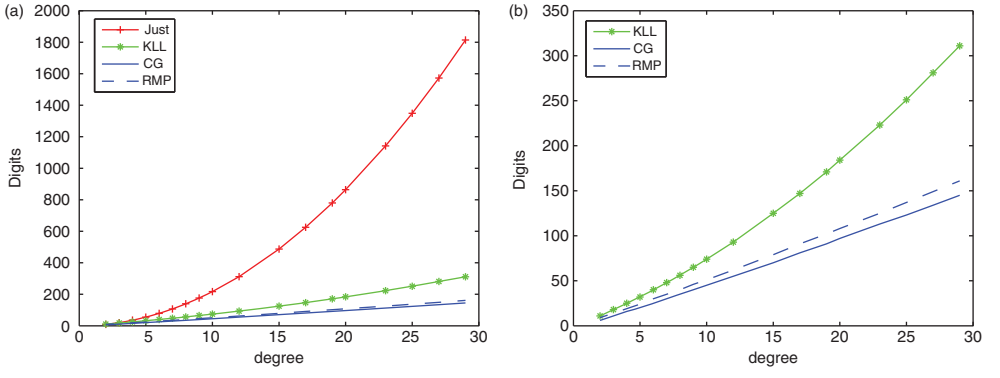


Figure 1. Digits of different minimal polynomial finding algorithms. (a) Digits of approximation. (b) Same as Figure 1(a), with a smaller Digits scaling.



## 5. Conclusions

In this article, we apply the floating-point parameterized integer relation constructed method and provide a sufficient condition on the precision of the approximation to its exact minimal polynomial. This previous algorithm relies on the LLL lattice reduction algorithm or Number Theory techniques. Compared with the subsistent methods, our new approach is more efficient and practical. Using our algorithm, we have succeeded in factoring polynomial with rational coefficients and providing an efficient method of converting the rational approximation representation to the minimal polynomial representation.

In the future, we would like to consider the further research of floating-point integer relation algorithm and exact multivariate polynomial factorization with rational coefficients by homotopy continued method. Furthermore, our basic idea can be generalized easily to complex algebraic numbers.

## Acknowledgements

This work was partially supported by the National Basic Research Program of China (2011CB302402), the National Natural Science Foundation of China (91118001, 11171053), and the West Light Foundation of the Chinese Academy of Sciences.

The authors are grateful to the anonymous referees for their helpful comments and suggestions.

## Note

1. $\varepsilon$ is defined by the same way for the rest of this article.

## References

[1] D. Bailey, J. Borwein, V. Kapoor, and E. Weisstein, *Ten problems in experimental mathematics*, Am. Math. Month. 113 (2006), pp. 481–509.
[2] J. Borwein and R. Corless, *Emerging tools for experimental mathematics*, Am. Math. Month. 106 (1999), pp. 889–909.
[3] P. Borwein, K.G. Hare, and A. Meichsener, *Reverse Symbolic Computations, The Identity Function*, Proceedings from the Maple Summer Workshop, Maple Software, Waterloo, 2002.
[4] J.M. Borwein and P. Lisonek, *Applications of integer relation algorithms*, Disc. Math. 217 (2000), pp. 65–82.
[5] G. Chèze and A. Galligo, *Four lectures on polynomial absolute factorization*, in *Solving Polynomial Equations: Foundations, Algorithms, and Applications*, A. Dickenstein and I. Emiris, eds., Algorithms and Computation in Mathematics Vol. 14, Springer-Verlag, 2005, pp. 339–392.

[6] G. Chèze and A. Galligo, *From an approximate to an exact absolute polynomial factorization*, J. Symbol. Comput. 41 (2006), pp. 682–696.
[7] J.W. Chen, Y. Feng, X.L. Qin, and J.Z. Zhang, *Exact Polynomial Factorization by Approximate High Degree Algebraic Numbers*, Proceedings of 2009 International Workshop on Symbolic–Numeric Computation SNC'09, ACM Press, New York, 2009, pp. 21–28.
[8] R.M. Corless, M.W. Giesbrecht, M. van Hoeij, I.S. Kotsireas, and S.M. Watt, *Towards Factoring Bivariate Approximate Polynomials*, Proceedings of 2001 International Symposium on Symbolic and Algebraic Computation ISSAC'01, ACM Press, New York, 2001, pp. 85–92.
[9] R.M. Corless, A. Galligo, I.S. Kotsireas, and S.M. Watt, *A Geometric-Numeric Algorithm for Absolute Factorization of Multivariate Polynomials*, Proceedings of 2002 International Symposium on Symbolic and Algebraic Computation ISSAC'02, ACM Press, New York, 2002, pp. 37–45.
[10] H.B. David, *Integer relation detection*, Comput. Sci. Eng. 2 (2000), pp. 24–28.
[11] C. Fieker and C. Friedrichs, *On reconstruction of algebraic numbers*, in *Algorithmic Number Theory (Leiden, 2000)*, LNCS Vol. 1838, Springer-Verlag, Berlin, Heidelberg, 2000, pp. 285–296.
[12] H.R.P. Ferguson, D.H. Bailey, and S. Arno, *Analysis of PSLQ, an integer relation finding algorithm*, Math. Comput. 68 (1999), pp. 351–369.
[13] B. Just, *Integer relations among algebraic numbers*, Math. Found. Comput. Sci. (1989), pp. 314–320.
[14] E. Kaltofen, *Challenges of symbolic computation: My favorite open problems*, J. Symbol. Comput. 29 (2000), pp. 891–919.
[15] E. Kaltofen, B. Li, Z.F. Yang, and L.H. Zhi, *Exact Certification of Global Optimality of Approximate Factorizations via Rationalizing Sums-of-Squares with Floating Point Scalars*, Proceedings of 2008 International Symposium on Symbolic and Algebraic Computation ISSAC'08, ACM Press, New York, pp. 155–163.
[16] R. Kannan, A.K. Lenstra, and L. Lovász, *Polynomial Factorization and Nonrandomness of Bits of Algebraic and Some Transcendental Numbers*, Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing STOC'84, ACM Press, New York, 1984, pp. 191–200.
[17] R. Kannan, A.K. Lenstra, and L. Lovász, *Polynomial factorization and nonrandomness of bits of algebraic and some transcendental numbers*, Math. Comput. 50 (1988), pp. 235–250.
[18] D.E. Knuth, *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*, 3rd ed., China Machine Press, Beijing, 2008.
[19] S. Lang, *Algebra*, 3rd ed., Springer-Verlag, New York, 2002.
[20] B. Li, J. Nie, and L.H. Zhi, *Approximate GCDs of polynomials and sparse SOS relaxations*, Theor. Comput. Sci. 409 (2008), pp. 200–210.
[21] M. Mignotte and D. Stefanescu, *Polynomials: An Algorithmic Approach*, Springer-Verlag, Singapore, 1999.
[22] I. Morel, D. Stehlé, and G. Villard, *H-LLL: Using Householder Inside LLL*, Proceedings of 2009 International Symposium on Symbolic and Computation ISSAC'09, ACM Press, New York, 2009, pp. 271–278.
[23] P.Q. Nguyen and D. Stehlé, *Floating-point LLL Revisited*, Proceedings of 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, EUROCRYPT 2005, 2005, pp. 215–233.
[24] G. Pablo and H. Erwin, *Error estimates for the approximation of a class of optimal control systems governed by linear PDEs*, Numer. Funct. Anal. Optim. 30 (2009), pp. 523–547.
[25] X.L. Qin, Y. Feng, J.W. Chen, and J.Z. Zhang, *Finding Exact Minimal Polynomial by Approximations*, Proceedings of 2009 International Workshop on Symbolic–Numeric Computation SNC'09, ACM Press, New York, 2009, pp. 125–131.
[26] X.L. Qin, Y. Feng, J.W. Chen, and J. Li, *Exact representation of real algebraic number by approximations and its applications*, J. Sichuan Univ. (Engineering Science Edition) (in Chinese) 42 (2010), pp. 126–131.
[27] S.M. Rump, *Computer-assisted proofs and self-validating methods*, in *Handbook of Accuracy and Reliability in Scientific Computation*, B. Einarsson, ed., Society for Industrial and Applied Mathematics, Philadelphia, 2005, pp. 195–240.
[28] A. Schinzel, *Polynomials with Special Regard to Reducibility*, Cambridge University Press, Cambridge, 2000.
[29] L.N. Trefethen, *Predictions for Scientific Computing Fifty Years from Now*, Technical Report, University of Manchester, 1998. Available at http://eprints.maths.ox.ac.uk/1304/.
[30] M.-P. Van Der Hulst and A.K. Lenstra, *Factorization of Polynomials by Transcendental Evaluation*, Proceedings of European Conference on Computer Algebra EUROCAL'85, LNCS Vol. 204, Springer-Verlag, Berlin, Heidelberg, 1985, pp. 138–145.
[31] W.T. Wu, *Hybrid computation*, in: *100 Interdisciplinary Science Puzzles of the 21st Century* (in Chinese), J.X. Li, ed., Science Press, Beijing, 2005, pp. 656–657.
[32] W.T. Wu and X.S. Gao, *Mathematics mechanization and applications after thirty years*, Front. Comput. Sci. China 1 (2007), pp. 1–8.
[33] A.R. Yaakub and D.J. Evans, *A fourth order Runge–Kutta RK(4,4) method with error control*, Int. J. Comput. Math. 71 (1999), pp. 383–411.
[34] L. Yang, J.Z. Zhang, and X.H. Hou, *Nonlinear Algebraic Equation System and Automated Theorem Proving* (in Chinese), Shanghai Scientific and Technological Education Publishing House, Shanghai, 1996.
[35] J.Z. Zhang and Y. Feng, *Obtaining exact value by approximate computations*, Sci. China Math. 50 (2007), pp. 1361–1368.
[36] S.P. Zhou and M.G. Cui, *Approximate solution for a variable-coefficient semilinear heat equation with nonlocal boundary conditions*, Int. J. Comput. Math. 86 (2009), pp. 2248–2258.