Analysis of PSLQ, an Integer Relation Finding Algorithm

Helaman R.P. Ferguson[1], David H. Bailey[2], and Steve Arno[1]

NAS Technical Report NAS-96-005 April 96

helamanf@super.org, dbailey@nas.nasa.gov, and arno@super.org
NASA Ames Research Center
Mail Stop T27A-1
Moffett Field, CA 94035-1000

Abstract

Let K be either the real, complex, or quaternion number system and let $O(K)$ be the corresponding integers. Let $x = (x_1, \ldots, x_n)$ be a vector in K^n. The vector x has an integer relation if there exists a vector $m = (m_1, \ldots, m_n) \in O(K)^n$, $m \neq 0$, such that $m_1 x_1 + m_2 x_2 + \ldots + m_n x_n = 0$. In this paper we define the parameterized integer relation construction algorithm $PSLQ(\tau)$, where the parameter τ can be freely chosen in a certain interval.

Beginning with an arbitrary vector $x = (x_1, \ldots, x_n) \in K^n$, iterations of $PSLQ(\tau)$ will produce lower bounds on the norm of any possible relation for x. Thus $PSLQ(\tau)$ can be used to prove that there are no relations for x of norm less than a given size. Let M_x be the smallest norm of any relation for x. For the real and complex case and each fixed parameter τ in a certain interval, we prove that $PSLQ(\tau)$ constructs a relation in less than $O(n^3 + n^2 \log M_x)$ iterations.

[1] Center for Computing Sciences, 17100 Science Drive, Bowie, MD 20715-4300

[2] NASA Ames Research Center, Moffett Field, CA 94035-1000

1. Introduction

Let $\mathbb{K}$ be either the real, complex or quaternion number system and let $\mathbb{O}(\mathbb{K})$ be the corresponding system of integers (i.e., ordinary integers, Gaussian integers, or Hamiltonian integers, respectively). Let $x = (x_1, \ldots, x_n)$ be a vector in $\mathbb{K}^n$. The vector x has an *integer relation* if there exists a vector $m = (m_1, \ldots, m_n) \in \mathbb{O}(\mathbb{K})^n$, $m \neq 0$, such that $m_1x_1 + m_2x_2 + \ldots + m_nx_n = 0$. In this paper we define the parameterized integer relation construction algorithm PSLQ(τ). The parameter τ can be freely chosen in the interval $1 < \tau < \rho$, where ρ is 2, $\sqrt{2}$, or 1, depending on whether $\mathbb{K}$ is the real, complex or quaternion number system, respectively. We analyze PSLQ(τ) for these three number systems. We describe in detail some efficient Fortran multiprecision computer implementations of PSLQ(τ). We also present working Mathematica^{TM} code for PSLQ(τ) and, for comparison, some other relation finding algorithms from the literature.

Beginning with an arbitrary vector $x = (x_1, \ldots, x_n) \in \mathbb{K}^n$, a finite number of iterations of PSLQ(τ) will produce lower bounds on the norm of any possible relation for x. The computation of such a lower bound constitutes a proof that x has no integer relations whatsoever of norm less than this lower bound. Any finite computation can only prove that no small relation exists.

Let M_x be the smallest norm of a relation for x. Let $\tau = 1/\sqrt{1/\rho^2 + 1/\gamma^2}$, where $\rho = 2$ for the real number field and $\rho = \sqrt{2}$ for the complex number field. For each fixed parameter τ in the interval $1 < \tau < \rho$, we prove in the real and complex case that PSLQ(τ) constructs a relation in less than $\binom{n}{2} \log_\tau (\gamma^{n-1} M_x)$ iterations. This shows that PSLQ(τ) is "polynomial time" in the dimension and the number of bits of a smallest integer relation. Different τ or γ choices lead to different time and space requirements for the algorithm.

For dimension $n = 2$ we prove that PSLQ(τ) will construct a relation of smallest norm M_x. We give examples in dimension $n = 3$, for some τ, for which PSLQ(τ) does not construct a relation of smallest norm M_x. However for any dimension $n \geq 2$, we do prove that any relation constructed by PSLQ(τ) has norm less than or equal to $\gamma^{n-2} M_x$.

The "polynomial time" and "small norm" proofs given here are straightforward generalizations to the parameter τ and to the complex numbers of the original "polynomial time" proofs that appear in Lagarias et al, [HJLS89]. We show, however, that the algorithm of [HJLS89] is distinct from any of these PSLQ(τ) algorithms.

PSLQ(τ) was introduced by the authors [BaFe91] in 1991. PS refers to partial sums of squares, LQ to a lower trapezoidal orthogonal decomposition, and (τ) is a parameter defined as above. Since PSLQ(τ) was introduced it has been used to discover numerous previously unknown identities among real numbers. One example is

$$\sum_{k=1}^{\infty}\left(1-\frac{1}{2}+\cdots+\frac{(-1)^{k+1}}{k}\right)^2 (k+1)^{-3}$$

$$= 4L_5(1/2) - \frac{1}{30}\ln^5(2) - \frac{17}{32}\zeta(5) - \frac{11}{720}\pi^4\ln(2)$$
$$+\frac{7}{4}\zeta(3)\ln^2(2) + \frac{1}{18}\pi^2\ln^3(2) - \frac{3}{24}\pi^2\zeta(3),$$

where $L_n(x)$ denotes the polylogarithm function $\sum_k x^k k^{-n}$. See [BaBG94] for details. Another example is the following formula for π:

$$\pi = \sum_{i=0}^{\infty}\frac{1}{16^i}\left(\frac{4}{8i+1}-\frac{2}{8i+4}-\frac{1}{8i+5}-\frac{1}{8i+6}\right).$$

This remarkable series permits one to rapidly compute individual digits from the hexadecimal expansion of π. See [BaBP95] for details. It was found by applying PSLQ(τ) to the vector $X = (X_1, X_2, \cdots, X_8, \pi)$ where $X_j = \sum_{k\geq 0} 1/(16^k(8k+j))$. The smallest relation known,

$$(4,0,0,-2,-1,-1,0,0,-1),$$

yields the above "base 16" formula for π. A next smallest relation known,

$$(0,8,4,4,0,0,-1,0,-2),$$

was subsequently discovered by Ferguson and this relation yields a similar "base 16" formula for π. Together these two integral lattice relation vectors generate a two-dimensional lattice of relations of this "base 16" type. It is conjectured there are no further such relations outside this lattice. Note that

$$(-8,8,4,8,2,2,-1,0,0)$$

is in this lattice, so evidently X_7 is integrally dependent upon $X_1, \ldots, X_6$.

Of course, a numerical discovery of a relation using PSLQ(τ) does not constitute a rigorous proof of the relation. However, in the wake of this numerical evidence, proofs have subsequently been found for many of these relations, including the above formula for π. See [BorBG95] and [BaBP95] for details.

In the theoretical proofs in Section 2, 3, 4, and 5, we will assume exact arithmetic over the real numbers augmented by comparisons over the reals and the nearest integer function.

2. Lower Bounds on Integer Relations

If $\mathbb{K}$ is the complex number field, then $z^\star$ denotes the complex conjugate of z, i.e. if $z = x + iy$, then $z^\star = x - iy$. $|\cdot|$ denotes the complex absolute value, i.e. $|z|^2 = z^\star z = zz^\star = x^2 + y^2$. If A is a matrix or vector, then $A^\star$ is the conjugate transpose of A. A unit in the complex number field is any element z such that $|z| = 1$. For real z, the conjugate operation is null, and z is the usual absolute value.

Similarly, if $\mathbb{K}$ is the quaternion number system, then $z^\star$ denotes the quaternion conjugate of z, i.e. if $z = x+yi+uj+vk$, then $z^\star = x-yi-uj-vk$. The quaternion absolute value or norm is similarly defined, so that $|z|^2 = zz^\star = z^\star z = x^2 + y^2 + u^2 + v^2$. Units and conjugates of matrices are defined analogously.

If $\mathbb{K}$ is any of the above three number fields, two vectors $x, y \in \mathbb{K}^n$ are said to be orthogonal if $xy^\star = 0$. Let $|A| = (\mathrm{tr}(A^\star A))^{1/2}$ denote the Frobenius norm of the matrix A, i.e., $|A| = \left(\sum a_{i,j}^\star a_{i,j}\right)^{1/2}$. An $n \times n$ matrix A is unitary if $A^\star A = AA^\star = I_n$. $U(n, \mathbb{K})$ denotes the group of unitary matrices over $\mathbb{K}$. An $n \times n$ matrix A is unimodular if $\det A$ is a unit. $GL(n, \mathbb{O}(\mathbb{K}))$ is the group of unimodular matrices with entries in the integers $\mathbb{O}(\mathbb{K})$.

Definition 1: (M_x). Assume $x = (x_1, \ldots, x_n) \in \mathbb{K}^n$ has norm $|x| = 1$. Define $x^\perp$ to be all vectors in $\mathbb{K}^n$ orthogonal to x. Let $\mathbb{O}(\mathbb{K})^n \cap x^\perp$ be the discrete lattice of integral relations for x. Define $M_x > 0$ to be the smallest norm of any relation for x in this lattice.

Definition 2: (H_x). Assume $x = (x_1, \ldots, x_n) \in \mathbb{K}^n$ has norm $|x| = 1$. For $1 \leq j \leq n$ define the partial sums

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

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

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

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

Lemma 1. *Let H_x be the lower trapezoidal matrix defined above. Then (i) $H_x^\star H_x = I_{n-1}$, i.e., the columns of H_x are orthogonal,*

(ii) $|H_x| = \sqrt{n-1}$,
(iii) $xH_x = 0$.

Proof. The columns can be proven orthogonal by considering the cases $i = j$ and $i < j$ separately. When $i = j$ the inner product is

$$\frac{s_{i+1}^2}{s_i^2} + \sum_{i<k\leq n} \frac{x_i x_i^\star x_k x_k^\star}{s_i^2 s_{i+1}^2} = \frac{s_{i+1}^2}{s_i^2} + \frac{x_i x_i^\star}{s_i^2 s_{i+1}^2} \sum_{i<k\leq n} x_k x_k^\star$$

$$= \frac{s_i^2 - x_i x_i^\star}{s_i^2} + \frac{x_i x_i^\star}{s_i^2} = 1.$$

When $i < j$ the inner product is

$$-\frac{s_{j+1} x_i^\star x_j}{s_j s_i s_{i+1}} + \sum_{j<k\leq n} \frac{x_i^\star x_j x_k x_k^\star}{s_i s_{i+1} s_j s_{j+1}}$$

$$= -\frac{s_{j+1} x_i^\star x_j}{s_j s_i s_{i+1}} + \frac{x_i^\star x_j}{s_i s_{i+1} s_j s_{j+1}} \sum_{j<k\leq n} x_k x_k^\star = 0.$$

Item (i) shows that $H_x^\star H_x = I_{n-1}$ which has trace $n-1$ so $|H_x| = \sqrt{n-1}$. To prove (iii), fix $1 \leq j \leq n-1$, then

$$\sum_{1\leq k\leq n} x_k h_{k,j} = \frac{x_j s_{j+1}}{s_j} - \sum_{j<k\leq n} \frac{x_k x_k^\star x_j}{s_j s_{j+1}} =$$

$$\frac{x_j s_{j+1}}{s_j} - \frac{x_j s_{j+1}^2}{s_j s_{j+1}} = 0. \qquad \square$$

Lemma 2. *For a unit vector $x \in \mathbb{K}^n$ define $P_x = H_x H_x^\star$. Then P_x satisfies:*
(i) $P_x^\star = P_x$,
(ii) $P_x = I_n - x^\star x$,
(iii) $P_x^2 = P_x$,
(iv) $|P_x| = \sqrt{n-1}$,
(v) $P_x z^\star = z^\star$ *for any* $z \in x^\perp$
(vi) $P_x m^\star = m^\star$ *for any relation* $m \in \mathbb{O}(\mathbb{K})^n$ *for* x.

Proof. Item (i) follows from $H_x H_x^\star = (H_x H_x^\star)^\star$. To prove (ii) note that from Lemma 1 (iii), H_x is an $n \times (n-1)$ rank $n-1$ matrix whose columns transposed form an orthonormal basis for $x^\perp$. Defining $U = (H_x|x^\star)$, an $n \times n$ unitary matrix, we have $UU^\star = H_x H_x^\star + x^\star x = I_n$. To prove (iii) note that

$$P_x^2 = (I_n - x^\star x)^2 = I_n^2 - 2I_n x^\star x + x^\star(xx^\star)x = P_x.$$

To prove (iv) note that $|P_x|^2 = \mathrm{tr}(P_x^\star P_x) = \mathrm{tr}P_x = \mathrm{tr}H_x^\star H_x = n-1$. Item (vi) follows from (v) which follows from (ii) and the associativity $(x^\star x)z^\star = x^\star(xz^\star)$. □

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

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

Proof. Let m be any relation for x. By the hypothesis, there exists a unitary matrix $Q \in U(n-1)$ such that $H = AH_xQ$ is lower trapezoidal (this is equivalent to QR factorization). There is an $n \times n-1$ matrix T with diagonal ones and an $n-1 \times n-1$ diagonal matrix D where $H = TD$ with diagonal entries $h_{j,j} \neq 0, 1 \le j \le n-1$ from the hypothesis. On the other hand, $AP_x = HQ^\star H_x^\star$, from the definition of P_x in Lemma 2. The equation $AP_x = TDQ^\star H_x^\star$ gives a decomposition of AP_x into the product of a lower trapezoidal matrix T with diagonal 1's, an invertible diagonal matrix D with diagonal h's, and an $n-1 \times n$ matrix $Q^\star H_x^\star$ with orthonormal rows since $Q^\star H_x^\star H_x Q = Q^\star I_{n-1} Q = I_{n-1}$ by Lemma 1. So the norm of the j-th row of $DQ^\star H_x^\star$ is $|h_{j,j}|$.

From Lemma 2, part (vi), $m^\star = P_x m^\star$, so that $Am^\star = AP_x m^\star$. From the above decomposition of $AP_x = TDQ^\star H_x^\star$, we have $Am^\star = AP_x m^\star = TD(Q^\star H_x^\star)m^\star$. Let $Q_{H,j}$ be the j-th row of $Q^\star H_x^\star$ and let A_j be the j-th row of A. Then

$$A_j m^\star = h_{j,j}Q_{H,j}m^\star + \sum_{k<j} t_{j,k}h_{k,k}Q_{H,k}m^\star.$$

Since A is invertible, $Am^\star \neq 0$. Let j be the least j for which $A_j m^\star \neq 0$ so that $A_k m^\star = 0$ for $k < j$. Then the $k < j$ rows of $TDQ^\star H_x^\star m^\star$ are zero, and since T is lower trapezoidal by recursion, the k-th rows of $Q^\star H_x^\star m^\star$ are also zero. With this least choice of j then $A_j m^\star = h_{j,j}Q_j m^\star$. Therefore, from $A \in GL(n, \mathbb{O}(\mathbb{K}))$,

$$1 \le |A_j m^\star| \le |h_{j,j}Q_{H,j}m^\star| \le |h_{j,j}||m^\star|,$$

because $Q_{H,j}$ is a unit vector. □

Comment on Theorem 1. Theorem 1 suggests a strategy to construct a relation finding algorithm: Find a way to reduce the norm of the matrix H_x by multiplication by some unimodular A on the left. The inequality of Theorem 1 offers an increasing lower bound on the size of any possible relation. Theorem 1 can be used with any algorithm that produces any $GL(n, \mathbb{O}(\mathbb{K}))$ matrices. Any $GL(n, \mathbb{O}(\mathbb{K}))$ matrix A whatsoever can be put into Theorem 1.

Definition 3: (Hermite reduction). Let H be a lower trapezoidal matrix, with $h_{i,j} = 0$ if $j > i$ and $h_{j,j} \neq 0$. Define the matrix $D = (d_{i,j}) \in GL(n, \mathbb{O}(K))$ recursively as follows. For fixed i, decrement j from n to 1, setting

$$d_{i,j} = \begin{cases} 0 & \text{if } i < j \\ 1 & \text{if } i = j \\ \operatorname{nint}((-\sum_{j<k\leq i} d_{i,k}h_{k,j})/h_{j,j}) & \text{if } j < i, \end{cases}$$

We will say that DH is the *Hermite reduction* of H and we will say that D is the *reducing matrix* of H. The function nint denotes a nearest integer function, e.g., $\operatorname{nint}(t) = \lfloor t + 1/2 \rfloor$. This definition of nint can be extended to each coordinate for complex or quaternion arguments.

Definition 4: (Modified Hermite reduction). With the same notation as in Definition 3, set $D = I_n$. For i from 2 to n, and for j from $i - 1$ to 1 (step -1), set $q = \operatorname{nint}(h_{i,j}/h_{j,j})$; then for k from 1 to j replace $h_{i,k}$ by $h_{i,k} - qh_{j,k}$, and for k from 1 to n replace $d_{i,k}$ by $d_{i,k} - qd_{j,k}$.

Lemma 3. *For a lower triangular matrix H with $h_{i,j} = 0$ if $j > i$ and $h_{j,j} \neq 0$, Hermite reduction is equivalent to modified Hermite reduction.*

Comment. This variation can be found in [Berg80] and later in [LLjL82]. This recursion replaces the input H with DH while developing the left multiplying reduction matrix D.

Lemma 4. *There exists a constant $\rho_{\mathbb{K}} = \rho \geq 1$, with the property that the entries of the Hermite reduced matrix $H' = (h'_{i,j}) = DH$ satisfy the inequality*

$$|h'_{k,i}| \leq |h'_{i,i}|/\rho = |h_{i,i}|/\rho$$

for all $k > i$. The constant $\rho = 2$ for the real case, $\rho = \sqrt{2}$ for the complex case, and $\rho = 1$ for the quaternion case.

Proof. This follows from the definitions of the nint function, Hermite reduction, and the fact that $|z - \operatorname{nint}(z)| \leq \sqrt{\dim_{\mathbb{R}} \mathbb{K}}/2$ for $z \in \mathbb{K}$. □

3. Statement of the Algorithm PSLQ(τ)

Definition 5: (The parameters γ and τ). Fix the real number $\gamma > 2/\sqrt{3}$ or $\gamma > \sqrt{2}$ or $\gamma = \infty$ for the real, complex, and quaternion cases respectively. In terms of this γ, define the real number τ by

$$1/\tau^2 = 1/\rho^2 + 1/\gamma^2,$$

where ρ is defined as in Lemma 4. For the proof of Theorem 2, we will require that $1 < \tau$ and that $\tau \leq \rho$; clearly these conditions are satisfied in the real and complex cases. In the quaternion case $\tau = 1$ and $\rho = 1$.

For the proofs that follow assume $\mathbb{K}$ is real or complex, not quaternion. Note however that the statement of the algorithm is valid for the quaternions.

Initial conditions: Given the input unit vector $x \in \mathbb{K}^n$, set $H = H_x$ where H_x is defined as above. Set the $n \times n$ matrices A and B to the identity I_n. Perform Hermite reduction on H, producing $D \in GL(n, \mathbb{O}(\mathbb{K}))$. Replace x by xD^{-1}, H by DH, A by DA, B by BD^{-1}.

One four-step iteration:

Step 1: Exchange

Let $H = (h_{i,j})$ where $h_{i,j}$ is the i-th row, j-th column entry of H. Let

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

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. Replace x by xR, H by RH, A by RA, and B by BR.

Step 2: Corner

At this point the updated matrix H may not be lower trapezoidal since λ may not be zero. If $r < n-1$ replace H by HQ where Q is the unitary $n-1 \times n-1$ matrix $Q = (q_{i,j}) \in U(n-1, \mathbb{K})$ defined by

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

where the $\alpha, \beta, \lambda, \delta$ are defined in Step 1. If $r = n - 1$ then H is unchanged.

Step 3: Reduction

Perform Hermite reduction on H, producing $D \in GL(n, \mathbb{O}(\mathbb{K}))$. Replace x by xD^{-1}, H by DH, A by DA, B by BD^{-1}.

Step 4: Termination

Terminate the algorithm if $x_j = 0$ for some $1 \leq j \leq n$ or if $h_{i,i} = 0$ for some $1 \leq i \leq n - 1$.

4. Number of Iterations of PSLQ(τ)

Let $H(k) = H$, A, and $B = A^{-1}$ be the result after exactly k iterations of PSLQ. Let $\alpha = h_{r,r}(k)$ and $\beta = h_{r+1,r}(k)$. These definitions of α and β are consonant with those of Step 2. Because H is Hermite reduced in Step 3, from Lemma 4, $|\beta| < |\alpha|/\rho$. For $r < n - 1$ set $\lambda = h_{r+1,r+1}(k)$ and define t by $t = \sqrt{\beta\beta^\star + \lambda\lambda^\star}/|\alpha|$. From this definition of t we have

$$|\lambda| \leq |\alpha| t.$$

From the Step 1 Exchange, $0 \leq |\lambda| \leq |\alpha|/\gamma$. It follows that

$$t = \sqrt{\beta\beta^\star + \lambda\lambda^\star}/|\alpha| \leq \sqrt{1/\rho^2 + 1/\gamma^2} = \tau,$$

as in Definition 5. For this proof we will require that $t < 1 < \tau$, clearly satisfied in the real and complex cases.

Lemma 5. *If $h_{j,j}(k) = 0$ for some $1 \leq j \leq n-1$ and no smaller k, then $j = n - 1$ and a relation for x must appear as a column of the matrix B.*

Proof. (Alyson Reeves) First we show that $h_{j,j} = 0$ implies that $j = n - 1$. Consider the matrix $H(k-1)$, the end result of the $k-1$-th iteration. By the hypothesis on k we know that no diagonal elements in $H(k-1)$ are zero. In particular, for the r about to be chosen in Step 1 of the k-th iteration, we know that $h_{r,r}(k-1) \neq 0$ and that $h_{r+1,r+1}(k-1) \neq 0$. Now, suppose the r chosen in Step 1 is not $n - 1$. Let

$$\begin{pmatrix} \alpha & 0 \\ \beta & \lambda \end{pmatrix}$$

be the submatrix of $H(k-1)$ consisting of the r and $r+1$ rows of columns r and $r+1$. After Step 1 has been performed this submatrix becomes

$$\begin{pmatrix} \beta & \lambda \\ \alpha & 0 \end{pmatrix}.$$

At Step 2, we post-multiply the matrix by the unitary sub-matrix of Q

$$\begin{pmatrix} \beta^\star/\delta & -\lambda/\delta \\ \lambda^\star/\delta & \beta/\delta \end{pmatrix},$$

where $\delta = \sqrt{\beta\beta^\star + \lambda\lambda^\star}$. The result is the matrix

$$\begin{pmatrix} \delta & 0 \\ \alpha\beta^\star/\delta & -\alpha\lambda/\delta \end{pmatrix}.$$

Since λ and α are not zero (they were diagonal elements of $H(k-1)$), we know that δ and $-\alpha\lambda/\delta$, the two diagonal elements in the matrix, are also not zero. Note that since the rest of Q is the identity matrix none of the other diagonal elements is affected by the multiplication. Thus, at the end of Step 2, all diagonal elements are non-zero. Since Hermite reduction doesn't introduce any new zeros on the diagonal, the end result of the k-th iteration has all non-zero diagonal elements. But this contradicts the hypothesis on k and our assumption that $r < n-1$ was false. Note that for $r = n-1$ in order to have $h_{n-1,n-1}(k) = 0$, we must have $h_{n,n-1}(k-1) = 0$ and $h_{n-1,n-1}(k-1) \neq 0$.

Next we show that a relation for x must appear as a column of the matrix B. By Lemma 1, $xH_x = 0$. $BA = I_n$ implies $0 = xBAH_x = xBAH_xQ = xBH(k-1)$, where Q is an appropriate unitary $n-1 \times n-1$ matrix. Let $z = xB$. The above gives

$$(0, \ldots, 0) = xBH(k-1) = zH(k-1) = (\ldots, z_{n-1}h_{n-1,n-1}(k-1)).$$

Since $h_{n-1,n-1}(k-1) \neq 0$ then $z_{n-1} = 0$. Hence the $n-1$-th column of B is a relation for x. □

Lemma 6. *At any k-th iteration of the algorithm the diagonal entries of $H(k)$ satisfy the inequality $|h_{i,i}(k)| \leq 1$.*

Proof. We follow the α, β, λ definitions of the proof of Lemma 5 and use induction. For $k = 1$ the diagonal entries of $H(k)$ are those of H_x and $s_{j+1} \leq s_j \leq 1$ gives the required inequality. Assume that the inequality also holds up to $k-1$. The diagonal entries of $H(k)$ are equal to those of $H(k-1)$ except for row r where Step 1 Exchange occurs. When $r = n-1$, after the exchange, the r-th diagonal element is β. But $|\beta| \leq |\alpha|/\rho \leq 1$ because $\rho > 1$ and $|\alpha| \leq 1$ by induction. When $r < n-1$, after the exchange the r-th diagonal element is δ. But $|\delta| = |\alpha|t \leq 1$ since $t < 1$ and $|\alpha| \leq 1$. The

$r+1$-th diagonal element of H is $-\alpha\lambda/\delta$ (as in the proof of Lemma 5) so that $|-\alpha\lambda/\delta| = |\lambda|/t \leq |\alpha|$ because $|\lambda|^2 < |\lambda|^2 + |\beta|^2$ and $|\lambda| \leq |\alpha|t$. □

We show that every iteration of PSLQ causes a geometric monotonic increase in a certain function $\Pi(k)$ which is roughly the product of all the principal minors of the matrix $H(k)$. If a relation for x exists, this product will be bounded above and below. Assume x has some relation and as usual let M_x denote the norm of a smallest relation for x. We will need the following technical lemma in the proof of Lemma 9.

Lemma 7. *Consider the quotient*

$$q(A,B,t) = \frac{\min\{B,t\}\cdot\min\{A,1\}}{\min\{B,1\}\cdot\min\{A,t\}}$$

Suppose that the four positive real numbers $A, B, 1, t$ satisfy the three inequalities

$$A \geq B, \quad A \geq t, \quad 1 \geq t.$$

Then,

$$q(A,B,t) \quad \geq \quad 1.$$

Proof. Of the 16 possible choices in the min's, the inequality $A \geq t$ removes 8, $A \geq B$ removes 2, and $1 \geq t$ removes 1 leaving 5. These five are
$A \geq B \geq 1 \geq t$ with quotient $t/1 \cdot 1/t = 1$,
$A \geq 1 \geq B \geq t$ with quotient $t/B \cdot 1/t \geq t/1 \cdot 1/t = 1$,
$1 \geq A \geq B \geq t$ with quotient $t/B \cdot A/t = A/B \geq 1$,
$1 \geq A \geq t \geq B$ with quotient $B/B \cdot A/t = A/t \geq 1$,
$A \geq 1 \geq t \geq B$ with quotient $B/B \cdot 1/t = 1/t \geq 1$. □

Lemma 8. *For α, γ, M_x as above,*

$$\gamma^{n-2}M_x|\alpha| \geq 1.$$

Proof. By the choice of r in Step 1 Exchange, we have $\gamma^r|\alpha| \geq \gamma^j|h_{j,j}|$ for any j, $1 \leq j \leq n-1$, which implies

$$\gamma^{n-1}/|h_{j,j}| \geq \gamma^r/|h_{j,j}| \geq \gamma^j/|\alpha| \geq \gamma^1/|\alpha|,$$

for all j including that j_o for which $M_x \geq 1/|h_{j_o,j_o}|$ from Theorem 1. Thus $\gamma^{n-2}M_x \geq 1/|\alpha|$ and $\gamma^{n-2}M_x|\alpha| \geq 1$ □

Definition 6: (The Π function). Recall $\tau = \sqrt{1/\rho^2 + 1/\gamma^2}$. Define

$$\Pi(k) = \prod_{1\leq j\leq n-1} \min\{\gamma^{n-1}M_x, 1/|h_{j,j}(k)|\}^{n-j}.$$

Lemma 9. *For any $k > 1$ we have*

(i)

$$(\gamma^{n-1}M_x)^{\binom{n}{2}} \geq \Pi(k) \geq 1,$$

(ii)

$$\Pi(k) \geq \tau\Pi(k-1).$$

Proof. For the k's so far, $h_{j,j}(k) \neq 0$ for all $1 \leq j \leq n-1$. $M_x \geq 1$ and $1/|h_{j,j}(k)| \geq 1$ by Lemma 6. This gives

$$\min\{M_x, 1/|h_{j,j}(k)|\} \geq 1,$$

for all $1 \leq j \leq n-1$, which implies the right hand inequality of (i). On the other hand, it is always the case that $M_x \geq \min\{M_x, 1/|h_{j,j}(k)|\}$, which together with the fact that $\binom{n}{2} = n-1+\cdots+2+1$ and that $\gamma \geq 1$ gives the left hand inequality of (i).

The proof of part (ii) is more involved. Let r be given by the Step 1 Exchange of PSLQ. Recall the definitions of the two successive diagonal elements α, λ, the single off diagonal element β, $t = \sqrt{\beta\beta^* + \lambda\lambda^*}/|\alpha|$ in the Step 2 (Corner development) of the unitary matrix in terms of β and λ.

Suppose that $r < n-1$. Then only two diagonal elements change. These correspond to the 2×2 submatrix of H

$$\begin{pmatrix} \alpha & 0 \\ \beta & \lambda \end{pmatrix}$$

which after a single iteration becomes

$$\begin{pmatrix} \delta & 0 \\ \alpha\beta^*/\delta & -\alpha\lambda/\delta \end{pmatrix}.$$

But $|\delta| = |\alpha|t$ so that the absolute values of the of the α, λ diagonal elements are replaced by the absolute values of the $\delta, -\alpha\lambda/\delta$ diagonal elements. All the factors of $\Pi(k)$ are the same except these two so that

$$\frac{\Pi(k)}{\Pi(k-1)} = \left(\frac{\min\{\gamma^{n-1}M_x, 1/(|\alpha|t)\}}{\min\{\gamma^{n-1}M_x, 1/|\alpha|\}}\right)^{n-r} \cdot \left(\frac{\min\{\gamma^{n-1}M_x, t/|\lambda|\}}{\min\{\gamma^{n-1}M_x, 1/|\lambda|\}}\right)^{n-r-1}.$$

Set

$$A = \gamma^{n-1}M_x|\alpha|t \quad \textit{and} \quad B = \gamma^{n-1}M_x|\lambda|,$$

so that

$$\frac{\Pi(k)}{\Pi(k-1)} = \left(\frac{\min\{A,1\}}{\min\{A,t\}}\right)\cdot\left(\frac{\min\{B,t\}}{\min\{B,1\}}\cdot\frac{\min\{A,1\}}{\min\{A,t\}}\right)^{n-r-1}.$$

We now show that the assumptions for Lemma 7 hold. Note that $1 > t$ by the definition of t; also, $A \geq B$ since $|\alpha|t \geq |\lambda|$. By Lemma 8 we have $A \geq t\gamma \geq t$. By Lemma 7 we have

$$\frac{\Pi(k)}{\Pi(k-1)} \geq \frac{\min\{A,1\}}{\min\{A,t\}} \geq \frac{1}{t} \geq \tau.$$

Now suppose that $r = n-1$. By Step 3 Reduction, under one iteration the absolute value of the last diagonal element α is less than $|\alpha|\rho$. All the factors of $\Pi(k)$ except the last are the same so that

$$\frac{\Pi(k)}{\Pi(k-1)} \leq \frac{\min\{\gamma^{n-1}M_x, 1/(|\alpha|\rho)\}}{\min\{\gamma^{n-1}M_x, 1/|\alpha|\}} = \frac{\min\{A, t/\rho\}}{\min\{A,t\}}.$$

But we always have $\gamma^{n-2}M_x|\alpha| \geq 1$, so if $A \geq t/\rho \geq t$

$$\frac{\Pi(k)}{\Pi(k-1)} \geq 1/\rho \geq \tau.$$

By Lemma 8, $A \geq t\gamma \geq t$. If $t \leq A \leq t/\rho$ then

$$\frac{\Pi(k)}{\Pi(k-1)} \geq A/t \geq \gamma \geq \tau.$$

Thus for $r \leq n-1$, $\Pi(k) \geq \tau\Pi(k-1)$. □

Theorem 2. *Assume real or complex numbers, $n \geq 2$, $\tau > 1$, and that $0 \neq x \in \mathbb{K}^n$ has $\mathbb{O}(\mathbb{K})$ integer relations. Let M_x be the least norm of relations for x. Then PSLQ(τ) will find some integer relation for x in no more than*

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

iterations.

Proof. Suppose we have done k iterations, then from Lemma 6 and Lemma 7, $|h_{j,j}(k)| \neq 0$ and not all $|h_{j,j}(l)| < 1/M_x$ for $l < k$. By Lemma 6, $\Pi(0) \geq 1$ and by Lemma 7, $\Pi(k) \geq \tau^k$ so that

$$(\gamma^{n-1}M_x)^{\binom{n}{2}} \geq \tau^k$$

Taking natural logarithms of both sides of this inequality gives

$$\binom{n}{2}\log\left(\gamma^{n-1}M_x\right) \quad \geq \quad k\log\tau. \qquad \square$$

Corollary 2. *Let $\mathbb{K}$ be the real numbers $\mathbb{R}$ or the complex numbers $\mathbb{C}$. Fix $n > 1$ and assume given a unit n-tuple $x \in \mathbb{K}^n$ which has a relation $m_x \in \mathbb{O}(\mathbb{K})^n$ of least norm M_x. Then there exists a γ such that the algorithm PSLQ(τ) will construct some $\mathbb{O}(\mathbb{K})^n$ relation for x in no more than*

$$2 \cdot (\dim_{\mathbb{R}} \mathbb{K}) \cdot (n^3 + n^2 \log M_x)$$

iterations.

Proof. Let $\gamma = 2$. Then for either $\mathbb{K}$, $\tau > 1$, specifically, $1/\log \tau < 4 \dim_{\mathbb{R}} \mathbb{K}$. PSLQ($\tau$) takes $O(n)$ exact arithmetic operations per iteration, so in this sense finds relations in "polynomial time" $O(n^4 + n^3 \log M_x)$. □

5. Upper Bounds on Integer Relations

We compare the relation found by PSLQ to a shortest possible relation.

Lemma 9. *Suppose m is the relation found on the $k+1$-st iteration so that $h_{n-1,n-1}(k+1) = h_{n,n-1}(k) = 0$ and $h_{n-1,n-1}(k) \neq 0$. Then*

$$|m| = 1/|h_{n-1,n-1}(k)|.$$

Proof. At this iteration we have developed the matrix $A \in GL(n, \mathbb{O}(\mathbb{K}))$ where the $(n-1)$-st column of A^{-1} by Lemma 5 is m and the vector $Am^\star = e_{n-1}$ has as its only non-zero entry a 1 in the $(n-1)$-st position. Since $AP = TDQ$, $Qm^\star = D^{-1}T^\dagger Am^\star$, where $T^\dagger$ is the generalized inverse of T and D is a diagonal matrix with last entry $h_{n-1,n-1}(k)$, which is also the last entry of $D^{-1}T^\dagger Am^\star$. Because Q is unitary $|Qm^\star| = |m^\star|$. □

Theorem 3. *Let M_x be the smallest possible norm of any relation for x. Let m be any relation found by PSLQ(τ). For all $\gamma > \sqrt{4/3}$ for real vectors and for all $\gamma > \sqrt{2}$ for complex vectors*

$$|m| \leq \gamma^{n-2} M_x.$$

Proof. Assume we are at the k-th step of PSLQ where a Step 1 Exchange $r = n-1$ was made with $h_{n-1,n-1}(k) \neq 0$ and $h_{n-1,n-1}(k+1) = 0$. Then

$$\gamma^{n-1}|h_{n-1,n-1}(k)| \geq \gamma^j |h_{j,j}(k)|$$

for all $1 \leq j \leq n-2$ by the choice of r. Hence, by Theorem 1 and Lemma 8

$$M_x \geq 1/\max |h_{i,i}(k)| \geq \gamma^{2-n}/|h_{n-1,n-1}(k)| = \gamma^{2-n}|m|. \qquad \square$$

Comment on Theorem 3. For $n = 2$, Theorem 3 proves that any relation $0 \neq m \in \mathbb{O}(\mathbb{K}^2)$ found has norm $|m| = M_x$. In other words, PSLQ(τ) finds a shortest relation. For real numbers this corresponds to the case of the Euclidean algorithm, [Euclid, Book X], [Fowl79], [Knut81]. For complex numbers this corresponds to the case of an algorithm in [Schm75].

For $n = 3$, let $x = (113, 343, 311)$. This vector has a shortest relation $m_x = (7, -15, 14)$ with the shortest norm $|m_x| = M_x = 21.6794\ldots$. This can be verified directly, cf., [Kann88], [PoZa84], [Cohe93]. On the other hand, for $\tau = 1.0000\ldots$, $\gamma = 1.1547\ldots$, PSLQ(τ) in iteration 6 produces the relation $m_1 = (24, -7, -1)$. Indeed

$$M_x < |m_1| = 25.0199\ldots \leq \gamma M_x = 25.0333\ldots .$$

This relation appears from a zero in the second coordinate of the xA_6^{-1} vector. Continuing to iteration 8 gives the relations appearing from the first and second coordinates of the current xA_8^{-1} vector, $m_2 = (-17, -8, 15)$ and $m_3 = (41, 1, -16)$ of norms $24.0416\ldots$ and $44.0227\ldots$, respectively. The vector m_2 has smaller but not smallest norm. Continuing to iterations 9 and 10 gives the relations appearing from the first and second coordinates of xA_9^{-1} of $m_4 = (7, -15, 14)$ and $m_2 = (-17, -8, 15)$, so a shortest vector m_4 was eventually found. In iteration 11 the $h_{2,2}(11) = 0$ condition appeared for the first time giving the relation $m_5 = (-10, -23, 29)$ of norm $38.3405\ldots$.

This example is instructive in that various choices of the parameter τ give different outputs. The "legal" τ are such that $1 < \tau < 2$, although the PSLQ(τ) sometimes works for "illegal" τ outside of this interval. For the "legal" τ, $\tau = 1.1$, iteration 6 yields m_1, 8 yields m_2, m_3, 9 yields m_4, m_2, and 10 yields m_5. On the other hand, for $\tau = 1.8$, iterations 4, 5, 6 all yield only the shortest length relation m_4. For the "illegal" τ below 0.7 and above 2.1 the algorithm cycles indefinitely. The end point $\tau = 1.0$ gives essentially the same outputs as $\tau = 1.1$. The other end point $\tau = 2.0$ yields two new relations $m_6 = (1, -91, 100)$ and $m_7 = (0, -311, 343)$ of norms $135.2109\ldots$ and $463.0010\ldots$, respectively.

6. Multiple Relations.

A given unit vector $x \in \mathbb{K}^n$ may have 0, 1, 2, or up to $n - 1$ relations. Once a relation has been constructed, one of the coordinates of xB for the appropriate $B \in GL(n, \mathbb{O}(\mathbb{K}))$ will be zero, and the corresponding column of B will be a relation. The remaining $n - 1$ coordinates can be used to form a new unit vector in $y \in \mathbb{K}^n$. Apply PSLQ(τ) to this y. Any second relation so found will be integrally independent from the first and can be referred back to the original x. In this way as many as $n-1$ integrally independent relations for x can be constructed. We omit here the tangent discussion of using classical lattice reduction techniques to find integer relations; this is the case for the Recognize[ ] function in Mathematica^{TM} which calls the function LatticeReduce[ ], cf., [Cohe93], [CJOS83], [LaLS82]. Lattice reduction there applies typically only to integer relations for *integer* vectors. Integer relation finding here is directed specifically at integer or Gaussian integer relations for *real* or *complex number* vectors.

7. Variations of PSLQ(τ).

The algorithm PSLQ(τ) as stated may be performed for various "illegal" τ or "illegal" γ, and under these circumstances will find relations for some x vectors. This can happen for $\gamma < \sqrt{4/3}$ in the real case, for $\gamma < \sqrt{2}$ in the complex case, and for $\gamma < \infty$ in the quaternion case, so that $\tau < 1$ and the conclusions of Theorem 2 or Theorem 3 make no sense or have no apparent content. The reason for this apparent anomaly is that for a specific n-tuple x the actual field or division ring constant ρ bound in Lemma 4 is not universal and could depend upon an input vector x. Say ρ_x gives a bound such as that of Lemma 4 for some special x or collection of them. Then there may be an "illegal" γ so that $\tau_x = 1/\sqrt{1/\rho^2 + 1/\gamma^2} > 1$. For such x one could expect to see some relation emerge before the number of iterations indicated by Theorem 2 for this $\tau_x = \tau$.

On the other hand, it is possible to use the real PSLQ(τ) algorithm to find complex and quaternion relations at the expense of doubling and quadrupling the dimension. For example, suppose $z = x + yi + uj + vk$ is a vector in $\mathbb{H}^n$ with vector components $x, y, u, v \in \mathbb{R}^n$. Suppose the corresponding relation is $m = a + bi + cj + dk$ which is a lattice point in $\mathbb{W}^n$ with integral vector components $a, b, c, d \in \mathbb{Z}^n$. Then $zm^\star = 0$ implies four integer relations among the interlaced and suitably sign changed coordinates of z. For the first set $\sum_{1\le j\le n}(a_j x_j - b_j y_j - c_j u_j - d_j v_j) = 0$ and one can apply real PSLQ(τ) to the real $4n$-tuple $(\ldots, x_j, y_j, u_j, v_j, \ldots)$. There are three others which are similar. A relation for z will be in the intersection of the four associated lattices. Alternatively, one can give a PSLQ(τ) algorithm along the lines of [HJLS89, *Section 5. Finding simultaneous integer relations*].

8. Computer Implementation of PSLQ(τ)

The PSLQ(τ) algorithm can be implemented using ordinary floating point arithmetic on a computer. Using double precision (i.e., 64-bit) arithmetic, relations of two or three digits in size can be recovered for n up to five or so. Beyond this level, precision is quickly exhausted, and recovered relations and norm bounds are meaningless. Thus a serious implementation of PSLQ (or any other integer relation algorithm for real numbers) must employ some form of multiprecision arithmetic. The authors employed the MPFUN multiprecision translator and computation package. The Fortran-77 version of this software is described in [Bail93], and the newer Fortran-90 version is described in [Bail95]. A C++ translator that employs these routines is also now available. Alternatively, one may employ the multiprecision facilities of symbolic math software packages, such as Maple, Pari or Mathematica^{TM}.

The descriptions presented here of computer implementation of PSLQ(τ) are for the case of the real number system. Extensions to the case of the complex and quaternions number systems are straightforward, provided one's multiprecision system supports these datatypes.

One key to an efficient implementation is to utilize a simplified version of Hermite reduction and the associated update. As noted in Lemma 3 above, Hermite reduction can be done more efficiently by a triply nested loop. In fact, the update operations associated with Hermite reduction (updating x, H, A and B) can also be done in a loop of this form. Further, if these updates are done in this manner, then it is not necessary to compute the D matrix. This simplified scheme is as follows. In the initialization step, Hermite reduction and the subsequent updates are replaced with the following:

For i from 2 to n, for j from $i-1$ to 1 (step -1), set $t = \text{nint}(h_{i,j}/h_{j,j})$ and replace x_j by $x_j - tx_i$; then for k from 1 to j replace $h_{i,k}$ by $h_{i,k} - th_{j,k}$; for k from 1 to n replace $a_{i,k}$ by $a_{i,k} - ta_{j,k}$ and replace $b_{k,j}$ by $b(k,j) + tb(k,i)$.

Step 3 is also replaced with this, except i is incremented from $r+1$ to n, and j is decremented from $\min\{i-1, r+1\}$ to 1. Here r denotes the row index selected in Step 1. These more restrictive limits on i and j merely reflect the fact that $t = 0$ outside these limits.

Obviously in a computer implementation some care must be taken in testing for zero. This is typically done by checking that the absolute value of the tested value is less than the "epsilon" appropriate for the level of numeric precision being used. Also, a run should be terminated if any entry of the A matrix exceeds the level of numeric precision being used (so that these integer values can no longer be represented exactly).

The level of working precision required for PSLQ is generally only a few digits greater than the accuracy of the input x vector. Along this line, if one wants to recover (or to exclude) relations of size d digits, then the input data must be specified to at least nd digits in order to obtain numerically meaningful results. The significance of a recovered result can be measured by noting the ratio between the multiprecision epsilon and the largest entry of the updated x vector when a relation is recovered. If this ratio is very small, such as 10^{-40}, then one can be fairly certain that the relation produced by PSLQ is a real relation. But if this ratio is only a few orders of magnitude below unity, then the result is suspect, and higher accuracy in the input data, as well as correspondingly higher working precision, is required.

The above implementation is satisfactory for most applications. For more demanding applications, a "two-level" implementation is significantly faster. In a two-level implementation, most operations are performed in ordinary double precision arithmetic, with occasional updates of multiprecision arrays using multiprecision arithmetic. This two-level scheme can be described as follows. Here the prime notation is used to denote double precision approximations to multiple precision values.

To initialize, perform the initialization step as described above using full precision. Then perform a "double precision initialization": (1) set $x' = x / \max_{i,j} |x_j|$ and set $H' = H$; (2) perform a LQ decomposition on H', using double precision arithmetic, setting H' to be the lower triangular part; (3) set $A' = B' = I_n$.

PSLQ iterations are then performed as above on the arrays x', H', A' and B', using double precision arithmetic. Some care must be taken to insure numerical accuracy in these iterations. Obviously these iterations must be halted before entries in A' grow so large (9×10^{15} on IEEE systems) that they cannot be exactly represented as double precision values. In the authors' implementation, double precision iterations are halted when the largest entry of A' exceeds 10^{10}. Tests for zero in these iterations must reflect the accuracy of double precision arithmetic — the authors used an "epsilon" of 10^{-13} here. As an additional measure to insure numerical integrity, the authors' code aborts the modified Hermite reduction procedure (and restores arrays to their previous values) if the multiplier q exceeds 10^7.

When the double precision iterations are halted, either due to large entries in A', or to a tentative zero in x' or H', it is necessary to perform a "multiprecision update": (1) replace A by $A'A$, replace B by BB', replace H by $A'H$, and replace x by xB'; (2) check for zero entries in x, using the multiprecision epsilon. If no zeroes are found, then a double precision initialization is performed, followed by more double precision PSLQ iterations.

One detail has been omitted here. In some cases, the entries of the updated x vector have such a large dynamic range (greater than 10^{10} in the authors' implementation) that when converted to double precision, additions and subtractions would produce results of questionable reliability. In these cases it is necessary to perform PSLQ iterations on the multiprecision arrays, using multiprecision arithmetic, for a number of iterations until this large dynamic range is eliminated. If this situation is encountered on any iteration other than the very first, a multiprecision LQ decomposition of H must be performed prior to performing these multiprecision iterations (so that the H array contains the same entries as the H array defined in the PSLQ algorithm statement).

9. Summary of the Literature

The problem of finding integer relations among sets of rational and real numbers is quite old. When $n = 2$ this problem can be solved for rationals by the first Euclidean algorithm in Euclid, Book VII, and for reals by the second Euclidean algorithm given in Book X, cf., [Knut81], [Cohe93], [Shim94]. Generalizations of this algorithm to higher real dimensions were proposed without proof by many authors, including Jacobi [Ja1868], Hermite [He1850], Poincaré [Po1882], Perron [Perr07], Brun [Brun19, Brun57] and Szekeres [Szek70].

The first integer relation finding algorithm with proofs for the case of real numbers was discovered in 1977 by Ferguson and Forcade, [FeFo79, FeFo82]. These algorithms were shown to be polynomial time in the logarithm of the size of a smallest relation. They were not shown to be polynomial in the dimension. Since then, other algorithms for finding relations for real vectors have appeared in [Berg80], [Ferg86]. [Bail88] reports on a computer implementation of [Ferg86]. The sequence including [HJLS89] (HJLS), [BaFe91] and [ArFe93] (PSLQ), [BaBG94] (a concise statement of PSLQ), and [RoSc95] (a stable variation of HJLS) will be discussed below.

These algorithms all depend upon an orthogonal decomposition of some sort. See [GoVL90], for a list of various orthogonalization algorithms and their differences. PSLQ is of the QR type. HJLS follows the lattice reduction work of [LLjL82], which is classical Gram-Schmidt type, cf. [Cohe93]. This difference may explain some of the differences observed between PSLQ and HJLS, cf. [BaFe91].

Rigorous proofs that the algorithm under investigation must find a relation if one exists appeared in [FeFo79, Berg80, FeFo82, Ferg86]. All of these proofs gave a linear bound in the logarithm of the size of a relation, but were not known to be polynomial in the dimension. [Berg80] and [Ferg86] had unsatisfactory proofs in the sense that they were shown to be at worst exponential in the dimension rather than polynomial in the dimension. This unsatisfactory state of affairs was resolved affirmatively with the proofs that appeared in [HJLS89] for the "small integer relation algorithm". We will refer to this "small integer relation algorithm" as HJLS, as stated in [HJLS89, Section 3] as a reflection of that in [Berg80, Section 3]. In fact, this proof in [HJLS89] was the first appearance in the literature of a polynomial time bound for a relation finding algorithm, polynomial in both dimension and logarithm of relation size.

This important progress was made when [HJLS89] combined two independent streams of research, [FeFo79, Berg80, FeFo82, Ferg86, Ferg88] and [LLjL82, ScEu94, Cohe93, ScRo95]. Inspired by the polynomial result, but not the details, the first author of this paper formulated what he thought was a new algorithm [BaFe92, ArFe92] and gave a polynomial proof. This proof was independent of that of [HJLS89], a different analysis, but flawed by giving a slightly higher degree polynomial in the dimension than the polynomial proof given in [HJLS89]. This algorithm in [BaFe92, ArFe92] was called PSLQ and had the advantage of the adjustable parameter γ or τ. Applications and implementation of this earlier version of PSLQ(τ) were described in [BaBG94, Bail95, BaBP95]. These implementations showed that the parameters were a helpful feature of the algorithm. The bound on iterations for HJLS proven in [HJLS89] was $O(n^3 + n^2 \log_2 M_x)$; this is consonant with the bound proven in this paper for PSLQ($\sqrt{2}$). The subsequent paper [RoSc95] included parameters as well as addressing a certain issue of stability.

Examples can be generated from the Mathematica^{TM} implementations described in Section 10. Specifically, in three dimensions, consider the triple $x = (11, 27, 31)$. We list the sequence of A^{-1} matrices for each algorithm. A relation if found will be constructed as a column of one of these A^{-1} matrices.

For PSLQ(1.1547) the successive iterations $k = 0, 1, 2, 3, 4$, yield the five A^{-1} matrices

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 3 & 8 & 1 \\ -3 & -7 & -1 \end{pmatrix}, \begin{pmatrix} -2 & 1 & 0 \\ 2 & 3 & 1 \\ -1 & -3 & -1 \end{pmatrix},$$

$$\begin{pmatrix} 3 & -2 & 0 \\ 1 & 2 & 1 \\ -2 & -1 & -1 \end{pmatrix}, \begin{pmatrix} -1 & -8 & -2 \\ 5 & 9 & 2 \\ -4 & -5 & -1 \end{pmatrix}.$$

Note that PSLQ has constructed two relations appearing as the first and second columns of the last matrix, iteration $k = 4$.

For HJLS the successive iterations $k = 0, 1, 2, 3, 4, 5, 6$ yield the seven A^{-1} matrices

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & -1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & -1 \end{pmatrix}, \begin{pmatrix} 1 & -2 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & -1 \end{pmatrix},$$

$$\begin{pmatrix} 1 & 0 & -2 \\ 0 & 1 & 2 \\ 0 & -1 & -1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & -2 \\ 1 & 3 & 2 \\ -1 & -3 & -1 \end{pmatrix}, \begin{pmatrix} 0 & -2 & -1 \\ 1 & 2 & 5 \\ -1 & -1 & -4 \end{pmatrix}.$$

Note that only one relation is found; it appears in the last column of the last matrix, iteration $k = 6$. The report [ScEu94] claimed that HJLS is a special case of PSLQ(τ) for $\gamma = \sqrt{2}$ or equivalently $\tau = \sqrt{4/3}$. The example just given shows that this claim is not true. This fact is also underscored by the results reported in [BaFe92], which show that HJLS, as stated in [HJLS89], often requires a level of numeric precision far higher than that of the input data, whereas PSLQ(τ) typically only requires 10 digits or so more than the input data. Indeed, it is clear from the results in [BaFe92] that without some suitable modification to the HJLS algorithm, such as that proposed in [ScRo95], it is not usable for many problems of interest.

The various algorithms in the literature stand independently of any proofs. Though the proofs were exponential, the algorithms stated in [FeFo79], and in [FeFo82], and again in [Ferg86] were parametric. The parameter b in [FeFo79, FeFo82] satisfies $1 < b < 2$ whereas in [Ferg86] the parameter γ is emphasized. The algorithm in [Berg80, Sect. x] seems closest to PSLQ($\sqrt{4/3}$) with the τ parameter set by $\gamma = \sqrt{2}$. This parameter choice appears in [Berg80, Sect. x] without the [LLjL82] setting and reappears in [HJLS89] as the "small integer relation algorithm", which we call HJLS, rewritten in the [LLjL82] language and accompanied by a polynomial time proof.

Bergman discussed the complex case of finding gaussian integer relations for complex vectors in [Berg80, Sect. 5: Variants]. Bergman also gave an algorithm for the simultaneous real vector case in [Berg80, Sect. 7]. Following Bergman, the paper defining HJLS for simultaneous real vectors, [HJLS89, cf., Sect. 5], implicitly includes the complex and quaternion vector case as well. As an alternate approach, inspired by [Shim94], in this paper we have extended the base field of PSLQ(τ) to these division rings and introduced unitary matrices into the algorithm directly. The proof given here of polynomial number of iterations covers the real and complex cases, but fails for quaternions. However, the quaternion version of PSLQ(τ) performs reasonably well experimentally in finding hamiltonian integer relations for quaternion vectors. This was explained in Section 8.

10. Working Mathematica^{TM} code for the algorithms PSLQ, PSOS, HJLS, BRUN.

Attached to this paper as an appendix is a list of working Mathematica^{TM} procedures for PSLQ and a few other lattice relation algorithms. Each algorithm **algo** is given by an initialization procedure `initalgo[ ]` followed by one iteration procedure `algo[ ]`. The input is some tuple such as

$$(x, A^{-1}, A, H, 1/\max_{1\leq j\leq n-1} |h_{j,j}(H)|, k)$$

where k is the iteration number, where $h_{j,j}(H), 1 \leq j \leq n-1$, are the diagonal elements for a matrix $A \in GL(n, Z)$ with $xH = 0$ for $H \in M(n \times n-1, R)$, H a rank $n-1$ and $n \times n-1$ lower trapezoidal matrix, $xH = 0$. The output is the sextuple

$$(xB^{-1}, A^{-1}B^{-1}, BA, BH, 1/\max_{1\leq j\leq n-1} |h_{j,j}(BH)|, k+1)$$

for the matrix $A \in GL(n, Z)$. PSLQ and PSOS are written in real $GL(n, Z)$ and complex $GL(n, Z+iZ)$ form. Variations from this format will be clear from the actual Mathematica^{TM} notation itself.

Each algorithm is characterized by the sequence of matrices developed at each iteration, from the group $GL(n, Z)$ or the group $GL(n, Z+iZ)$,

$$A_1, A_2, \ldots, A_k, \ldots$$

("partial quotients") and their accumulated products ("convergents")

$$B_1, B_2, \ldots, B_k \ldots.$$

The general scenario is that $B_k H_x$ will converge to zero if the coordinates of x are integrally linearly independent, $B_k H_x$ gives lower bounds on the size of possible relations, the rows of B_k will converge to x, and if x has a relation at all, then one will appear as a column of B_k^{-1} for some k. There are counterexamples for BRUN (relations not always found), no counterexamples are known for PSOS. This paper proves that PSLQ(τ) fits this scenario where k is polynomially bounded by $\binom{n}{2} \log_\tau (\gamma^{n-1} M_x)$ with M_x the least norm of any relation. HJLS fits this scenario as well, cf., [HJLS89].

In these listings, `s0[x,j,d]` computes the square root of the j-th partial sum of squares for x, precision d decimals, `x0[x,d]` computes the unit vector with the same direction as x, `h0[x,d]` constructs the initial H_x matrix out of the partial sums of squares, `hLQ[H]` is the maximum of the absolute values of the diagonal elements in the LQ decomposition of H. By Theorem 1, the reciprocal of this number gives a lower bound on the L_2 norm of any relation for x. `brun` is Brun's algorithm: this is a simple generalization of Euclid's anthyphairesis algorithm $n = 2$ to $n \geq 2$ quantities. Anthyphairesis, or $\alpha\theta\eta\phi\alpha\iota\rho\epsilon\sigma\iota\varsigma$, means "continually subtracted in turn from", cf.,[Fowl79]. This concept appears in Euclid in Book VII and Book X, [Euclid]. Brun's algorithm, cf., [Brun19], [Brun57], is a natural generalization of $\alpha\theta\eta\phi\alpha\iota\rho\epsilon\sigma\iota\varsigma$ from a pair of numbers to a list of numbers. This generalization is rediscovered by almost everyone working in this area. Brun's algorithm can cycle and does not always find relations. However, according to a theorem of Forcade, cf., [Forc81], Brun's algorithm finds relations almost everywhere. `pslq` is the PSLQ algorithm described in this paper; `tau` and `rho` are the PSLQ parameters defined in this paper. `psos` is the partial sum of squares algorithm defined in [Ferg88]. `hjls` is the "small integer relation algorithm" defined in [HJLS89]; see also [Berg80], [Ferg87].

11. Open Questions

1) Is there a relation finding algorithm that finds one of the shortest relations (there may be more than one with the same minimum height), with a guaranteed iteration count that is a polynomial function of the dimension?

2) What are the best choices for the parameter τ or γ relative to the number of iterations, time, and precision requirements of PSLQ?

3) Does PSOS have a counterexample in dimension 5 or less? The complete Mathematica^{TM} definition of PSOS for real and complex numbers, with possible extension to quaternions, is described in Section 10.

12. Acknowledgments

The authors thank (in alphabetical order) Peter Borwein, M. Euchner, Rod Forcade, Jeff Lagarias, Alyson Reeves, Robert Riley, M. L. Robinson, Carsten Rössner, Claus Schnorr, and Francis Sullivan for their helpful and motivating comments about PSLQ. Specifically we thank Alyson Reeves for her lucid rewriting of the proof of Lemma 5 and Rod Forcade for some counterexamples.

References

[ArFe92] Steve Arno and Helaman Ferguson, *A new polynomial time algorithm for finding relations among real numbers*, Supercomputing Research Center Tech Report SRC-93-093 (March 1993), 1–13.

[BaFe91] D. H. Bailey and H. R. P. Ferguson, *A polynomial time, numerically stable integer relation algorithm*, SRC Technical Report SRC-TR-92-066; RNR Technical Report RNR-91-032 (16 December 1991; 14 July 1992), 1–14.

[BaBG94] D. H. Bailey, J. Borwein, and R. Girgensohn, *Experimental evaluation of Euler sums*, Experimental Mathematics **3** (October 1994), 17 - 30.

[BaBP95] D. H. Bailey, P. Borwein, and S. Plouffe, *On the rapid computation of various polylogarithmic constants*, Cf., Science News **148** (28 October 1995), no. 18, http://www.cecm.sfu.ca/personal/pborwein/, 279.

[Bail88] D. H. Bailey, *Numerical Results on the Transcendence of Constants Involving π, e, and Euler's Constant*, Mathematics of Computation **50** (January 1988), no. 181, 275 - 281.

[Bail93] D. H. Bailey, *Multiprecision Translation and Execution of Fortran Programs*, ACM Transactions on Mathematical Software **19** (1993), no. 3, 288 - 319.

[Bail95] D. H. Bailey, *A Fortran-90 Based Multiprecision System RNR-94-013*, ACM Transactions on Mathematical Software, to appear (January 6, 1995), 1 - 10.

[Berg80] G. Bergman, *Notes on Ferguson and Forcade's generalized Euclidean algorithm*, University of California at Berkeley, unpublished (1980), no. November, 823–826.

[Brun19] V. Brun, *En generalisatiken av kjedebroøken, I, II*, Norske Videnskapsselskapets Skrifter I. Matematisk Naturvidenskapelig Klasse **6** (1919, 1920), 1-29, 1-24.

[Brun57] V. Brun, *Algorithmes euclidiens pour trois et quatre nombres*, tenu a Helsinki 18–23 août 1957, Treizième congrès des mathematiciens scandinaves (1958), 46 - 64.

[Cohe93] H. Cohen, *A Course in Computational Algebraic Number Theory: 2.5.2. The Gram-Schmidt Orthogonalization Procedure, 2.6.1. The LLL Algorithm, 2.7.2. Linear and Algebraic Dependence Using LLL*, Graduate Texts in Mathematics 138, Springer-Verlag, Berlin Heidelberg New York, 1993.

[CJOS93] M. J. Coster, A. Joux, B. A. LaMacchia, A. M. Odlyzko, C. P. Schnorr, J. Stern, *Improved Low-Densidy Subset Sum Algorithms*, Computational Complexity (1992-3).

[Euclid] Euclid, translated from the text of Heiberg with introduction and commentary by Sir Thomas L. Heath, *The Thirteen Books of Euclid's Elements, Book VII, Proposition 1, Volume II, pages 296-7 [integers], Book X, Proposition 2, Volume III, pages 17-20 [reals]*, Second Edition, revised with additions, unabridged, Volumes I, II, III, Dover Publications, Inc., New York, 1956.

[FeFo79] H. R. P. Ferguson and R. W. Forcade, *Generalization of the Euclidean algorithm for real numbers to all dimensions higher than two*, Bulletin (New Series) of the American Mathematical Society **1** (1979), 912 - 914.

[FeFo82] H. R. P. Ferguson and R. W. Forcade, *Multidimensional Euclidean Algorithms*, (Crelle's) Journal für die reine und angewandte Mathematik **334** (1982), 171 - 181.

[Ferg86] Helaman Ferguson, *A Short Proof of the Existence of Vector Euclidean Algorithms*, Proceedings of the American Mathematical Society **97** (May 1986), no. 1, 8 - 10.

[Ferg87] Helaman Ferguson, *A non-inductive $GL(n, Z)$ algorithm that constructs integral linear relations for n Z-linearly dependent real numbers*, Journal of Algorithms (1987), no. 8, 131 - 145.

[Ferg88] Helaman Ferguson, *PSOS: A new integral relation finding algorithm involving partial sums of squares and no square roots*, Abstracts of the American Mathematical Society **9** (March 1988), no. 56; 88T-11-75, 214.

[Forc81] Rodney W. Forcade, *Brun's Algorithm*, unpublished manuscript (November 1981), 1 - 27.

[Fowl79] David Fowler, *Ratio in Early Greek Mathematics*, Bulletin (New Series) of the American Mathematical Society **1** (November 1979), no. 6, 807 - 846.

[GoVL90] G. H. Golub and C. F. Van Loan, *Matrix Computations: 5.2 The QR Factorization, 5.2.7 Classical Gram-Schmidt, 5.2.8 Modified Gram-Schmidt*, 2nd Edition, The Johns Hopkins University Press, Baltimore, Maryland, 1990.

[He1868] C. Hermite, *Extraits de lettres de M. Ch. Hermite à M. Jacobi sur differénts objets de la théorie de nombres*, (Crelle's) Journal für die reine und Angewandte Mathematik (1850), no. 3, 4, 261 - 315.

[HJLS89] J. Hastad, B. Just, J. C. Lagarias, and C. P. Schnorr, *Polynomial time algorithms for finding integer relations among real numbers*, SIAM J. of Comput. **18** (1989), 859 - 881.

[Ja1868] C. G. J. Jacobi, *Allgemeine Theorie der Kettenbruchahnlichen Algorithmen, in welchen jede Zahl aus drei vorhergehenden gebildet wird (Aus den hinterlassenen Papieren von C. G. J. Jacobi mitgetheilt durch Herrn E. Heine.)*, Journal für die reine und Angewandte Mathematik **69** (1868), no. 1, 29 - 64.

[Kann88] R. Kannan, *Lattices, basis reduction, and the shortest vector problem*, Colloquia Mathematica Societatis János Bolyai, Theory of Algorithms, Pécs, (Hungary) **44** (1984), 283-311.

[Knut81] D. E. Knuth, *The Art of Computer Programming, Vol. 2 Seminumerical Algorithms: 4.5.2. The Great Common Divisor, 4.5.3. Analysis of Euclid's Algorithm*, Second Edition, Addison-Wesley, Reading, MA, 1981.

[LaLS82] J. C. Lagarias, H. W. Lenstra Jr., and C. P. Schnorr, *Korkin-Zolotarev Bases and Successive Minima of a Lattice and its Reciprocal Lattice*, Combinatorica **10** (1990), no. 4, 333 - 348.

[LLjL82] A. K. Lenstra, H. W. Lenstra Jr., and L. Lovasz, *Factoring polynomials with rational coefficients*, Math. Ann. (1982), no. 21, 515 - 534.

[LoSc92] Laszlo Lovasz and Herbert E. Scarf, *The Generalized Basis Reduction Algorithm*, Mathematics of Operations Research **17** (August 1992), no. 3, 751 - 764.

[Perr07] O. Perron, *Grundlagen für eine Theorie des Jacobischen Kettenbruchalgorithmus*, Math. Ann. (1907), no. 64, 1 - 76.

[PoZa84] M. Pohst and H. Zassenhaus, *Algorithmic Algebraic Number Theory, Chapter 3: Methods from the Geometry of Numbers*, Encyclopedia of Mathematics and its Applications, Cambridge University Press. New York, 1989, pp. xiv, 465.

[Po1884] H. Poincaré, *Sur une Généralisation des fractions continues*, Comptes Rendus Acad. Sci. Paris **99** (1884), 1014 - 1016.

[Schm75] Asmus L. Schmidt, *Diophantine Approximation of Complex Numbers*, Acta Mathematica **134** (1975), 1 - 85.

[ScEu94] M. Euchner and C. Schnorr, *Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems*, Proceedings of the FCT'91 (July 1991), 1-21.

[Schn94] C. Schnorr, et al, *Referees Report on "A polynomial time, numerically stable integer relation algorithm"*, submitted to editor of Mathematics of Computation (December 1994), 1-8.

[ScRo95] C. Rössner and C. P. Schnorr, *A stable integer relation algorithm*, FB Mathematik/Informatik Universität Frankfurt **TR-94-016** (1994), 1 - 11.

[ScRo95] C. P. Schnorr, *A More Efficient Algorithm for Lattice Basis Reduction*, Journal of Algorithms **9** (1988), 47 - 62.

[Shim94] G. Shimura, *Fractional and Trigonometric Expressions for Matrices*, The American Mathematical Monthly **101** (October 1994), no. 8, 744 - 758.

[Szek70] G. Szekeres, *Multidimensional continued fractions*, Ann. Univ. Sci. Budapest Eötvös Sect. Math. **XIII** (1970), 113 - 140.

HELAMAN FERGUSON AND STEVE ARNO: CENTER FOR COMPUTING SCIENCES, 17100 SCIENCE DRIVE, BOWIE, MD 20715-4300 **helamanf@super.org** AND **arno@super.org**;

DAVID H. BAILEY: NASA AMES RESEARCH CENTER, MAIL STOP T27A-1, MOFFETT FIELD, CA 94035-1000 **dbailey@nas.nasa.gov**

NAS TECHNICAL REPORT

NAS

Title:

ANALYSIS OF PSLQ, AN INTEGER RELATION FINDING ALGORITHM

Author(s):

Helaman R. P. Ferguson, David Bailey, and Steve Arno

Reviewers:

"I have carefully and thoroughly reviewed this technical report. I have worked with the author(s) to ensure clarity of presentation and technical accuracy. I take personal responsibility for the quality of this document."

Two reviewers must sign.

Signed: ______________________

Name: MAURICE YARROW

Signed: ______________________

Name: ______________________

After approval, assign NAS Report number.

Branch Chief:

Approved: D H Bailey

Date: 4-12-96

NAS ReportNumber: NAS-96-005