MATHEMATICS OF COMPUTATION
Volume 88, Number 317, May 2019, Pages 1479–1501
https://doi.org/10.1090/mcom/3356
Article electronically published on June 20, 2018

THE PSLQ ALGORITHM FOR EMPIRICAL DATA

YONG FENG, JINGWEI CHEN, AND WENYUAN WU

Abstract. The celebrated integer relation finding algorithm PSLQ has been successfully used in many applications. PSLQ was only analyzed theoretically for exact input data, however, when the input data are irrational numbers, they must be approximate ones due to the finite precision of the computer. When the algorithm takes empirical data (inexact data with error bounded) instead of exact real numbers as its input, how do we theoretically ensure the output of the algorithm to be an exact integer relation?

In this paper, we investigate the PSLQ algorithm for empirical data as its input. Firstly, we give a termination condition for this case. Secondly, we analyze a perturbation on the hyperplane matrix constructed from the input data and hence disclose a relationship between the accuracy of the input data and the output quality (an upper bound on the absolute value of the inner product of the exact data and the computed integer relation), which naturally leads to an error control strategy for PSLQ. Further, we analyze the complexity bound of the PSLQ algorithm for empirical data. Examples on transcendental numbers and algebraic numbers show the meaningfulness of our error control strategy.

1. Introduction

A vector $\boldsymbol{m} \in \mathbb{Z}^n \setminus \{\boldsymbol{0}\}$ is called an *integer relation* for $\boldsymbol{\alpha} \in \mathbb{R}^n$ if $\langle \boldsymbol{\alpha}, \boldsymbol{m} \rangle = 0$. The problem of finding integer relations for rational or real numbers can be dated back to the time of Euclid. It is closely related to the problem of finding a small vector in a Euclidean lattice. In fact, the celebrated Lenstra–Lenstra–Lovász (LLL) lattice basis reduction algorithm can be used to find an integer relation. This was already pointed out in [20, page 525]. The HJLS algorithm [16] is the first proved polynomial time algorithm for integer relation finding. The PSLQ algorithm [13, 14] is one of the most frequently used algorithms to find integer relations. Both HJLS and PSLQ can be viewed as algorithms to compute the intersection between a lattice and a vector space; see [11]. For detailed historical notes, we refer to [14, 16]. Nowadays, integer relation finding has been successfully used in different areas, such as experimental math [8, 22] and physics [7]. For more applications, we refer the reader to [10] and the references therein.

Received by the editor July 17, 2017, and, in revised form, November 8, 2017, and January 9, 2018.

2010 *Mathematics Subject Classification.* Primary 11A05, 11Y16; Secondary 68-04.

Key words and phrases. Integer relation, PSLQ, empirical data.

The first author was supported by NNSF (China) Grant 11671377 and 61572024.

The second author was supported by NNSF (China) Grant 11501540, CAS "Light of West China" Program and Youth Innovation Promotion Association of CAS.

The second author is the corresponding author.

The third author was supported by NNSF (China) Grant 11471307 and 11771421, Chongqing Research Program (cstc2015jcyjys40001, KJ1705121), and CAS Research Program of Frontier Sciences (QYZDB-SSW-SYS026).

However, there always exist some data that can only be obtained with limited accuracy. Indeed, all the input data in applications above are of limited accuracy, and hence not exact values. Consequently, it is of great importance to study how to obtain exact integer relations for $\boldsymbol{\alpha}$ from an approximation of $\bar{\boldsymbol{\alpha}}$ by PSLQ.

To the best of our knowledge, there exists only an experimental result on this topic, due to Bailey. Bailey in [5] suggested that if one wishes to recover an integer relation with coefficients bounded by G for an n-dimensional vector $\boldsymbol{\alpha}$, then the input vector $\boldsymbol{\alpha}$ must be specified to at least $n \log_{10} G$ decimal digits, and one must employ floating-point arithmetic with at least $n \log_{10} G$ accurate digits. Using this experimental result, a lot of nontrivial integer relations have been discovered by several implementations of PSLQ, such as MPFUN90 [4], ARPREC [2], etc., all of which employ high precision floating-point arithmetic. Recently, a PSLQ implementation in a new arbitrary precision package MPFUN2015 [3] has been used to discover large Poisson polynomials [8], including the largest successful integer relation computations performed to date (using $64,000$ decimal digits), based on the precision estimation suggested by Bailey. Bailey's precision estimation works well in practice, however it lacks theoretical support. In this paper, we attempt to provide a theory for the error control of PSLQ.

Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n$ be the *intrinsic data* (exact data that may not be known) with an integer relation within a 2-norm bound M, and let $\bar{\boldsymbol{\alpha}}$ be the *empirical data* with $\|\boldsymbol{\alpha} - \bar{\boldsymbol{\alpha}}\|_2 < \varepsilon_1$. Generally, $\bar{\boldsymbol{\alpha}}$ may not have an integer relation within the bound M. Therefore, the PSLQ algorithm may not terminate when we compute an integer relation from $\bar{\boldsymbol{\alpha}}$ because the element $h_{n,n-1}$ of the hyperplane matrix (see (2.1) and Algorithm 4) may never be transformed to zero.

So, firstly, we propose a new termination condition for the PSLQ algorithm. Secondly, even if PSLQ returns $\boldsymbol{m}$ from $\bar{\boldsymbol{\alpha}}$, we need to determine whether $\langle \boldsymbol{\alpha}, \boldsymbol{m} \rangle = 0$, without knowing the intrinsic data $\boldsymbol{\alpha}$. To do this requires a gap bound δ for $|\langle \boldsymbol{\alpha}, \boldsymbol{m} \rangle|$. A so-called *gap bound* for $|\langle \boldsymbol{\alpha}, \boldsymbol{m} \rangle|$ is that there exist a given $\delta > 0$ such that $|\langle \boldsymbol{\alpha}, \boldsymbol{m} \rangle| > \delta$ whenever $|\langle \boldsymbol{\alpha}, \boldsymbol{m} \rangle| \neq 0$. If there exists no further information about $\boldsymbol{\alpha}$, then there does not exist a gap bound in general. However, a gap bound can be given when the α_i's are algebraic numbers [18, 19]. Once we have a gap bound δ and $|\langle \boldsymbol{\alpha}, \boldsymbol{m} \rangle| < \delta$, it guarantees $\langle \boldsymbol{\alpha}, \boldsymbol{m} \rangle = 0$, even without knowing $\boldsymbol{\alpha}$. In this paper, we will not discuss the gap bound, but focus on how to estimate $|\langle \boldsymbol{\alpha}, \boldsymbol{m} \rangle|$ via establishing a relation between $|\langle \boldsymbol{\alpha}, \boldsymbol{m} \rangle|$ and $|\langle \bar{\boldsymbol{\alpha}}, \boldsymbol{m} \rangle|$. Thirdly, we analyze the computation complexity of the PSLQ algorithm for empirical data. Finally, we also give some illustrative examples that show how helpful the error control strategies are for applications of PSLQ.

2. Preliminaries

For completeness, we recall the PSLQ algorithm in this section. As indicated in [14], PSLQ works for both the real case and the complex case. For the complex case, it may find a Gaussian integer relation for a given $\boldsymbol{\alpha} \in \mathbb{C}^n$. For simplicity, we only consider the real case here. Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n$ with $\alpha_i \neq 0$ for $i = 1, \ldots, n$. Given $\boldsymbol{\alpha}$ as above, define the *hyperplane matrix* $\boldsymbol{H}_\alpha = (h_{i,j})$ with

$$
h_{i,j} = \begin{cases} 0, & \text{if } 1 \le i < j \le n-1, \\ s_{i+1}/s_i, & \text{if } 1 \le i = j \le n-1, \\ -\alpha_i \alpha_j / (s_j s_{j+1}), & \text{if } 1 \le j < i \le n, \end{cases} \tag{2.1}
$$

where $s_j^2 = \sum_{k=j}^{n} \alpha_k^2 > 0$ for $j = 1, 2, \ldots, n$.

Further, we can assume that $\|\boldsymbol{\alpha}\| = 1$ ($\|\cdot\|$ is the Euclidean norm), since the hyperplane matrix $\boldsymbol{H}_\alpha$ is scale-invariant with respect to $\boldsymbol{\alpha}$, i.e., $\boldsymbol{H}_\alpha = \boldsymbol{H}_{c\cdot\alpha}$ for $c \in \mathbb{R} \setminus \{0\}$.

Algebraically, PSLQ produces a series of unimodular matrices in $\mathrm{GL}_n(\mathbb{Z})$ multiplying $\boldsymbol{H}_\alpha$ from the left and a series of orthogonal matrices multiplying $\boldsymbol{H}_\alpha$ from the right. These matrices are produced by the following subroutines (Algorithms 1, 2, and 3).

Algorithm 1 (`SizeReduction`)

Require: A lower trapezoidal $n \times (n-1)$ matrix $\boldsymbol{H} = (h_{i,j})$ with $h_{i,j} = 0$ if $j > i$ and $h_{j,j} \neq 0$.

Ensure: A unimodular matrix $\boldsymbol{D}$ such that $\boldsymbol{H} := \boldsymbol{D} \cdot \boldsymbol{H} = (h_{i,j})$ satisfying $|h_{i,j}| \leq |h_{j,j}|/2$ for $1 \leq j < i \leq n$.

$\boldsymbol{D} := \boldsymbol{I}_n$.
for i from 2 to n **do**
  for j from $i-1$ to 1 by stepsize -1 **do**
    $q := \lfloor h_{i,j}/h_{j,j} + 0.5 \rfloor$.
    for k from 1 to n **do**
      $d_{i,k} := d_{i,k} - q d_{j,k}$.
    end for
  end for
end for

We call the process in Algorithm 1 *size reduction*. In the PSLQ paper [14], size reduction is called Hermite reduction. To avoid confusion with the Hermite Normal Form for integral matrices or the Hermite reduction in the integration of algebraic functions [17] (also for creative telescoping) and to be consistent with the similar process used in lattice basis reduction algorithms, we replace "Hermite reduction" by "size reduction".

Algorithm 2 (`BergmanSwap`)

Require: A lower trapezoidal $n \times (n-1)$ matrix $\boldsymbol{H} = (h_{i,j})$ with $h_{i,j} = 0$ if $j > i$ and $h_{j,j} \neq 0$, and a parameter $\gamma > 2/\sqrt{3}$.

Ensure: A unimodular matrix $\boldsymbol{D}$ resulting from the exchange of two rows of the identity matrix and the exchange position r.

$\boldsymbol{D} := \boldsymbol{I}_n$.
Choose r such that $\gamma^r |h_{r,r}| = \max_{j \in \{1,\ldots,n-1\}} \left\{\gamma^j \cdot |h_{j,j}|\right\}$, and then swap the rth row and the $(r+1)$th row of $\boldsymbol{D}$.

After a Bergman swap, $\boldsymbol{H} := \boldsymbol{D}\boldsymbol{H}$ is usually not lower trapezoidal. We may multiply the updated $\boldsymbol{H}$ by an orthogonal matrix $\boldsymbol{Q}$ from the right such that $\boldsymbol{H}\boldsymbol{Q}$ is again a lower trapezoidal matrix. This procedure is called `Corner`, which is equivalent to performing the LQ-decomposition of $\boldsymbol{H}$ (QR-decomposition of $\boldsymbol{H}^T$). Suppose after a Bergman swap, the rth and $(r+1)$th rows of $\boldsymbol{H}$ are swapped. Let

$$\eta = h_{r,r}, \quad \beta = h_{r+1,r}, \quad \lambda = h_{r+1,r+1}, \quad \delta = \sqrt{\beta^2 + \lambda^2}. \tag{2.2}$$

Then we can give the following explicit formula for `Corner` instead of computing the full LQ-decomposition.

Algorithm 3 (`Corner`)

Require: An $n \times (n-1)$ matrix $\boldsymbol{H}$ that is obtained by a Bergman swap with the rth and $(r+1)$th rows swapped, where $r < n-1$.

Ensure: An orthogonal matrix $\boldsymbol{Q}$ such that $\boldsymbol{HQ}$ is the L-factor of the LQ-decomposition of $\boldsymbol{H}$.

1: Return $\boldsymbol{Q} = (q_{i,j}) \in \mathbb{R}^{(n-1)\times(n-1)}$ with

$$q_{i,j} = \begin{cases} \beta/\delta & \text{if } i = r, j = r, \\ -\lambda/\delta & \text{if } i = r, j = r+1, \\ \lambda/\delta & \text{if } i = r+1, j = r, \\ \beta/\delta & \text{if } i = r+1, j = r+1, \\ 1 & i = j \neq r \text{ or } i = j \neq r+1, \\ 0 & \text{otherwise.} \end{cases}$$

Now, we are ready to give the following description of the PSLQ algorithm. Note that we suppose that $\boldsymbol{\alpha} \in \mathbb{R}^n$ has integer relations. In fact, this hypothesis is reasonable, because Babai, Just, and Meyer auf der Heide [1] showed that under the exact real arithmetic computation model, it is not possible to decide whether there exists an integer relation for given input $\boldsymbol{\alpha} \in \mathbb{R}^n$. In addition, we omit an early termination condition that checks whether there exists a column of $\boldsymbol{B}$ that is an integer relation, because it does not impact the analysis for the worst case.

Algorithm 4 (`PSLQ`)

Require: An n-dimensional vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ with $\|\alpha\| = 1$ (suppose that $\boldsymbol{\alpha}$ has integer relations) and $\gamma > 2/\sqrt{3}$.

Ensure: An integer relation $\boldsymbol{m}$ for $\boldsymbol{\alpha}$.

1: Construct $\boldsymbol{H}_\alpha$ as in formula (2.1). Set $\boldsymbol{H} := \boldsymbol{H}_\alpha$. Set the $n \times n$ matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ to the identity matrix $\boldsymbol{I}_n$. Let $\boldsymbol{D} :=$ `SizeReduce`$(\boldsymbol{H})$. Update $\boldsymbol{\alpha} := \boldsymbol{\alpha D}^{-1}$, $\boldsymbol{H} := \boldsymbol{DH}$, $\boldsymbol{A} := \boldsymbol{DA}$, and $\boldsymbol{B} := \boldsymbol{BD}^{\boldsymbol{-1}}$.
2: **while** $h_{n,n-1} \neq 0$ **do**
3: Let $(\boldsymbol{D}, r) :=$ `BergmanSwap`$(\boldsymbol{H}, \gamma)$, where $\boldsymbol{D}$ is the transform matrix and r is the exchange position. Update $\boldsymbol{\alpha} := \boldsymbol{\alpha D}^{-1}$, $\boldsymbol{H} := \boldsymbol{DH}$, $\boldsymbol{A} := \boldsymbol{DA}$, and $\boldsymbol{B} := \boldsymbol{BD}^{\boldsymbol{-1}}$.
4: **if** $r < n-1$ **then**
5: Let $\boldsymbol{Q} =$ `Corner`$(\boldsymbol{H})$ and update $\boldsymbol{H} := \boldsymbol{HQ}$.
6: **end if**
7: Let $\boldsymbol{D} :=$ `SizeReduce`$(\boldsymbol{H})$. Update $\boldsymbol{\alpha} := \boldsymbol{\alpha D}^{-1}$, $\boldsymbol{H} := \boldsymbol{DH}$, $\boldsymbol{A} := \boldsymbol{DA}$, and $\boldsymbol{B} := \boldsymbol{BD}^{\boldsymbol{-1}}$.
8: **end while**
9: Return the $(n-1)$th column of $\boldsymbol{B}$.

Remark 2.1. At the beginning, the hyperplane matrix $\boldsymbol{H}_\alpha$ has all diagonal elements nonzero. During the algorithm, all diagonal elements of $\boldsymbol{H}$ are always nonzero until the termination of `PSLQ`.

For the convenience of description, the procedure from step 3 to step 7 in Algorithm 4 is called an iteration of `PSLQ` as in [14, Section 3].

Theorem 2.2 ([14, Theorem 2]). *Assume that $\boldsymbol{\alpha} \in \mathbb{R}^n$ has integer relations. Let λ_α be the least 2-norm of relations for $\boldsymbol{\alpha}$. Then PSLQ will find an integer relation for $\boldsymbol{\alpha}$ in no more than*

$$\binom{n}{2} \frac{\log\left(\gamma^{n-1}\lambda_\alpha\right)}{\log \tau}$$

iterations, where $\tau = 1/\sqrt{1/\rho^2 + 1/\gamma^2}$ with $\gamma > 2/\sqrt{3}$ and $\rho = 2$.

3. The $\texttt{PSLQ}_\varepsilon$ algorithm

The termination of PSLQ requires one to check whether $h_{n,n-1} = 0$. When the input data $\boldsymbol{\alpha}$ with integer relations are exact, it will hold that $h_{n,n-1} = 0$ after finitely many iterations of PSLQ. And hence the output is an integer relation for $\boldsymbol{\alpha}$. However, when the input data $\bar{\boldsymbol{\alpha}}$ is an approximation of $\boldsymbol{\alpha}$, there may not exist any integer relation for $\bar{\boldsymbol{\alpha}}$. So $h_{n,n-1}$ is usually not equal to zero. This leads to the nontermination of PSLQ. Therefore, we first need to explore the termination condition of PSLQ for empirical data.

3.1. **An invariant relation of PSLQ.** Indeed, the quantity $h_{n,n-1}$ plays a very important role in PSLQ, not only for exact data, but also for empirical data. The following theorem gives a relationship between the $(n-1)$th column of $\boldsymbol{B}$ $(= \boldsymbol{A}^{-1})$ in PSLQ and $h_{n,n-1}$, which will be shown to be crucial for the study of termination of PSLQ with empirical data.

Denote by $\boldsymbol{H}(k)$ the end result of $\boldsymbol{H}$ after exactly k iterations of PSLQ.

Theorem 3.1. *Assume that $\boldsymbol{H}(k) = \boldsymbol{A}\boldsymbol{H}_\alpha\boldsymbol{Q}$, where $\boldsymbol{H}(k) = (h_{i,j}(k))$ is a lower trapezoidal matrix. Set $(z_1(k), \ldots, z_{n-1}(k), z_n(k)) = (\alpha_1, \ldots, \alpha_{n-1}, \alpha_n)\boldsymbol{A}^{-1}$. Then, it holds that*

$$|z_{n-1}(k)| \leq \sqrt{\alpha_{n-1}^2 + \alpha_n^2}\,|h_{n,n-1}(k)|.$$

Proof. From

$$(z_1(k), \ldots, z_{n-1}(k), z_n(k))\boldsymbol{H}(k) = \boldsymbol{\alpha}\boldsymbol{A}^{-1}\boldsymbol{A}\boldsymbol{H}_\alpha\boldsymbol{Q} = \boldsymbol{\alpha}\boldsymbol{H}_\alpha\boldsymbol{Q} = \boldsymbol{0}$$

it follows that

$$z_{n-1}(k)h_{n-1,n-1}(k) + z_n(k)h_{n,n-1}(k) = 0.$$

From [14, Lemma 5], it holds that $h_{n-1,n-1}(k) \neq 0$ before the termination of Algorithm 4. Then, it is obtained that

$$z_{n-1}(k) = -\frac{z_n(k)}{h_{n-1,n-1}(k)} h_{n,n-1}(k). \tag{3.1}$$

We claim that $|\frac{z_n(k)}{h_{n-1,n-1}(k)}|$ does not increase as k increases. In Algorithm 4, this quantity can be possibly changed only in SizeReduce, BergmanSwap, and Corner, so we consider them next, respectively. When the size reduction (step 7) is performed on row $i \leq n-1$ of $\boldsymbol{H}$, z_n and $h_{n-1,n-1}$ are unchanged, so $|\frac{z_n}{h_{n-1,n-1}}|$ is unchanged. When $i = n$, the size reduction matrix is as follows:

$$\boldsymbol{D} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ k_1 & k_2 & k_3 & \cdots & k_{n-1} & 1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{I}_{n-1} & 0 \\ \boldsymbol{K} & 1 \end{pmatrix},$$

where $\boldsymbol{K} = (k_1, \ldots, k_{n-1})$ is an integer vector and $\boldsymbol{I}_{n-1}$ is the $(n-1) \times (n-1)$ identify matrix. Its inverse is

$$D^{-1} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ -k_1 & -k_2 & -k_3 & \cdots & -k_{n-1} & 1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{I}_{n-1} & 0 \\ -\boldsymbol{K} & 1 \end{pmatrix}.$$

It is easy to see that the nth column of $\boldsymbol{A}^{-1}\boldsymbol{D}^{-1}$ is the same as that of $\boldsymbol{A}^{-1}$. Therefore, z_n is unchanged. On the other hand, $h_{n-1,n-1}$ is also unchanged after size reduction. Hence, $\frac{z_n}{h_{n-1.n-1}}$ is unchanged. In step 3 of Algorithm 4, the Bergman swap is performed between the rth and $(r+1)$th rows. When $r < n-2$, it is obvious that z_n and $h_{n-1,n-1}$ are unchanged. When $r = n-2$, the columns $n-2$ and $n-1$ of $\boldsymbol{A}^{-1}$ are swapped. So the nth column of $\boldsymbol{A}^{-1}$ is unchanged and z_n is also unchanged, that is $z_n(k+1) = z_n(k)$, while $h_{n-1,n-1}$ is changed as follows. Before step 3, let $\eta = h_{n-2,n-2}(k)$, $\beta = h_{n-1,n-2}(k)$, $\lambda = h_{n-1,n-1}(k)$, and $\delta = \sqrt{\beta^2 + \lambda^2}$; then we have

$$\begin{pmatrix} \eta & 0 \\ \beta & \lambda \end{pmatrix} \xrightarrow{\text{step 3}} \begin{pmatrix} \beta & \lambda \\ \eta & 0 \end{pmatrix} \xrightarrow{\text{step 5}} \begin{pmatrix} \delta & 0 \\ \frac{\eta\beta}{\delta} & -\frac{\eta\lambda}{\delta} \end{pmatrix}.$$

Therefore, after step 5, the new $h_{n-1,n-1}(k+1) = -\frac{\eta\lambda}{\delta}$. Since the swap occurs at rows $n-2$ and $n-1$, it holds that $|\eta| > \gamma|\lambda|$. Note that $|\beta| < \frac{|\eta|}{\rho}$ yields

$$\left|\frac{-\eta}{\delta}\right| = \frac{1}{\sqrt{\frac{\beta^2}{\eta^2} + \frac{\lambda^2}{\eta^2}}} > \frac{1}{\sqrt{\frac{1}{\rho^2} + \frac{1}{\gamma^2}}} = \tau,$$

where $\rho = 2$. So, it follows that

$$|h_{n-1,n-1}(k+1)| = |-\frac{\eta\lambda}{\delta}| > \tau|\lambda|.$$

Hence, it holds that

$$\left|\frac{z_n(k+1)}{h_{n-1,n-1}(k+1)}\right| < \frac{z_n(k)}{\lambda}\frac{1}{\tau} = \frac{1}{\tau}\left|\frac{z_n(k)}{h_{n-1,n-1}(k)}\right|.$$

Since $\frac{1}{\tau} < 1$, it implies that $\left|\frac{z_n}{h_{n-1,n-1}}\right|$ decreases. When $r = n-1$, rows $n-1$ and n of $\boldsymbol{H}$ are swapped, and so are columns $n-1$ and n of $\boldsymbol{A}^{-1}$. Hence $h_{n-1,n-1}$ and $h_{n,n-1}$ are swapped, and z_{n-1} and z_n are exchanged. Therefore, $h_{n-1,n-1}(k+1) = h_{n,n-1}(k)$ and $z_n(k+1) = z_{n-1}(k)$. From formula (3.1), it follows that

$$z_n(k+1) = z_{n-1}(k) = -\frac{h_{n,n-1}(k)}{h_{n-1,n-1}(k)}z_n(k) = -\frac{h_{n-1,n-1}(k+1)}{h_{n-1,n-1}(k)}z_n(k).$$

In this case, $\left|\frac{z_n}{h_{n-1,n-1}}\right|$ remains unchanged. Up to now, we have shown that $|\frac{z_n}{h_{n-1,n-1}}|$ either decreases or remains unchanged after the $(k+1)$th iteration of `PSLQ`. At the beginning of `PSLQ`, we have that $z_n(1) = \alpha_n$ and $h_{n-1,n-1}(k) = \frac{|\alpha_n|}{\sqrt{\alpha_{n-1}^2+\alpha_n^2}}$. Hence

$$\left|\frac{z_n(k)}{h_{n-1,n-1}(k)}\right| \leq \left|\frac{z_n(1)}{h_{n-1,n-1}(1)}\right| \leq \sqrt{\alpha_{n-1}^2 + \alpha_n^2},$$

which completes the proof. □

The property presented in Theorem 3.1 is an invariant of PSLQ in the sense that it always holds during the algorithm. Furthermore, Theorem 3.1 can be used to design an algorithm to find approximate integer relations in the following sense. Given $\boldsymbol{\alpha}$ which may not have an integer relation, if we take the $(n-1)$th column of $\boldsymbol{B}$ as an approximate integer relation for $\boldsymbol{\alpha}$ in Algorithm 4, Theorem 3.1 gives an error estimate, i.e., if PSLQ returns the $(n-1)$th column of $\boldsymbol{B}$, denoted by $\boldsymbol{m}$, when $|h_{n,n-1}| < \varepsilon_2$, then

$$|\langle \boldsymbol{\alpha}, \boldsymbol{m} \rangle| \leq \sqrt{\alpha_{n-1}^2 + \alpha_n^2}\, \varepsilon_2.$$

Now we improve the algorithm as follows.

Algorithm 5 ($\mathtt{PSLQ}_\varepsilon$)

Require: A lower trapezoidal matrix $\boldsymbol{H} \in \mathbb{R}^{n\times(n-1)}$ with all diagonal entries nonzero, $\varepsilon_2 > 0$ and $\gamma > 2/\sqrt{3}$.

Ensure: An n-dimensional integer vector $\boldsymbol{m}$.

1: Set the $n \times n$ matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ to the identity matrix $\boldsymbol{I}_n$. Let $\boldsymbol{D} := \mathtt{SizeReduce}(\boldsymbol{H})$. Update $\boldsymbol{H} := \boldsymbol{DH}$, $\boldsymbol{A} := \boldsymbol{DA}$, and $\boldsymbol{B} := \boldsymbol{BD}^{-\mathbf{1}}$.
2: **while** $|h_{n,n-1}| \geq \varepsilon_2$ **do**
3: Let $(\boldsymbol{D}, r) := \mathtt{BergmanSwap}(\boldsymbol{H}, \gamma)$, where $\boldsymbol{D}$ is the transform matrix and r is the exchange position. Update $\boldsymbol{H} := \boldsymbol{DH}$, $\boldsymbol{A} := \boldsymbol{DA}$, and $\boldsymbol{B} := \boldsymbol{BD}^{-\mathbf{1}}$.
4: **if** $r < n-1$ **then**
5: Let $\boldsymbol{Q} = \mathtt{Corner}(H)$ and update $\boldsymbol{H} := \boldsymbol{HQ}$.
6: **end if**
7: Let $\boldsymbol{D} := \mathtt{SizeReduce}(\boldsymbol{H})$. Update $\boldsymbol{H} := \boldsymbol{DH}$, $\boldsymbol{A} := \boldsymbol{DA}$, and $\boldsymbol{B} := \boldsymbol{BD}^{-\mathbf{1}}$.
8: **end while**
9: Return the $(n-1)$th column of $\boldsymbol{B}$.

Besides the termination condition being replaced by $|h_{n,n-1}| < \varepsilon_2$, the main difference of $\mathtt{PSLQ}_\varepsilon$ from PSLQ is that the input is changed as a more general lower trapezoidal matrix which may not satisfy the fine structure in (2.1). The remainder of this section will be devoted to analyze $\mathtt{PSLQ}_\varepsilon$.

3.2. **Termination and complexity.** We now show that $\mathtt{PSLQ}_\varepsilon$ terminates after a finitely many number of iterations stated in the following theorem.

Theorem 3.2. *Given $\boldsymbol{H} \in \mathbb{R}^{n\times(n-1)}$, $\mathtt{PSLQ}_\varepsilon$ terminates in no more than*

$$\frac{n(n+1)((n-1)\log\gamma + \log\frac{1}{\varepsilon_2})}{2\log\tau}$$

iterations, where $\tau = 1/\sqrt{1/\rho^2 + 1/\gamma^2}$.

Proof. Define the Π function after k iterations as follows:

$$\Pi(k) = \prod_{j=1}^{n-1} \max\left(|h_{i,i}(k)|, \frac{h_{\max}(k)}{\gamma^{n-1}}\right)^{n-j},$$

where $h_{\max}(k)$ is the maximum of $|h_{i,i}(k)|$ for $i = 1, 2, \ldots, n-1$. Then the proof is similar to the proof of [14, Theorem 2]. See Appendix B for the full proof. ☐

Note that if $\boldsymbol{H}$ is the hyperplane matrix for $\boldsymbol{\alpha} \in \mathbb{R}^n$ and $\boldsymbol{\alpha}$ has an integer relation, let M_α be the minimal 2-norm of integer relations for $\boldsymbol{\alpha}$. Then from [14, Theorem 1], it holds that $\frac{1}{h_{\max(k)}} \leq M_\alpha$. From inequality (B.2), it can be obtained that

$$k \leq \frac{n(n-1)((n-1)\log\gamma + \log\frac{1}{h_{\max(k)}})}{2\log\tau} \leq \frac{n(n-1)((n-1)\log\gamma + \log M_\alpha)}{2\log\tau},$$

which is the same as [14, Theorem 2].

3.3. **Perturbation analysis of $\texttt{PSLQ}_\varepsilon$.** Before we present the technical details, we recall some notation. For the intrinsic data $\boldsymbol{\alpha}$, we assume that we can only obtain the corresponding empirical data $\bar{\boldsymbol{\alpha}}$ with $\|\boldsymbol{\alpha} - \bar{\boldsymbol{\alpha}}\|_2 < \varepsilon_1$. For $\bar{\boldsymbol{\alpha}}$, we can construct its hyperplane matrix $\boldsymbol{H}_{\bar{\alpha}}$ as in (2.1). But we cannot use $\boldsymbol{H}_{\bar{\alpha}}$ as the input matrix for $\texttt{PSLQ}_\varepsilon$. Instead, we use $\overline{\boldsymbol{H}}_\alpha$ to represent a more general perturbation to $\boldsymbol{H}_\alpha$ including round-off errors in computing $\boldsymbol{H}_{\bar{\boldsymbol{\alpha}}}$, which only keeps the lower trapezoidal structure and satisfies

$$\|\overline{\boldsymbol{H}}_\alpha - \boldsymbol{H}_\alpha\|_F \leq \varepsilon_3, \tag{3.2}$$

where $\|\cdot\|_F$ is the matrix Frobenius norm. Suppose that one wants to find an integer relation for $\boldsymbol{\alpha} \in \mathbb{R}^n$ by using $\texttt{PSLQ}_\varepsilon$, where the input is $\overline{\boldsymbol{H}}_\alpha$, the termination condition is $|h_{n,n-1}| < \varepsilon_2$, and the output is $\boldsymbol{m}$. Next, we investigate the relations among ε_2, ε_3, and $|\langle \boldsymbol{m}, \boldsymbol{\alpha}\rangle|$.

Denote by $\boldsymbol{H}_{[1..n-1]}$ the submatrix of $\boldsymbol{H}_\alpha$ that consists of the first $n-1$ rows and the first $n-1$ columns. It follows from (3.2) that $\|\overline{\boldsymbol{H}}_{[1..n-1]} - \boldsymbol{H}_{[1..n-1]}\|_F \leq \varepsilon_3$.

First, we give explicit formulae for the F-norm of $\boldsymbol{H}_{[1..n-1]}$ and $\boldsymbol{H}^{-1}_{[1..n-1]}$; see Appendix A for the proof.

Lemma 3.3. *Let the notation be as above. Then*

$$\|\boldsymbol{H}^{-1}_{[1..n-1]}\|_F^2 = (n-2) + \frac{\|\boldsymbol{\alpha}\|^2}{\alpha_n^2},$$
$$\|\boldsymbol{H}_{[1..n-1]}\|_F^2 = (n-2) + \frac{\alpha_n^2}{\|\boldsymbol{\alpha}\|^2}.$$

The following lemma enables us to give an estimation on $\|\overline{\boldsymbol{H}}^{-1}_{[1..n-1]}\|_F$.

Lemma 3.4 ([15, Theorem 2.3.4])**.** *Let $\boldsymbol{A}$ be a nonsingular matrix with perturbation $\boldsymbol{E}$. Let $\|.\|$ denote any matrix norm satisfying inequality $\|\boldsymbol{BC}\| \leq \|\boldsymbol{B}\|\|\boldsymbol{C}\|$ for any matrices $\boldsymbol{B}$ and $\boldsymbol{C}$. If $\|\boldsymbol{EA}^{-1}\| < 1$, then $\boldsymbol{A}+\boldsymbol{E}$ is nonsingular, and it holds that*

$$\|(\boldsymbol{A}+\boldsymbol{E})^{-1} - \boldsymbol{A}^{-1}\| \leq \frac{\|\boldsymbol{EA}^{-1}\|}{1 - \|\boldsymbol{EA}^{-1}\|}\|\boldsymbol{A}^{-1}\|.$$

Applying the above lemma to $\boldsymbol{H}_{[1..n-1]}$ yields the following corollary.

Corollary 3.5. *Let $\overline{\boldsymbol{H}}_\alpha = \boldsymbol{H}_\alpha + \Delta\boldsymbol{H}_\alpha$ and $\|\Delta\boldsymbol{H}_\alpha\|_F < \varepsilon_3$, $\boldsymbol{H}_{[1..n-1]}$ and $\overline{\boldsymbol{H}}_{[1..n-1]}$ denote submatrices consisting of the first $(n-1)$ rows and the first $(n-1)$ columns of $\boldsymbol{H}_\alpha$ and $\overline{\boldsymbol{H}}_\alpha$, respectively. When $\varepsilon_3 < \frac{1}{\|\boldsymbol{H}^{-1}_{[1..n-1]}\|_F}$, $\overline{\boldsymbol{H}}_{[1..n-1]}$ is nonsingular and it holds that*

$$\|\overline{\boldsymbol{H}}^{-1}_{[1..n-1]}\|_F \leq \frac{1}{1 - \varepsilon_3\|\boldsymbol{H}^{-1}_{[1..n-1]}\|_F}\|\boldsymbol{H}^{-1}_{[1..n-1]}\|_F.$$

Proof. When $\varepsilon_3 < \frac{1}{\|\boldsymbol{H}^{-1}_{[1..n-1]}\|_F}$, it holds that

$$\begin{aligned}\|\Delta \boldsymbol{H}_{[1..n-1]}\boldsymbol{H}^{-1}_{[1..n-1]}\|_F &\le \|\Delta \boldsymbol{H}_{[1..n-1]}\|_F \|\boldsymbol{H}^{-1}_{[1..n-1]}\|_F \\ &\le \|\Delta \boldsymbol{H}_\alpha\|_F \cdot \|\boldsymbol{H}^{-1}_{[1..n-1]}\|_F \\ &< \varepsilon_3 \cdot \|\boldsymbol{H}^{-1}_{[1..n-1]}\|_F < 1.\end{aligned}$$

From Lemma 3.4, $\overline{\boldsymbol{H}}_{[1..n-1]}$ is nonsingular and it follows that

$$\begin{aligned}\|\overline{\boldsymbol{H}}^{-1}_{[1..n-1]}\|_F &< \frac{1}{1-\|\Delta \boldsymbol{H}_{[1..n-1]}\boldsymbol{H}^{-1}_{[1..n-1]}\|_F}\|\boldsymbol{H}^{-1}_{[1..n-1]}\|_F \\ &\le \frac{1}{1-\varepsilon_3\|\boldsymbol{H}^{-1}_{[1..n-1]}\|_F}\|\boldsymbol{H}^{-1}_{[1..n-1]}\|_F.\end{aligned}$$

This completes the proof. □

Corollary 3.5 shows that when $\varepsilon_3 < 1/\|\boldsymbol{H}^{-1}_{[1..n-1]}\|_F$, it holds that $\overline{h}_{i,i} \neq 0$ for $i = 1, \ldots, n-1$. Denote by $\overline{\boldsymbol{\alpha}} = (\overline{\alpha}_1, \ldots, \overline{\alpha}_n)$ a unit real vector satisfying $\overline{\boldsymbol{\alpha}}\overline{\boldsymbol{H}}_\alpha = 0$. Without loss of generality, we assume that $\overline{\alpha}_n \neq 0$. Otherwise we can deduce $\overline{\boldsymbol{\alpha}} = \mathbf{0}$, which contradicts the fact that $\overline{\boldsymbol{\alpha}}$ is a unit vector. (In fact, since $\bar{\alpha}_{n-1}\bar{h}_{n-1,n-1} + \bar{\alpha}_n\bar{h}_{n,n-1} = 0$ and $\overline{h}_{n-1,n-1} \neq 0$ we have that $\bar{\alpha}_n = 0$ implies $\bar{\alpha}_{n-1} = 0$. Similarly, $\bar{\alpha}_i = 0$ for $i = 1, 2, \ldots, n-2$.) Moreover, we can choose vector $\overline{\boldsymbol{\alpha}}$ with $\overline{\alpha}_n > 0$. Next, we give a nonzero lower bound on $\overline{\alpha}_n$.

Lemma 3.6. *Let $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_{n-1}, 1)$ be a real vector with $\|\boldsymbol{\xi}\| \le M$, and let $\boldsymbol{\beta} = \frac{\boldsymbol{\xi}}{\|\boldsymbol{\xi}\|} = (\beta_1, \ldots, \beta_{n-1}, \beta_n)$. Then it holds that $|\beta_n| \ge \frac{1}{M}$.*

Proof. According to assumptions,

$$1 = \|\boldsymbol{\beta}\| = \left|\beta_n\left(\frac{\beta_1}{\beta_n}, \ldots, \frac{\beta_{n-1}}{\beta_n}, 1\right)\right| = \|\beta_n\boldsymbol{\xi}\| \le |\beta_n| \cdot \|\boldsymbol{\xi}\| \le |\beta_n| M.$$

The proof of lemma is finished. □

The above lemma enables us to give a lower bound of some component of a unit vector.

Lemma 3.7. *Let $\overline{\boldsymbol{\alpha}} = (\overline{\alpha}_1, \ldots, \overline{\alpha}_{n-1}, \overline{\alpha}_n)$ be a unit vector such that $\overline{\boldsymbol{\alpha}}\overline{\boldsymbol{H}}_\alpha = 0$. If ε_3 given in* (3.2) *is less than $\frac{\alpha_n}{\sqrt{(n-2)\alpha_n^2+1}}$, then*

$$|\overline{\alpha}_n| \ge \frac{\alpha_n}{2\sqrt{1-\alpha_n^2}\sqrt{(n-2)\alpha_n^2+1} + 2\alpha_n}.$$

Proof. Consider the linear system $(x_1, \ldots, x_n)\overline{\boldsymbol{H}}_\alpha = 0$ with unknowns x_i for $i = 1, 2, \ldots, n$. Since the rank of $\overline{\boldsymbol{H}}_\alpha$ is at most $n-1$, we can assume that $x_n = 1$; then it reduces to the following system:

$$(x_1, \ldots, x_{n-1})\overline{\boldsymbol{H}}_{[1..n-1]} = -(\overline{h}_{n,1}, \ldots, \overline{h}_{n,n-1}).$$

If $\varepsilon_3 < \frac{\alpha_n}{2\sqrt{(n-2)\alpha_n^2+1}}$, then $\overline{\boldsymbol{H}}_{[1..n-1]}$ is nonsingular by Lemma 3.3 and Corollary 3.5, so $(x_1, \dots, x_{n-1}) = -(\overline{h}_{n,1}, \dots, \overline{h}_{n,n-1})\overline{\boldsymbol{H}}_{[1..n-1]}^{-1}$. Hence, it holds that

$$\begin{aligned}
\|(x_1, \dots, x_{n-1})\|_2 &\le \|(\overline{h}_{n,1}, \dots, \overline{h}_{n,n-1})\|_2 \|\overline{\boldsymbol{H}}_{[1..n-1]}^{-1}\|_2 \\
&\le \|(\overline{h}_{n,1}, \dots, \overline{h}_{n,n-1})\|_2 \|\overline{\boldsymbol{H}}_{[1..n-1]}^{-1}\|_F \\
&\le (\|(h_{n,1}, \dots, h_{n,n-1})\|_2 + \varepsilon_3)\|\overline{\boldsymbol{H}}_{[1..n-1]}^{-1}\|_F \\
&\le \left(\sqrt{1-\alpha_n^2} + \frac{\alpha_n}{2\sqrt{(n-2)\alpha_n^2+1}}\right) 2\|\boldsymbol{H}_{[1..n-1]}^{-1}\|_F \\
&= 2\sqrt{1-\alpha_n^2}\|\boldsymbol{H}_{[1..n-1]}^{-1}\|_F + 1 \\
&= \frac{2\sqrt{1-\alpha_n^2}\sqrt{(n-2)\alpha_n^2+1}}{\alpha_n} + 1.
\end{aligned}$$

Thus, it is obtained that

$$\begin{aligned}
\|(x_1, \dots, x_{n-1}, x_n)\|_2 &\le \|(x_1, \dots, x_{n-1})\|_2 + 1 \\
&\le \frac{2\sqrt{1-\alpha_n^2}\sqrt{(n-2)\alpha_n^2+1}}{\alpha_n} + 2 \\
&= \frac{2\sqrt{1-\alpha_n^2}\sqrt{(n-2)\alpha_n^2+1} + 2\alpha_n}{\alpha_n}.
\end{aligned}$$

From Lemma 3.6, it follows that

$$|\overline{\alpha}_n| \ge \frac{1}{\|(x_1, x_2, \dots, x_n)\|} \ge \frac{\alpha_n}{2\sqrt{1-\alpha_n^2}\sqrt{(n-2)\alpha_n^2+1} + 2\alpha_n}.$$

The proof of the lemma is finished. ∎

We now give the main theorem of this paper, which can be seen as a forward error analysis of $\texttt{PSLQ}_\varepsilon$ for the perturbation introduced in (3.2).

Theorem 3.8. *Given a real vector $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)$, let $\boldsymbol{H}_\alpha$ be the hyperplane matrix constructed as in* (2.1). *Let $\overline{\boldsymbol{H}}_\alpha$ be an approximate matrix of $\boldsymbol{H}_\alpha$ with $\|\boldsymbol{H}_\alpha - \overline{\boldsymbol{H}}_\alpha\|_F < \varepsilon_3 < \frac{\alpha_n}{2\sqrt{(n-2)\alpha_n^2+1}}$. Let $\boldsymbol{A}$ be the unimodular matrix, and let $\boldsymbol{Q}$ be the orthogonal matrix such that $\boldsymbol{H} = (h_{i,j}) = \boldsymbol{A}\overline{\boldsymbol{H}}_\alpha\boldsymbol{Q}$ is a lower trapezoidal matrix at the termination of $\texttt{PSLQ}_\varepsilon$ with $|h_{n,n-1}| < \varepsilon_2$. Let $\boldsymbol{m}$ denote the $(n-1)$th column of $\boldsymbol{A}^{-1}$. Then*

$$|\langle \boldsymbol{\alpha}, \boldsymbol{m}\rangle| < C \cdot (\|\boldsymbol{m}\|\varepsilon_3 + \alpha_n\varepsilon_2),$$

where $C = \frac{2(\sqrt{(n-2)\alpha_n^2+1}+\alpha_n)}{\alpha_n}$.

Proof. Suppose that $\texttt{PSLQ}_\varepsilon$ returns $\boldsymbol{m}$ with $\overline{\boldsymbol{H}}_\alpha$ as the hyperplane matrix, when $|h_{n,n-1}| < \varepsilon_2$. Then this process can be seen as running $\texttt{PSLQ}_\varepsilon$ for a unit vector $\overline{\boldsymbol{\alpha}}$ satisfying $\overline{\boldsymbol{\alpha}}\overline{\boldsymbol{H}}_\alpha = 0$. According to Theorem 3.1, we have $|\langle \overline{\boldsymbol{\alpha}}, \boldsymbol{m}\rangle| \le \varepsilon_2$. Now we consider the system

$$\overline{\boldsymbol{H}}_\alpha \boldsymbol{c} = \boldsymbol{m} + (0, 0, \dots, 0, b)^T, \tag{3.3}$$

where $\boldsymbol{c} = (c_1, c_2, \dots, c_{n-1})^T$ is the unknown vector. We have that

$$0 = \overline{\boldsymbol{\alpha}}\overline{\boldsymbol{H}}_\alpha\boldsymbol{c} = \langle \overline{\boldsymbol{\alpha}}, \boldsymbol{m}\rangle + \overline{\boldsymbol{\alpha}}_n b.$$

Hence $\overline{\alpha}_n b = -\langle \overline{\boldsymbol{\alpha}}, \boldsymbol{m}\rangle$, and we have

$$|b| < \frac{\varepsilon_2}{\overline{\alpha}_n}.$$

From (3.3) it holds that

$$\begin{aligned}
|\langle \boldsymbol{\alpha}, \boldsymbol{m}\rangle| &= \left|\boldsymbol{\alpha}\overline{\boldsymbol{H}}_\alpha \boldsymbol{c} - \alpha_n b\right| \leq \left|\boldsymbol{\alpha}\overline{\boldsymbol{H}}_\alpha \boldsymbol{c}\right| + |\alpha_n||b| \\
&\leq \left|\boldsymbol{\alpha}(\overline{\boldsymbol{H}}_\alpha - \boldsymbol{H}_\alpha)\boldsymbol{c}\right| + |\alpha_n||b| \\
&\leq \|\boldsymbol{\alpha}\|\|\overline{\boldsymbol{H}}_\alpha - \boldsymbol{H}_\alpha\|_2 \,\|\boldsymbol{c}\| + |\alpha_n||b| \\
&\leq \|\boldsymbol{\alpha}\|\|\overline{\boldsymbol{H}}_\alpha - \boldsymbol{H}_\alpha\|_2 \,\|\boldsymbol{c}\| + |\alpha_n||b| \\
&\leq \|\boldsymbol{\alpha}\|\,\|\boldsymbol{c}\|\,\varepsilon_3 + \frac{|\alpha_n|}{|\overline{\alpha}_n|}\varepsilon_2.
\end{aligned}$$

Since $\varepsilon_3 < \frac{\alpha_n}{2\sqrt{(n-2)\alpha_n^2+1}}$, by Lemma 3.7, it follows that

$$|\langle \boldsymbol{\alpha}, \boldsymbol{m}\rangle| < \|\boldsymbol{\alpha}\| \cdot \|\boldsymbol{c}\|\,\varepsilon_3 + 2(\sqrt{1-\alpha_n^2}\sqrt{(n-2)\alpha_n^2+1} + \alpha_n)\varepsilon_2. \tag{3.4}$$

The first $n-1$ equations of (3.3) give a square system

$$\overline{\boldsymbol{H}}_{[1..n-1]} \begin{pmatrix} c_1 \\ \vdots \\ c_{n-1} \end{pmatrix} = \begin{pmatrix} m_1 \\ \vdots \\ m_{n-1} \end{pmatrix}.$$

Then it is obtained that

$$\|\boldsymbol{c}\| \leq \|\overline{\boldsymbol{H}}_{[1..n-1]}^{-1}\|_2 \left\|\begin{pmatrix} m_1 \\ \vdots \\ m_{n-1} \end{pmatrix}\right\|_2 \leq \|\overline{\boldsymbol{H}}_{[1..n-1]}^{-1}\|_F \|\boldsymbol{m}\|.$$

Since $\varepsilon_3 < \frac{\alpha_n}{2\sqrt{(n-2)\alpha_n^2+1}}$, by Corollary 3.5 we have

$$\begin{aligned}
\|\boldsymbol{c}\| &\leq \frac{1}{1 - \varepsilon_3\|\boldsymbol{H}_{[1..n-1]}^{-1}\|_F}\|\boldsymbol{H}_{[1..n-1]}^{-1}\|_F\|\boldsymbol{m}\|_2 \\
&< 2\|\boldsymbol{H}_{[1..n-1]}^{-1}\|_F\|\boldsymbol{m}\|_2 < \frac{2\sqrt{(n-2)\alpha_n^2+1}}{\alpha_n}\|\boldsymbol{m}\|_2.
\end{aligned}$$

Substituting the above inequality into (3.4) yields

$$\begin{aligned}
|\langle \boldsymbol{\alpha}, \boldsymbol{m}\rangle| &< \|\boldsymbol{\alpha}\|\,\|\boldsymbol{c}\|\,\varepsilon_3 + 2(\sqrt{1-\alpha_n^2}\sqrt{(n-2)\alpha_n^2+1} + \alpha_n)\varepsilon_2 \\
&< \frac{2\sqrt{(n-2)\alpha_n^2+1}}{\alpha_n}\|\boldsymbol{m}\|\varepsilon_3 + 2(\sqrt{1-\alpha_n^2}\sqrt{(n-2)\alpha_n^2+1} + \alpha_n)\varepsilon_2 \\
&< \frac{2(\sqrt{(n-2)\alpha_n^2+1} + \alpha_n)}{\alpha_n}\|\boldsymbol{m}\|\varepsilon_3 + 2(\sqrt{(n-2)\alpha_n^2+1} + \alpha_n)\varepsilon_2 \\
&= \frac{2(\sqrt{(n-2)\alpha_n^2+1} + \alpha_n)}{\alpha_n}(\|\boldsymbol{m}\|\varepsilon_3 + \alpha_n\varepsilon_2).
\end{aligned}$$

The theorem is proved. □

Although the quantity $|\langle \boldsymbol{\alpha}, \boldsymbol{m}\rangle|$ is usually nonzero for empirical data, it somewhat measures how close it is from $\boldsymbol{m}$ to a true integer relation for $\boldsymbol{\alpha}$. So it can be seen as output error. In this sense, Theorem 3.8 says that if a perturbation of the input $\boldsymbol{H}_\alpha$ is small enough, then the "output error" of $\texttt{PSLQ}_\varepsilon$ can also be small. Roughly speaking, if we fix the termination condition ε_2 to be a tiny number, then the "output error" is amplified by a factor at most $C \cdot \|\boldsymbol{m}\|$.

4. $\texttt{PSLQ}_\varepsilon$ WITH EMPIRICAL DATA

Aiming to obtain $\boldsymbol{m}$ by $\texttt{PSLQ}_\varepsilon$ such that $|\langle \boldsymbol{\alpha}, \boldsymbol{m}\rangle| < \varepsilon$, we study how to determine the error control parameters ε_1, ε_2, and ε_3 in this section.

4.1. Error control of $\texttt{PSLQ}_\varepsilon$.

Lemma 4.1. *Let $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)$ be an n-dimensional unit vector with $|\alpha_n| = \max_i\{|\alpha_i|\}$, and let $\bar{\boldsymbol{\alpha}}$ be its approximation. Construct $\boldsymbol{H}_\alpha$ and $\boldsymbol{H}_{\bar{\alpha}}$ as in* (2.1) *for $\boldsymbol{\alpha}$ and $\bar{\boldsymbol{\alpha}}$, respectively. If $\|\boldsymbol{\alpha} - \bar{\boldsymbol{\alpha}}\| < \frac{1}{8n}$, then it holds that*

$$\|\boldsymbol{H}_\alpha - \boldsymbol{H}_{\bar{\alpha}}\|_F < 8n^{\frac{3}{2}}\|\boldsymbol{\alpha} - \overline{\boldsymbol{\alpha}}\|.$$

Proof. Let $s_i = \sqrt{\sum_{k=i}^n \alpha_k^2}$, let $\bar{s}_i = \sqrt{\sum_{k=i}^n \bar{\alpha}_k^2}$, let $\boldsymbol{b}_i = (0, \dots, 0, \alpha_i, \dots, \alpha_n)$, and let $\bar{\boldsymbol{b}}_i = (0, \dots, 0, \bar{\alpha}_i, \dots, \bar{\alpha}_n)$. It obviously holds that $\|\boldsymbol{b}_i - \bar{\boldsymbol{b}}_i\| \le \|\boldsymbol{\alpha} - \bar{\boldsymbol{\alpha}}\|$. So, it is obtained that $|s_i - \bar{s}_i| = |\|\boldsymbol{b}_i\| - \|\bar{\boldsymbol{b}}_i\|| \le \|\boldsymbol{b}_i - \bar{\boldsymbol{b}}_i\| \le \|\boldsymbol{\alpha} - \bar{\boldsymbol{\alpha}}\|$. By the way, from $|\alpha_n| = \max_i\{|\alpha_i|\}$ and $\|\boldsymbol{\alpha}\| = 1$, it follows that $|\alpha_n| \ge \frac{1}{\sqrt{n}}$. Thus, if $\|\boldsymbol{\alpha} - \bar{\boldsymbol{\alpha}}\| < \frac{1}{2\sqrt{n}}$, then it holds that $|\bar{\alpha}_n| > \frac{1}{2\sqrt{n}}$.

Recall that $\boldsymbol{H}_\alpha = (h_{i,j})$ and

$$h_{i,j} = \begin{cases} \frac{s_{i+1}}{s_i} & \text{if } i = j, \\ -\frac{\alpha_i \alpha_j}{s_j s_{j+1}} & \text{else if } i > j, \\ 0 & \text{otherwise.} \end{cases}$$

Let us consider the error of $\frac{s_{i+1}}{s_i}$:

$$\begin{aligned} \left|\frac{s_{i+1}}{s_i} - \frac{\bar{s}_{i+1}}{\bar{s}_i}\right| &= \left|\frac{s_{i+1}\bar{s}_i - s_i\bar{s}_{i+1}}{s_i\bar{s}_i}\right| = \left|\frac{s_{i+1}\bar{s}_i - s_i s_{i+1} + s_i s_{i+1} - s_i\bar{s}_{i+1}}{s_i\bar{s}_i}\right| \\ &\le \frac{s_{i+1}|s_i - \bar{s}_i|}{s_i\bar{s}_i} + \frac{s_i|s_{i+1} - \bar{s}_{i+1}|}{s_i\bar{s}_i} \le \frac{|s_i - \bar{s}_i|}{\bar{s}_i} + \frac{|s_{i+1} - \bar{s}_{i+1}|}{\bar{s}_i} \\ &\le \frac{2}{\bar{s}_i}\|\boldsymbol{\alpha} - \bar{\boldsymbol{\alpha}}\| \le \frac{2}{|\bar{\alpha}_n|}\|\boldsymbol{\alpha} - \bar{\boldsymbol{\alpha}}\| \le 4\sqrt{n}\|\boldsymbol{\alpha} - \bar{\boldsymbol{\alpha}}\|, \end{aligned}$$

and then consider the error of $\frac{\alpha_i\alpha_j}{s_j s_{j+1}} (i > j)$:

$$\begin{aligned} &\left|\frac{\alpha_i\alpha_j}{s_j s_{j+1}} - \frac{\bar{\alpha}_i\bar{\alpha}_j}{\bar{s}_j\bar{s}_{j+1}}\right| = \frac{|\alpha_i\alpha_j\bar{s}_j\bar{s}_{j+1} - \bar{\alpha}_i\bar{\alpha}_j s_j s_{j+1}|}{s_j s_{j+1}\bar{s}_j\bar{s}_{+1}} \\ &\le \frac{1}{s_j s_{j+1}\bar{s}_j\bar{s}_{j+1}}(|\alpha_i\alpha_j\bar{s}_j\bar{s}_{j+1} - \alpha_i\alpha_j s_j\bar{s}_{j+1}| + |\alpha_i\alpha_j s_j\bar{s}_{j+1} - \alpha_i\alpha_j s_j s_{j+1}| \\ &+ |\alpha_i\alpha_j s_j s_{j+1} - \bar{\alpha}_i\alpha_j s_j s_{j+1}| + |\bar{\alpha}_i\alpha_j s_j s_{j+1} - \bar{\alpha}_i\bar{\alpha}_j s_j s_{j+1}|) \\ &= \frac{\alpha_i\alpha_j\bar{s}_{j+1}}{s_j s_{j+1}\bar{s}_j\bar{s}_{j+1}}|\bar{s}_j - s_j| + \frac{\alpha_i\alpha_j s_j}{s_j s_{j+1}\bar{s}_j\bar{s}_{j+1}}|\bar{s}_{j+1} - s_{j+1}| \\ &+ \frac{\alpha_j s_j s_{j+1}}{s_j s_{j+1}\bar{s}_j\bar{s}_{j+1}}|\alpha_i - \bar{\alpha}_i| + \frac{\bar{\alpha}_i s_j s_{j+1}}{s_j s_{j+1}\bar{s}_j\bar{s}_{j+1}}|\alpha_j - \bar{\alpha}_j| \\ &\le \frac{|\bar{s}_j - s_j|}{\bar{s}_j} + \frac{|\alpha_j|}{\bar{s}_j}\frac{|\bar{s}_{j+1} - s_{j+1}|}{\bar{s}_{j+1}} + \frac{|\alpha_j|}{\bar{s}_j}\frac{|\alpha_i - \bar{\alpha}_i|}{\bar{s}_{j+1}} + \frac{|\alpha_j - \bar{\alpha}_j|}{\bar{s}_j}. \end{aligned} \tag{4.1}$$

We need to estimate $\frac{|\alpha_j|}{\bar{s}_j}$. First, if $|\alpha_j| \leq |\bar{\alpha}_j|$, then it holds that $\frac{|\alpha_j|}{\bar{s}_j} \leq 1$. When $|\alpha_j| > |\bar{\alpha}_j|$, it follows that

$$\bar{s}_j^2 = \bar{\alpha}_j^2 + \cdots + \bar{\alpha}_n^2 = \alpha_j^2 + 2\Delta\alpha_j\alpha_j + \Delta\alpha_j^2 + \bar{\alpha}_{j+1}^2 + \cdots + \bar{\alpha}_n^2,$$

so we have

$$\bar{s}_j^2 - \alpha_j^2 \geq \sum_{k=j+1}^{n} \bar{\alpha}_k^2 - 2|\Delta\alpha_j||\alpha_j| \geq \bar{\alpha}_n^2 - 2|\Delta\alpha_j|.$$

Note that $|\bar{\alpha}_n| > \frac{1}{2\sqrt{n}}$ and $|\Delta\alpha_j| < \frac{1}{8n}$ when $\|\boldsymbol{\alpha}-\bar{\boldsymbol{\alpha}}\| < \frac{1}{8n}$, which indicate $\bar{s}_j^2-\alpha_j^2 \geq \bar{\alpha}_n^2 - 2|\Delta\alpha_j| > \frac{1}{4n} - \frac{2}{8n} = 0$. So it is proved that

$$\frac{|\alpha_j|}{\bar{s}_j} \leq 1 \tag{4.2}$$

when $\|\boldsymbol{\alpha} - \bar{\boldsymbol{\alpha}}\| < \frac{1}{8n}$. Applying (4.2) to (4.1) gives

$$\begin{aligned}\left|\frac{\alpha_i\alpha_j}{s_j s_{j+1}} - \frac{\bar{\alpha}_i\bar{\alpha}_j}{\bar{s}_j\bar{s}_{j+1}}\right| &\leq \frac{|\bar{s}_j - s_j|}{\bar{s}_j} + \frac{|\alpha_j|}{\bar{s}_j}\frac{|\bar{s}_{j+1} - s_{j+1}|}{\bar{s}_{j+1}} + \frac{|\alpha_j|}{\bar{s}_j}\frac{|\alpha_i - \bar{\alpha}_i|}{\bar{s}_{j+1}} + \frac{|\alpha_j - \bar{\alpha}_j|}{\bar{s}_j}\\ &\leq \frac{|\bar{s}_j - s_j|}{\bar{s}_j} + \frac{|\bar{s}_{j+1} - s_{j+1}|}{\bar{s}_{j+1}} + \frac{|\alpha_i - \bar{\alpha}_i|}{\bar{s}_{j+1}} + \frac{|\alpha_j - \bar{\alpha}_j|}{\bar{s}_j}\\ &\leq \frac{4}{\frac{1}{2\sqrt{n}}}\|\boldsymbol{\alpha} - \bar{\boldsymbol{\alpha}}\| = 8\sqrt{n}\|\boldsymbol{\alpha} - \bar{\boldsymbol{\alpha}}\|.\end{aligned}$$

With the assumption of $\|\boldsymbol{\alpha} - \bar{\boldsymbol{\alpha}}\| < \frac{1}{8n}$, it follows that

$$\|\boldsymbol{H}_\alpha - \boldsymbol{H}_{\bar{\alpha}}\|_F \leq 8\sqrt{n}\sqrt{\frac{n(n-1)}{2} + (n-1)}\|\boldsymbol{\alpha} - \bar{\boldsymbol{\alpha}}\| \leq 8n^{3/2}\|\boldsymbol{\alpha} - \bar{\boldsymbol{\alpha}}\|.$$

The proof is finished. □

Now we construct the input $\overline{\boldsymbol{H}}_\alpha$ of $\texttt{PSLQ}_\varepsilon$ from empirical data $\bar{\boldsymbol{\alpha}}$. In this paper, we restrict ourselves under exact arithmetic, i.e., we take $\overline{\boldsymbol{H}}_\alpha = \boldsymbol{H}_{\bar{\alpha}}$. Applying this to Theorem 3.8 yields the following particular error control strategy.

Theorem 4.2. *Let $\boldsymbol{\alpha} \in \mathbb{R}^n$ be a unit vector with $|\alpha_n| = \max_i\{|\alpha_i|\}$, and let $\varepsilon > 0$. Suppose $\boldsymbol{\alpha}$ has an integer relation with 2-norm bounded from above by M. Given empirical data $\bar{\boldsymbol{\alpha}}$ with*

$$\|\boldsymbol{\alpha} - \bar{\boldsymbol{\alpha}}\| < \varepsilon_1 < \frac{\varepsilon}{16MCn^{3/2}},$$

if $\texttt{PSLQ}_\varepsilon$ with

$$\varepsilon_2 < \frac{\varepsilon}{2C\alpha_n}$$

returns $\boldsymbol{m}$ with $\|\boldsymbol{m}\| < M$, then $|\langle\boldsymbol{\alpha}, \boldsymbol{m}\rangle| < \varepsilon$, where $C = \frac{2(\sqrt{(n-2)\alpha_n^2+1}+\alpha_n)}{\alpha_n}$ and $M > 0$.

Proof. From Lemma 4.1, it holds that

$$\|\overline{\boldsymbol{H}}_\alpha - \boldsymbol{H}_\alpha\|_F = \|\boldsymbol{H}_{\bar{\alpha}} - \boldsymbol{H}_\alpha\|_F < \frac{\varepsilon}{2M \cdot C}.$$

Then Theorem 3.8 implies

$$|\langle\boldsymbol{\alpha}, \boldsymbol{m}\rangle| < C\left(M\frac{\varepsilon}{2M \cdot C} + \alpha_n\varepsilon_2\right) < \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon. \tag{4.3}$$

The theorem is proved. □

4.2. **Some remarks.** It should be noted that the results presented in Theorems 3.1, 3.8, and 4.2 can be applied not only to the standard PSLQ algorithm, but also to the multipair variant of PSLQ [9, Section 6]. The reason is that all the proofs of these theorems are independent of the swap strategy. The multipair variant can be seen as a parallel version of PSLQ, in which several pairs of rows of the matrix $\boldsymbol{H}$ are swaped simultaneously, and it is much more efficient than the standard PSLQ and hence utilized in almost all of the applications in practice. We also note that the iteration bound in Theorem 3.2 may not hold for the multipair variant; we refer to [9, page 1729] and [12, Section 3] for this topic.

It is not difficult to verify that Theorem 4.2 still holds for $\varepsilon_1 < \frac{\omega\varepsilon}{8MCn^{3/2}}$ and $\varepsilon_2 < \frac{(1-\omega)\varepsilon}{C\alpha_n}$ for any $0 < \omega < 1$. The error control strategy given in Theorem 4.2 just simply takes $\omega = 1/2$. Examples in the next section show the effectiveness of this strategy, but the optimal choice for ω is beyond the scope of this paper.

Figure 1 shows the relationships among the main notation of this paper. In this figure, the solid lines indicate the routine of $\mathtt{PSLQ}_\varepsilon$ for empirical input data $\bar{\boldsymbol{\alpha}}$ with $\|\boldsymbol{\alpha} - \bar{\boldsymbol{\alpha}}\| < \varepsilon_1$. According to Theorem 4.2, if the returned $\boldsymbol{m}$ by $\mathtt{PSLQ}_\varepsilon$ satisfies $\|\boldsymbol{m}\| < M$, then we can guarantee that $|\langle \boldsymbol{m}, \boldsymbol{\alpha}\rangle| < \varepsilon$.

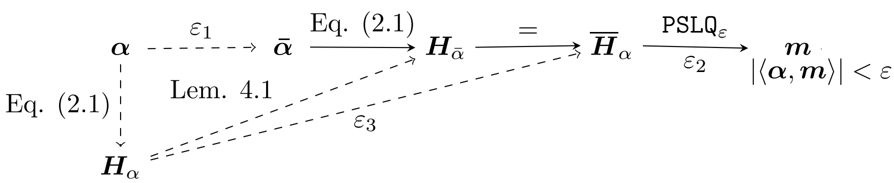

FIGURE 1. An illustrative picture of relationships among the main notation

As mentioned previously, high precision arithmetic must be used for almost all applications of PSLQ. In practice, Bailey (see, e.g., [5]) suggested that if one wishes to recover a relation for an n-dimensional vector, with coefficients of maximum size $\log_{10} G$ decimal digits, then the input vector $\boldsymbol{\alpha}$ must be specified to at least $n \log_{10} G$ digits, and one must employ floating-point arithmetic accurate to at least $n \log_{10} G$ digits. However, there seems no theoretical results about how to decide the precision generally. Theorems 3.8 and 4.2 in this paper can be seen as theoretical sufficient conditions for PSLQ with empirical input data. We show in the next subsection that these theoretical results indeed give some effective strategies for the input data precision and the termination condition in practice.

4.3. **Numerical examples.** In this subsection, we give some examples to illustrate our strategy of error control based on Theorem 4.2. We use our own implementation of $\mathtt{PSLQ}_\varepsilon$ in Maple, which takes the running precision `Digits`, a target accuracy ε, and an upper bound on the coefficients of the expected relation G as its input. We use $M = \sqrt{n}G$ as its 2-norm bound and fix `Digits := 200` and `Digits := 600` for the first two examples and the third example, respectively, so that it is sufficient to guarantee the correctness and that it can mimic the exact real arithmetic.

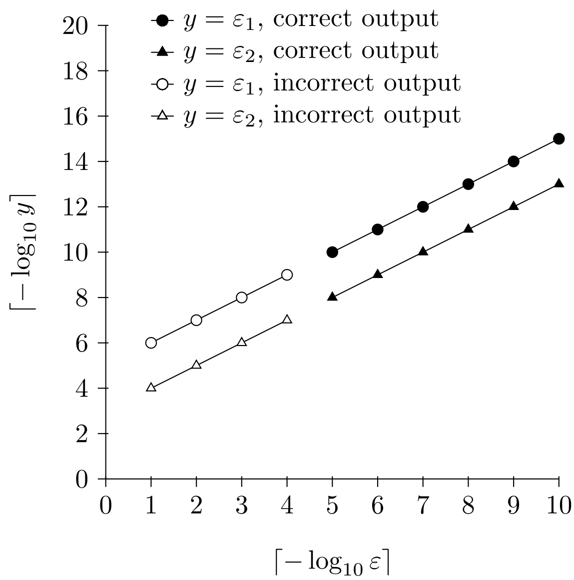

FIGURE 2. Error control strategy for Example 4.3

Example 4.3 (Transcendental numbers)**.** Equation (69) of [6] states that $\boldsymbol{\beta} = (t, 1, \ln 2, \ln^2 2, \pi^2) \in \mathbb{R}^5$ has an integer relation $\boldsymbol{m} = (1, -5, 4, -16, 1)$, where

$$t = \int_0^1 \int_0^1 \left(\frac{x-1}{x+1}\right)^2 \left(\frac{y-1}{y+1}\right)^2 \left(\frac{xy-1}{xy+1}\right)^2 dxdy.$$

We try to recover this relation for $\boldsymbol{\alpha} = \boldsymbol{\beta}/\|\boldsymbol{\beta}\|$.

Because of involving transcendental numbers, we can only obtain the empirical data of $\boldsymbol{\alpha}$. Suppose that the maximum of the coefficients is bounded by $G = 16$ and that the gap bound for this example is 10^{-6}. (In fact, by exhaustive search, we can obtain a gap bound that is about 6.37×10^{-6}.) Thus, the target precision ε is set as $\varepsilon = 10^{-5}$. It means that we want to find an integer vector $\boldsymbol{m}$ such that $|\langle \boldsymbol{\alpha}, \boldsymbol{m} \rangle| < \varepsilon = 10^{-5}$. According to Theorem 4.2, we obtain that $\varepsilon_1 \approx 2.60 \times 10^{-11}$ and $\varepsilon_2 \approx 8.39 \times 10^{-8}$. We run this example in the computer algebra system `Maple`. After 30 iterations of `PSLQ`, the procedure returns a relation $\boldsymbol{m} = (1, -5, 4, -16, 1)$, which is an exact integer relation for $\boldsymbol{\alpha}$.

If we do not know a gap bound on $|\langle \boldsymbol{m}, \boldsymbol{\alpha} \rangle|$, we can test $\varepsilon = 10^{-i}$ for $i = 1, 2, \ldots, 10$, where the corresponding ε_1 and ε_2 are decided according to Theorem 4.2. As shown in Figure 2, for $i = 1, 2, 3, 4$, no correct answer is obtained, but for $5 \leq i \leq 10$ the procedure always returns the same relation $\boldsymbol{m}$. Further, the difference between $\lceil -\log_{10} \varepsilon_1 \rceil$ and $\lceil -\log_{10} \varepsilon_2 \rceil$ does not change for different ε.

Bailey's estimation is $\lceil n \log_{10} G \rceil = 7$ decimal digits that indicates $\varepsilon_1 < 10^{-7}$, which is relatively compact for the above setting. However, Bailey's estimation still has the following drawbacks. For one thing, Bailey's estimation does not suggest when the algorithm terminates, i.e., how to choose ε_2, while Theorem 4.2 suggests the quantity that ε_2 should be larger than ε_1. This is consistent with intuition: the error would be amplified by exact computation with empirical data as input.

In fact, if we do not have the error control strategy as indicated by Theorem 4.2, we can only use a trial-and-error approach to decide the termination precision ε_2, since the procedure may miss the correct answer for an incorrect ε_2, even with relatively high precision.

For another thing, if we do not know such a tight bound on the maximum coefficient of the relation, instead, for example, we only know $G \leq 10^5$. For the same ε, we now have $\varepsilon_1 \approx 4.16\times10^{-15}$ and $\varepsilon_2 \approx 8.39\times10^{-8}$, for which our procedure work correctly, while at least $\lceil n\log_{10} G\rceil = 25$ decimal digits is needed according to Bailey's estimation, which implies $\varepsilon_1 \leq 10^{-25}$. For this example, by Bailey's estimation, $\lceil -\log_{10}\varepsilon_1\rceil$ increases linearly with $\lceil\log_{10} G\rceil$, whose slope is $n = 5$. According to Theorem 4.2, $\lceil -\log_{10}\varepsilon_1\rceil$ also increases linearly with $\lceil\log_{10} G\rceil$, but the slope is only about 1. In fact, according to Theorem 4.2, we have $\lceil -\log_{10}\varepsilon_1\rceil \geq \lceil \log G + \log_{10}(16n^{5/2}C) - \log_{10}\varepsilon\rceil$.

Example 4.4 (Algebraic numbers)**.** Let $\alpha = (\sqrt[5]{3} + \sqrt[4]{2})^{-1}$, and let $\boldsymbol{\alpha}$ be the normalized vector of $(\alpha^{20}, \alpha^{19}, \ldots, \alpha, 1)$. In this example, we try to recover the coefficients of the minimal polynomial of α. Suppose that we know in advance that the ∞-norm of the integer relation is at most $G = 7440$.

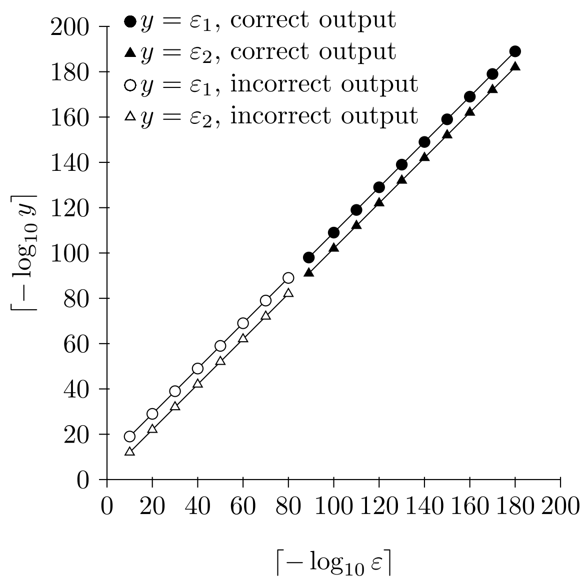

Figure 3. Error control strategy for Example 4.4

Bailey's estimation suggests that $\boldsymbol{\alpha}$ should be computed with at least $\lceil n\log_{10} G\rceil = 82$ exact decimal digits, which implies $\varepsilon_1 < 10^{-82}$. However, $\texttt{PSLQ}_\varepsilon$ does not return a relation with coefficient bounded by 7440. This may be caused by the fact that Bailey's estimation is not sufficient to compute an integer relation.

Let us set $\varepsilon = 10^{-89}$ so that $\varepsilon_1 \approx 1.73 \times 10^{-98}$ and $\varepsilon_2 \approx 4.99 \times 10^{-91}$, and our procedure returns a relation

$$\begin{aligned}\boldsymbol{m} = (&49, -1080, 3960, -3360, 80, -108, -6120, -7440,\\ &-80, 0, 54, -1560, 40, 0, 0, -12, -10, 0, 0, 0, 1)\end{aligned}$$

after 3525 iterations. It can be checked that this relation corresponds exactly to the coefficients of the minimal polynomial of α.

For the same ε and ε_1, if we do not set ε_2 as suggested by Theorem 4.2, say, $\varepsilon_2 \approx 10^{-96}$, then the procedure misses the correct relation.

If we set $\varepsilon = 10^{-88}$, our procedure does not return the correct answer. This can be seen as evidence that the sharp gap bound is near 10^{-89}. We also test for $\varepsilon = 10^{-(100-10i)}$ with $i = 1, 2, \ldots, 9$. Each of these tests does not return the correct answer. If we set ε more strictly, which means having more precision, for example $\varepsilon = 10^{-(100+10i)}$ with $i = 1, 2, \ldots, 8$, the procedure always works well and returns the same $\boldsymbol{m}$ as above. The quantities $\lceil -\log_{10} \varepsilon_1 \rceil$ and $\lceil -\log_{10} \varepsilon_2 \rceil$ obtained from Theorem 4.2 are as shown in Figure 3.

Example 4.5 (Algebraic numbers with higher degree)**.** Let $\alpha = (\sqrt[7]{3} + \sqrt[7]{2})^{-1}$, and let $\boldsymbol{\alpha}$ be the normalized vector of $(\alpha^{49}, \alpha^{48}, \ldots, \alpha, 1)$.

For this example, the dimension is 50 and the ∞-norm of the integer relation is $G = 966420105$. Bailey's estimation suggests that $\boldsymbol{\alpha}$ should be computed with at least $\lceil n \log_{10} G \rceil = 450$ exact decimal digits, which implies $\varepsilon_1 < 10^{-450}$. Under this setting, $\texttt{PSLQ}_\varepsilon$ fails to find the correct relation. The reason is that this precision is not enough to achieve the gap bound. In fact, according to our tests, the gap bound for this example is about 10^{-487}; see Figure 4. This shows that Bailey's estimation is not sufficient, but still necessary.

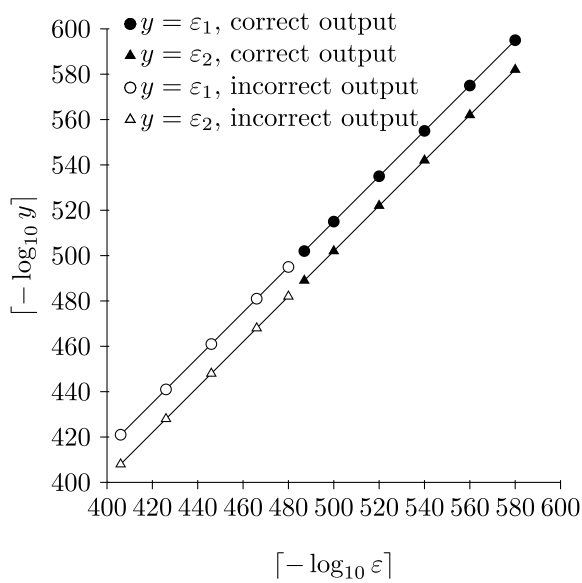

FIGURE 4. Error control strategy for Example 4.5

When we set $\varepsilon = 10^{-487}$ so that $\varepsilon_1 \approx 1.61 \times 10^{-502} (< 10^{-450})$ and $\varepsilon_2 \approx 3.47 \times 10^{-489}$ according to Theorem 4.2, then our procedure returns the correct relation corresponding to the coefficients of the minimal polynomial of α after 45385 iterations. Furthermore, the similar phenomenon shown in Figures 2 and 3 also appears for this example. When we set ε smaller than 10^{-487} (and set ε_1 and ε_2

accordingly), $\mathtt{PSLQ}_\varepsilon$ always returns the same integer relation, as shown in Figure 4. This shows that our error control strategy given in Theorem 4.2 plays an important role for the correctness of $\mathtt{PSLQ}_\varepsilon$.

From the examples above, we have the following two observations. Firstly, if one does not decide ε_1 and ε_2 by the error control strategy in Theorem 4.2, then one may miss the correct relation. Secondly, with an effective ε, we always obtain the same relation if we use the error control strategy in Theorem 4.2. This observation may be taken as strong evidence that the returned relation is a true integer relation. In fact, assume that for all arbitrarily small $\varepsilon > 0$, $\mathtt{PSLQ}_\varepsilon$ always returns the same relation. Then the relation must be an exact integer relation in the sense that $\mathtt{PSLQ}_\varepsilon \to \boldsymbol{m}$ for $\varepsilon \to 0$. However, if no gap bound is known, determining whether the returned relation is an exact integer relation within finite steps is still open.

5. Discussion and conclusion

In this paper, we give a new invariant relation of the celebrated integer relation finding algorithm PSLQ, and hence introduce a new termination condition for $\mathtt{PSLQ}_\varepsilon$. The new termination condition allows us to compute integer relations by $\mathtt{PSLQ}_\varepsilon$ with empirical data as its input. By a perturbation analysis, we disclose the relationship between the accuracy of the input data (ε_1) and the output quality (ε, an upper bound on the absolute value of the inner product of the intrinsic data and the computed relation) of the algorithm. This relationship still holds for the multipair variant of PSLQ. Examples show that our error control strategies based on this relationship are very helpful in practice.

We note that all results presented in this paper are under the exact arithmetic computational model. Although we obtain some results about the error control for applications, we did not analyze the algorithm under an inexact arithmetic model, such as floating-point arithmetic. However, we believe that the results in this paper, say Theorem 3.8, would be indispensable in the analysis of a numerical PSLQ algorithm.

In fact, it is an intriguing topic to design and analyze an efficient numerical PSLQ algorithm. For the moment, the main obstacle is to give a reasonable bound on the entries of unimodular matrices produced by the algorithm. Now, we can only give an upper bound that is double exponential with respect to the working dimension, and hence resulting in an exponential time algorithm. Thus, it is a very interesting challenge to obtain an upper bound similar to, e.g., [21, Lemma 6], where the upper bound is of a single exponential in the dimension.

Appendix A. Proof of Lemma 3.3

We consider the following submatrix of $\boldsymbol{H}_\alpha$, denoted by $\boldsymbol{H}_{[1..n-1]}$:

$$\boldsymbol{H}_{[1..n-1]} = \begin{pmatrix} \frac{s_2}{s_1} & 0 & 0 & \cdots & 0 & 0 \\ \frac{-\alpha_2\alpha_1}{s_1 s_2} & \frac{s_3}{s_2} & 0 & \cdots & 0 & 0 \\ \frac{-\alpha_3\alpha_1}{s_1 s_2} & \frac{-\alpha_3\alpha_2}{s_2 s_3} & \frac{s_4}{s_3} & \cdots & 0 & 0 \\ \frac{-\alpha_4\alpha_1}{s_1 s_2} & \frac{-\alpha_4\alpha_2}{s_2 s_3} & \frac{-\alpha_4\alpha_3}{s_3 s_4} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{-\alpha_{n-2}\alpha_1}{s_1 s_2} & \frac{-\alpha_{n-2}\alpha_2}{s_2 s_3} & \frac{-\alpha_{n-2}\alpha_3}{s_3 s_4} & \cdots & \frac{s_{n-1}}{s_{n-2}} & 0 \\ \frac{-\alpha_{n-1}\alpha_1}{s_1 s_2} & \frac{-\alpha_{n-1}\alpha_2}{s_2 s_3} & \frac{-\alpha_{n-1}\alpha_3}{s_3 s_4} & \cdots & \frac{-\alpha_{n-1}\alpha_{n-2}}{s_{n-2}s_{n-1}} & \frac{s_n}{s_{n-1}} \end{pmatrix}.$$

By linear algebra, its inverse is

$$\boldsymbol{H}^{-1}_{[1..n-1]} = \begin{pmatrix} \frac{s_1}{s_2} & 0 & 0 & \cdots & 0 & 0 \\ \frac{\alpha_1\alpha_2}{s_2 s_3} & \frac{s_2}{s_3} & 0 & \cdots & 0 & 0 \\ \frac{\alpha_1\alpha_3}{s_3 s_4} & \frac{\alpha_2\alpha_3}{s_3 s_4} & \frac{s_3}{s_4} & \cdots & 0 & 0 \\ \frac{\alpha_1\alpha_4}{s_4 s_5} & \frac{\alpha_2\alpha_4}{s_4 s_5} & \frac{\alpha_3\alpha_4}{s_4 s_5} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\alpha_1\alpha_{n-2}}{s_{n-2}s_{n-1}} & \frac{\alpha_2\alpha_{n-2}}{s_{n-2}s_{n-1}} & \frac{\alpha_3\alpha_{n-2}}{s_{n-2}s_{n-1}} & \cdots & \frac{s_{n-2}}{s_{n-1}} & 0 \\ \frac{\alpha_1\alpha_{n-1}}{s_{n-1}s_n} & \frac{\alpha_2\alpha_{n-1}}{s_{n-1}s_n} & \frac{\alpha_3\alpha_{n-1}}{s_{n-1}s_n} & \cdots & \frac{\alpha_{n-2}\alpha_{n-1}}{s_{n-1}s_n} & \frac{s_{n-1}}{s_n} \end{pmatrix}. \tag{A.1}$$

In the following, we compute the F-norm of $\boldsymbol{H}^{-1}_{[1..n-1]}$. First, consider the jth column of $\boldsymbol{H}^{-1}_{[1..n-1]}$:

$$\begin{aligned} \|H_j^{-1}\|^2 &= \frac{s_j^2}{s_{j+1}^2} + \sum_{k=j+1}^{n-1} \frac{\alpha_j^2\alpha_k^2}{s_k^2 s_{k+1}^2} = \frac{s_j^2}{s_{j+1}^2} + \alpha_j^2 \sum_{k=j+1}^{n-1} \frac{\alpha_k^2}{s_k^2 s_{k+1}^2} \\ &= \frac{s_j^2}{s_{j+1}^2} + \alpha_j^2 \sum_{k=j+1}^{n-1} (\frac{1}{s_{k+1}^2} - \frac{1}{s_k^2}) = \frac{s_j^2}{s_{j+1}^2} + \alpha_j^2(\frac{1}{s_n^2} - \frac{1}{s_{j+1}^2}) \\ &= \frac{s_j^2 - \alpha_j^2}{s_{j+1}^2} + \frac{\alpha_j^2}{s_n^2} = \frac{s_{j+1}^2}{s_{j+1}^2} + \frac{\alpha_j^2}{\alpha_n^2} = 1 + \frac{\alpha_j^2}{\alpha_n^2}, \end{aligned}$$

so we have

$$\begin{aligned} \|\boldsymbol{H}^{-1}_{[1..n-1]}\|_F^2 = \sum_{j=1}^{n-1} \|H_j^{-1}\|^2 &= (n-1) + \frac{\sum_{j=1}^{n-1} \alpha_j^2}{\alpha_n^2} \\ &= (n-1) + \frac{\|\alpha\|^2 - \alpha_n^2}{\alpha_n^2} = (n-2) + \frac{\|\alpha\|^2}{\alpha_n^2}. \end{aligned}$$

In addition, we can compute the F-norm of $\boldsymbol{H}_{[1..n-1]}$ as follows:

$$\begin{aligned} \|\boldsymbol{H}_{[1..n-1]}\|_F^2 &= \|\boldsymbol{H}_\alpha\|_F^2 - \sum_{i=1}^{n-1} \frac{\alpha_n^2\alpha_i^2}{s_i^2 s_{i+1}^2} = (n-1) - \alpha_n^2 \sum_{i=1}^{n-1} \frac{\alpha_i^2}{s_i^2 s_{i+1}^2} \\ &= (n-1) - \alpha_n^2 \sum_{i=1}^{n-1} (\frac{1}{s_{i+1}^2} - \frac{1}{s_i^2}) = (n-1) - \alpha_n^2(\frac{1}{s_n^2} - \frac{1}{s_1^2}) \\ &= (n-1) - 1 + \frac{\alpha_n^2}{\|\alpha\|^2} = (n-2) + \frac{\alpha_n^2}{\|\alpha\|^2}. \end{aligned}$$

as claimed in Lemma 3.3.

Appendix B. Proof of Theorem 3.2

Define the Π function after exactly k iterations as follows:

$$\Pi(k)=\prod_{j=1}^{n-1}\max\left(|h_{i,i}(k)|,\frac{h_{\max}(k)}{\gamma^{n-1}}\right)^{n-j},$$

where $h_{\max}(k)$ is the maximum of $|h_{i,i}(k)|$ for $i=1,2,\ldots,n-1$. It obviously holds that

$$\Pi(k)=\prod_{j=1}^{n-1}\max\left(|h_{i,i}(k)|,\frac{h_{\max}(k)}{\gamma^{n-1}}\right)^{n-j}\geq\left(\frac{h_{\max}(k)}{\gamma^{n-1}}\right)^{\frac{n(n-1)}{2}}.$$

First, we assert that $h_{\max}(k)\geq h_{\max}(k+1)$. Size reduction does not affect $h_{i,i}(k)$, and neither does $h_{\max}$. Let us consider the change of $h_{\max}$ in the Bergman swap. Let the Bergman swap occur at the rth row. For the case of $r<n-1$, after the Bergman swap, we have that

$$\begin{aligned}&|h_{r,r}(k+1)|<\frac{1}{\tau}|h_{r,r}(k)|<|h_{r,r}(k)|=h_{\max}(k)\\&|h_{r+1,r+1}(k+1)|=\frac{|h_{r,r}(k)h_{r+1,r+1}(k)|}{\sqrt{h_{r+1,r}^2(k)+h_{r+1,r+1}^2(k)}}\leq|h_{r,r}(k)|=h_{\max}(k)\end{aligned}$$

and the others are unchanged, i.e., $h_{i,i}(k+1)=h_{i,i}(k)$ for $i=1,\ldots,r-1,r+1,\ldots,n-1$. It shows that $h_{\max}(k)\geq h_{\max}(k+1)$ for $r<n-1$. For the case of $r=n-1$, after the Bergman swap, it holds that $|h_{n-1,n-1}(k+1)|<\frac{1}{\rho}|h_{n-1,n-1}(k)|\leq h_{\max}(k)$ and the other $h_{i,i}$'s are unchanged. Therefore it is obtained that $h_{\max}(k)\geq h_{\max}(k+1)$ for $r=n-1$.

Second, we show that $\Pi(k)>\tau\Pi(k+1)$. Let the Bergman swap occur at row r. Case $r=n-1$: We have

$$\begin{aligned}\frac{\Pi(k)}{\Pi(k+1)}&=\frac{\max\{|h_{n-1,n-1}(k)|,\frac{h_{\max}(k)}{\gamma^{n-1}}\}}{\max\{|h_{n-1,n-1}(k+1)|,\frac{h_{\max}(k+1)}{\gamma^{n-1}}\}}=\frac{|h_{n-1,n-1}(k)|}{\max\{|h_{n-1,n}(k)|,\frac{h_{\max}(k+1)}{\gamma^{n-1}}\}}\\&=\begin{cases}\frac{|h_{n-1,n-1}(k)|}{|h_{n,n-1}(k)|}\geq\rho\geq\tau, & \text{when } h_{n,n-1}(k)>\frac{h_{\max}(k+1)}{\gamma^{n-1}},\\ \frac{|h_{n-1,n-1}(k)|}{\frac{h_{\max}(k+1)}{\gamma^{n-1}}}\geq\frac{|h_{n-1,n-1}(k)|}{\frac{h_{\max}(k)}{\gamma^{n-1}}}\geq\gamma\geq\tau, & \text{otherwise,}\end{cases}\end{aligned}$$

where we used $\gamma^{n-1}h_{n-1,n-1}(k)\geq h_{\max}(k)$ and $h_{n-1,n-1}(k+1)=h_{n-1,n}(k)$.

Cases $r<n-1$: Let

$$A=\frac{\max\{|h_{r,r}(k)|,\frac{h_{\max}(k)}{\gamma^{n-1}}\}}{\max\{|h_{r,r}(k+1)|,\frac{h_{\max}(k+1)}{\gamma^{n-1}}\}},\ B=\frac{\max\{|h_{r+1,r+1}(k)|,\frac{h_{\max}(k)}{\gamma^{n-1}}\}}{\max\{|h_{r+1,r+1}(k+1)|,\frac{h_{\max}(k+1)}{\gamma^{n-1}}\}}.$$

Then $\frac{\Pi(k)}{\Pi(k+1)} = A(AB)^{n-r-1}$. Set $\eta = h_{r,r}(k)$, $\lambda = h_{r+1,r+1}(k)$, $\beta = h_{r+1,r}(k)$ and $\delta = \sqrt{\beta^2+\lambda^2}$. Noticing that $h_{\max}(k) \geq h_{\max}(k+1)$ and $|\eta| > \frac{h_{\max}(k)}{\gamma^{n-1}}$ yields

$$
\begin{aligned}
A &= \frac{\max\{|h_{r,r}(k)|, \frac{h_{\max}(k)}{\gamma^{n-1}}\}}{\max\{|h_{r,r}(k+1)|, \frac{h_{\max}(k+1)}{\gamma^{n-1}}\}} = \frac{|\eta|}{\max\{\delta, \frac{h_{\max}(k+1)}{\gamma^{n-1}}\}} \\
&= \begin{cases} \frac{|\eta|}{\delta} = \frac{1}{\sqrt{\frac{\beta^2}{\eta^2}+\frac{\lambda^2}{\eta^2}}} \geq \tau, & \text{when } \delta \geq \frac{h_{\max}(k+1)}{\gamma^{n-1}}, \\ \frac{|\eta|}{\frac{h_{\max}(k+1)}{\gamma^{n-1}}} = \frac{|\eta|\gamma^{n-1}}{h_{\max}(k+1)} \geq \frac{|\eta|\gamma^{n-1}}{h_{\max}(k)} \geq \gamma \geq \tau, & \text{otherwise.} \end{cases}
\end{aligned}
\tag{B.1}
$$

And then, we consider $AB = A \cdot \frac{\max\{|\lambda|, \frac{h_{\max}(k)}{\gamma^{n-1}}\}}{\max\{\frac{|\eta\lambda|}{\delta}, \frac{h_{\max}(k+1)}{\gamma^{n-1}}\}}$. When $|\lambda| \geq \frac{h_{\max}(k)}{\gamma^{n-1}}$, it is easily deduced that $\delta \geq |\lambda| \geq \frac{h_{\max}(k)}{\gamma^{n-1}} \geq \frac{h_{\max}(k+1)}{\gamma^{n-1}}$ and $\frac{|\eta\lambda|}{\delta} > \lambda \geq \frac{h_{\max}(k+1)}{\gamma^{n-1}}$. Hence from equation (B.1) it holds that

$$AB = A \cdot \frac{|\lambda|}{\frac{|\eta\lambda|}{\delta}} = A \cdot \frac{\delta}{|\eta|} = \frac{|\eta|}{\delta} \cdot \frac{\delta}{|\eta|} = 1.$$

When $|\lambda| < \frac{h_{\max}(k)}{\gamma^{n-1}}$, it holds that

$$
\begin{aligned}
AB =& A \cdot \frac{\frac{h_{\max}(k)}{\gamma^{n-1}}}{\max\{\frac{|\eta\lambda|}{\delta}, \frac{h_{\max}(k+1)}{\gamma^{n-1}}\}} \\
=& \begin{cases} A \cdot \frac{\frac{h_{\max}(k)}{\gamma^{n-1}}}{\frac{h_{\max}(k+1)}{\gamma^{n-1}}} \geq A \geq \tau > 1, \text{ if } \frac{|\eta\lambda|}{\delta} \leq \frac{h_{\max}(k+1)}{\gamma^{n-1}}, \\ A \cdot \frac{\frac{h_{\max}(k)}{\gamma^{n-1}}}{\frac{|\eta\lambda|}{\delta}} = \begin{cases} \frac{|\eta|}{\delta} \cdot \frac{h_{\max}(k)}{\gamma^{n-1}} \cdot \frac{\delta}{|\eta\lambda|} = \frac{h_{\max}(k)}{\lambda\gamma^{n-1}} > 1, & \text{else if } \delta > \frac{h_{\max}(k+1)}{\gamma^{n-1}}, \\ \frac{|\eta|}{\frac{h_{\max}(k+1)}{\gamma^{n-1}}} \cdot \frac{\frac{h_{\max}(k)}{\gamma^{n-1}}}{\frac{|\eta\lambda|}{\delta}} \geq \frac{\delta}{|\lambda|} \geq 1, & \text{otherwise.} \end{cases} \end{cases}
\end{aligned}
$$

Up to now, we have shown that $AB \geq 1$. Therefore

$$\frac{\Pi(k)}{\Pi(k+1)} = A[AB]^{n-r-1} > A > \tau.$$

It is proved that

$$\left(\frac{h_{\max}(k)}{\gamma^{n-1}}\right)^{\frac{n(n-1)}{2}} \leq \Pi(k) \leq \frac{1}{\tau^k}. \tag{B.2}$$

From $\tau > 1$, we have

$$k \leq \frac{n(n-1)((n-1)\log\gamma + \log\frac{1}{h_{\max}(k)})}{2\log\tau}.$$

From $|h_{n,n-1}(k)| < |h_{n-1,n-1}(k)| < h_{\max}(k)$, it always holds that $h_{\max}(k) \geq \varepsilon_2$ before termination. Hence, we deduce that

$$k \leq \frac{n(n-1)[(n-1)\log\gamma + \log\frac{1}{\varepsilon_2}]}{2\log\tau},$$

which completes the proof.

Acknowledgment

We would like to thank an anonymous referee for helpful suggestions that greatly improved the presentation of this paper.

References

[1] L. Babai, B. Just, and F. Meyer auf der Heide, *On the limits of computations with the floor function*, Inform. and Comput. **78** (1988), no. 2, 99–107. MR955578

[2] D. H. Bailey, *ARPREC: A C++/Fortran-90 arbitrary precision package*, Available at `http://www.davidhbailey.com/dhbsoftware`, accessed in November, 2017.

[3] D. H. Bailey, *MPFUN2015: A high-precision software directory*, Available at `http://www.davidhbailey.com/dhbsoftware`, accessed in November, 2017.

[4] D. H. Bailey, *MPFUN: A portable high performance multiprecision package*, Tech. Report RNR-90-022, NASA Ames Research Center, December 1990.

[5] D. H. Bailey, *Integer relation detection*, Computing in Science & Engineering **2** (2000), no. 1, 24–28.

[6] D. H. Bailey, *A collection of mathematical formulas involving* π, (2016), Available at `http://www.davidhbailey.com/dhbpapers/pi-formulas.pdf`.

[7] D. H. Bailey and Jonathan M. Borwein, *High-precision arithmetic in mathematical physics*, Mathematics **3** (2015), no. 2, 337–367.

[8] D. H. Bailey, J. M. Borwein, and J. S. Kimberley, *Computer discovery and analysis of large Poisson polynomials*, with an appendix by Watson Ladd, Exp. Math. **26** (2017), no. 3, 349–363. MR3642112

[9] D. H. Bailey and D. J. Broadhurst, *Parallel integer relation detection: techniques and applications*, Math. Comp. **70** (2001), no. 236, 1719–1736. MR1836930

[10] J. M. Borwein and P. Lisoněk, *Applications of integer relation algorithms*, Formal power series and algebraic combinatorics (Vienna, 1997) (English, with English and French summaries), Discrete Math. **217** (2000), no. 1-3, 65–82. MR1766260

[11] J. Chen, D. Stehlé, and G. Villard, *A new view on HJLS and PSLQ: sums and projections of lattices*, ISSAC 2013—Proceedings of the 38th International Symposium on Symbolic and Algebraic Computation, ACM, New York, 2013, pp. 149–156. MR3206352

[12] Y. Feng, J. Chen, and W. Wu, *Two variants of HJLS-PSLQ with applications*, SNC 2014—Proceedings of the 2014 Symposium on Symbolic-Numeric Computation, ACM, New York, 2014, pp. 88–96. MR3392719

[13] H, R. P. Ferguson and D. H. Bailey, *A polynomial time, numerically stable integer relation algorithm*, Tech. Report RNR-91-032, NASA Ames Research Center, 1992. `http://davidhbailey.com/dhbpapers/pslq.pdf`

[14] H. R. P. Ferguson, D. H. Bailey, and S. Arno, *Analysis of PSLQ, an integer relation finding algorithm*, Math. Comp. **68** (1999), no. 225, 351–369. MR1489971

[15] G. H. Golub and C. F. Van Loan, *Matrix computations*, 4th ed., Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, 2013. MR3024913

[16] J. Håstad, B. Just, J. C. Lagarias, and C.-P. Schnorr, *Polynomial time algorithms for finding integer relations among real numbers*, SIAM J. Comput. **18** (1989), no. 5, 859–881. MR1015261

[17] Ch. Hermite, *Sur l'intégration des fractions rationnelles* (French), Ann. Sci. École Norm. Sup. (2) **1** (1872), 215–218. MR1508584

[18] B. Just, *Integer relations among algebraic numbers*, Mathematical foundations of computer science 1989 (Porąbka-Kozubnik, 1989), Lecture Notes in Comput. Sci., vol. 379, Springer, Berlin, 1989, pp. 314–320. MR1036810

[19] R. Kannan, A. K. Lenstra, and L. Lovász, *Polynomial factorization and nonrandomness of bits of algebraic and some transcendental numbers*, Math. Comp. **50** (1988), no. 181, 235–250. MR917831

[20] A. K. Lenstra, H. W. Lenstra Jr., and L. Lovász, *Factoring polynomials with rational coefficients*, Math. Ann. **261** (1982), no. 4, 515–534. MR682664

[21] Saruchi, I. Morel, D. Stehlé, and G. Villard, *LLL reducing with the most significant bits*, ISSAC 2014—Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation, ACM, New York, 2014, pp. 367–374. MR3239948

[22] A. Stenger, *Experimental math for Math Monthly problems*, Amer. Math. Monthly **124** (2017), no. 2, 116–131. MR3608241

Chongqing Key Lab of Automated Reasoning and Cognition, Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, People's Republic of China
Email address: yongfeng@cigit.ac.cn

Chongqing Key Lab of Automated Reasoning and Cognition, Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, People's Republic of China
Email address: chenjingwei@cigit.ac.cn

Chongqing Key Lab of Automated Reasoning and Cognition, Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, People's Republic of China
Email address: wuwenyuan@cigit.ac.cn