18.784 PAPER: FINDING INTEGER RELATIONS VIA PSLQ

ADELMO MORRISON OROZCO

Contents

1. Introduction 1
2. Warmup 2
3. Prior Attempts and Alternative Algorithms 3
4. The PSLQ Algorithm 3
References 11

1. Introduction

We study algorithms for the "integer relation problem". Let $x_1, \cdots, x_n \in \mathbb{R}$ be arbitrary real numbers, find integers $t_1, \cdots, t_n \in \mathbb{Z}$ that are not all zero such that

$$\sum_{i=1}^{n} t_i x_i = 0.$$

We can express the above as vectors $\mathbf{x} = (x_i)_{i\in[n]}$ and $\mathbf{t} = (t_i)_{i\in[n]}$, to rephrase the integer relation problem as finding non-zero $\mathbf{t} \in \mathbb{Z}^n$ so

$$\langle \mathbf{x}, \mathbf{t} \rangle = 0,$$

where $\langle \cdot, \cdot \rangle$ is the usual Euclidean inner product.

Integer relations have many important applications. For instance, as can be seen in [BLe00], they are commonly used to test if a number has special properties. In particular, if a number is "algebraic" with low degree. A number $\alpha \in \mathbb{C}$ is said to be *algebraic* if there exists a polynomial p over the integers $\mathbb{Z}$ such that $p(\alpha) = 0$. The minimum degree over all the polynomials where $p(\alpha) = 0$ denotes the degree of α. These class of numbers plays a crucial role in number theory and Diophantine approximation.

Algorithms to find integer relations can be used to test if a number is algebraic with degree at most n. Given a number α, let $\mathbf{x} = (1, \alpha, \alpha^2, \cdots, \alpha^n)$. Apply the integer relation algorithm to $\mathbf{x}$. If an integer relation is found, then we have found $t_i \in \mathbb{Z}$ such that $\sum_{i=0}^{n} \alpha^i \cdot t_i = 0$. This implies for the degree n polynomial with coefficients given by $\mathbf{t}$, $p(x) = \sum_{i=0}^{n} t_i x^i$, α is a root and hence algebraic.

It has also been used in recreational mathematics more generally. An important example is the Bailey–Borwein–Plouffe formula for π found using an integer relation

Date: May 12, 2026.

algorithm (see [BBBP97] for further exposition on the story). The formula states that

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

Bailey, Borwein, and Plouffeset let $\mathbf{x}$ have

$$x_i = \sum_{k=0}^{\infty} \frac{1}{16^k} \left(\frac{1}{8k+i} \right),$$

for all $i = 1, \cdots, 7$ and $x_8 = \pi$, applied the integer relation algorithm, re-arranged the found equation, and then proved the above formula analytically.

The outline of our exposition is as follows. We first discuss simplifying assumptions for the integer relation problem and the case for 2 dimensions as a warm-up in section 2. We introduce and motivate the subroutines of the PSLQ algorithm in section 4.1, section 4.2, section 4.3, and then the full completed version of the algorithm in section 4.4. Our exposition will emphasize the geometric intuition and give sufficient, but not overwhelming, detail of the algorithm analysis. In particular, we give a couple of mathematical perspectives (composed on existing insights across the literature) on key steps of the algorithms to illuminate the geometric perspective.

2. Warmup

2.1. **Simplifying Assumptions.** In order to avoid trivial edge cases later, we make the following assumptions. Without loss of generality, assume that $\mathbf{x}$ is non-zero. If $\mathbf{x} = 0$ any integer relation works. In fact, we can assume that every entry of $\mathbf{x}$ is non-zero. If $\mathbf{x}$ has any $x_i = 0$, the corresponding coefficient a_i of the integer relation can be any real number. Additionally, assume the ℓ_2-norm of $\mathbf{x}$, $||\mathbf{x}||_2$, is one. The integer relation is scale invariant: re-scaling all entries x_i by any fixed non-zero constant does not change any integer relation.

2.2. **2D Case.** As a warm-up, we first present a simple algorithm for the case in 2 dimensions. We can find an integer relation between $x_1, x_2 \in \mathbb{R}$ if it exists by an application of an extension of the Euclidean algorithm. In particular, finding an integer relation between x_1, x_2 reduces to finding the rational number $q \in \mathbb{Q}$ such that $x_1/x_2 = q$. This ratio is rational exactly if they have an integer relation, and the ratio (if in $\mathbb{Q}$) can be found by using the continued fraction expansion of the the ratio x_1/x_2 without ever explicitly dividing these numbers.

Using the same idea as the euclidean algorithm, we can find the rational representation of q using the continued fractions algorithm (e.g. [Sch80, Chapter 1]). Simply take $a_0 = \lfloor q \rfloor + \{q\}$, where the brackets indicate the fractional component, and recurse on $1/\{q\}$ to get a_1, and so on. Each of these steps can be done without

directly calculating q. We will get a continued fraction

$$q = a_1 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{a_4 + \cdots}}}}$$

equal to the desired fraction q. This will terminate if $q \in \mathbb{Q}$. The integer relation case for $n > 2$, however, is not obvious (the above scheme will not directly work) and will exploit a more geometric perspective on the integer relation problem.

3. Prior Attempts and Alternative Algorithms

Given section 2.2, it can be seen that an integer relation algorithm for dimensions $n \geq 3$ would, in some sense, be a generalization of the euclidean algorithm. Thus, the study of high-dimensional integer relation algorithms has been of great interest. The first efficient algorithm was discovered by Ferguson and Forcade [FF79]. The HJLS algorithm [HsJLS89] named after Hastad, Just, Lagarias, and Schnorr, was discovered earlier than PSLQ. Nevertheless, we give an exposition to the PSLQ algorithm since it is much better in practice. Recall that algorithms to find integer relations are typically used when there are irrational input entries. Unfortunately, we cannot represent these numbers in a computer – we do not have access to an idealized machine that can compute on the reals. Therefore, given an input of a known precision, we must increase the precision of the intermediate computations accordingly to avoid too many accumulated errors. When the amount of precision necessary to run the algorithm is not much larger than the input precision we say an algorithm is *numerically stable*. The PSLQ algorithm is much more numerically stable than the HJLS algorithm, and therefore we give exposition to the former. The HJLS and PSLQ algorithm are in a sense equivalent (see [Bor02, Appendix B, Theorem 7] for a formal statement), so in the presentation of PSLQ the reader will obtain a sufficient exposition to the main ideas of HJLS as well. In the presentation of the algorithm will assume that we can perform basic arithmetic operations (addition, subtraction, multiplication, and division) with arbitrary precision in our computation model. However, finite but large enough precision is sufficient for practical purposes.

Algorithms to find integer relations $\mathbf{t}$ of $\mathbf{x}$ will actually just find integer relations with norm at most M for some parameter M, assuming such a relation exists. This is indeed necessary since [BJMadH88] showed the problem of deciding whether an integer relation exists is undecidable (unsolvable by any algorithm in finite time).

4. The PSLQ Algorithm

We want an *efficient* algorithm which corresponds to an algorithm with runtime polynomial in parameters n and $\log M$, where M is the maximum ℓ_2-norm of the integer relation we are constrained to find[1]. We now give exposition to the PSLQ[2]

[1]We enforce polynomial in $\log M$ instead of M because representing M takes $\log M$ bits. Polynomial time typically refers to having runtime polynomial in the input size.

[2]The name comes from Partial Sum of Least Squares and the 'LQ' decomposition

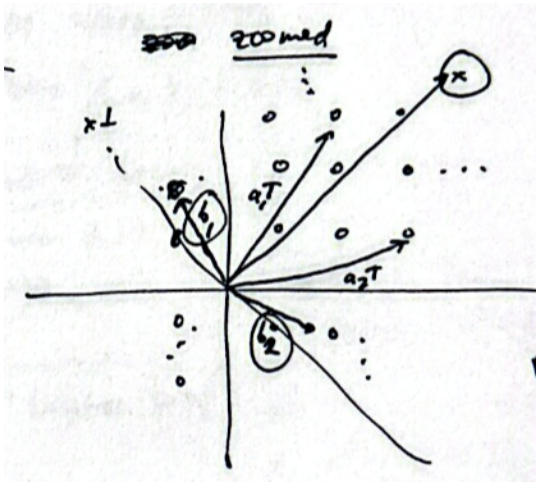

FIGURE 1. The above image shows the geometric visual for the $n = 2$ dimensional case. The span of $\mathbf{x}$ and span of $\mathbf{x}^{\perp}$ is shown, vector $\mathbf{x}$ is circled. The vectors $b_1, b_2 \in \mathbb{Z}^2$ are circled. Vectors a_1^T and a_2^T in lattice $\mathbb{Z}^2$ are also shown. Observe that for the a_i^T vectors close to span($\mathbf{x}$), we have the b_i vectors close to span($\mathbf{x}^{\perp}$).

algorithm [FBA99] [Bor02, Appendix B] for finding integer relations. The earlier algorithm HJLS [HsJLS89] achieves comparable worst-case performance, but (again) with worse numerical stability.

From a geometric perspective, we wish to find a a vector $\mathbf{t}$ with $||\mathbf{t}|| \leq M$ in the intersection of vectors in the $\mathbb{Z}^n$ integer lattice and vectors orthogonal to $\mathbf{x}$ (span of $\mathbf{x}^{\perp}$).

The PSLQ algorithm will iteratively construct a set of vectors that are close to $\mathbf{x}^{\perp}$. It is more straightforward to accomplish this indirectly by iteratively finding a basis for $\mathbb{Z}^n$, $a_1^T, \cdots, a_n^T$, that is close to span($\mathbf{x}$). We let $a_1^T, \cdots, a_n^T$ be the rows of an $n \times n$ matrix we denote A. We will want integer matrices A whose inverse are also integer matrices in order to use A^{-1} to find integer relations later. To this end, we will take A in the group $GL_n(\mathbb{Z})$. Equivalently, these are matrices over $\mathbb{Z}$ with determinant ± 1. Matrices of this form are called *unimodular*. If $B = A^{-1} \in \mathrm{GL}_n(\mathbb{Z})$ then the columns of B, $b_1, \cdots, b_n$ will have the property, using $A \cdot B = I$, that

$$a_i^T \cdot b_j = \delta_{ij}.$$

The δ_{ij} above is the Kronecker delta function which is 1 if $i = j$ and 0 otherwise. Thus, when the a_i are close to span($\mathbf{x}$), vectors b_j should be close to $\mathbf{x}^{\perp}$.

To measure the closeness of vectors a_i^T to $\mathbf{x}$, we control the squared magnitude of the projection onto $\mathbf{x}^\perp$. To conveniently deal with $\mathbf{x}^\perp$ we introduce matrix $H \in \mathbb{R}^{n\times(n-1)}$ whose $n-1$ columns form an orthonormal basis for $\mathbf{x}^\perp$. As $(h_i)_{1\le i\le n}$ form an orthonormal basis of the orthogonal subspace $\mathbf{x}^\perp$, we can express $\left|\left|proj_{\mathbf{x}^\perp} a_i^T\right|\right|$ as

$$\left|\left|proj_{\mathbf{x}^\perp} a_i^T\right|\right| = \sum_{j=1}^{n} \langle a_i, h_j\rangle \cdot h_j$$

Letting $H' := AH$ we get that $h'_{ij} = \langle a_i, h_j\rangle$, meaning the magnitude we wish to minimize is

$$\left|\left|proj_{\mathbf{x}^\perp} a_i^T\right|\right|^2 = \sum_{j=1}^{n} (h'_{ij})^2$$

Given an H we know of no such way to directly find a matrix $A \in \mathrm{GL}_n(\mathbb{Z})$ minimizing this quantity. Instead, we will iteratively left-multiply by a matrix $A_k \in \mathrm{GL}_n(\mathbb{Z})$ so that (inductively) $(A_k \cdots A_1 A_0)H$ will minimize the projection magnitudes. As $\mathrm{GL}_n(\mathbb{Z})$ is closed (since it forms a group) $A = A_k \cdots A_1 A_0$ will contain the desired $\mathbb{Z}^n$ lattice basis in the row vectors.

When can we ever guarantee that at least one of $a_i^T \in \mathbb{Z}^n$ will be in $\mathrm{span}(\mathbf{x})$? Assume that our the initial matrix $H \in \mathbb{R}^{n\times(n-1)}$ has the additional property of being lower trapezoidal: all the entries above the diagonal are zero. We can prove that a process that reduces the magnitude of the diagonals is not only sufficient, but gives lower bounds on the ℓ_2-norm of *any* integer relation of x. The following theorem is crucial.

THEOREM 1. *Let $A \in \mathrm{GL}_n(\mathbb{Z})$ and $\mathbf{x} \in \mathbb{R}^n$. Let $H \in \mathbb{R}^{n\times(n-1)}$ have columns forming an orthonormal basis for $\mathbf{x}^\perp$. If $H' = AH$ is lower trapezoidal with non-zero diagonals, then for* any *integer relation $\mathbf{t}$ of $\mathbf{x}$*

$$||\mathbf{t}|| \ge \frac{1}{\max_i |h'_{ii}|}$$

Proof. For any integer relation $\mathbf{t}$, $\langle \mathbf{t}, x\rangle = 0$ so $\mathbf{t} \in \mathbf{x}^\perp$. The matrix HH^T will project onto $\mathbf{x}^\perp$, so $HH^T\mathbf{t} = \mathbf{t}$. Thus,

$$A(HH^T\mathbf{t}) = H'H^T\mathbf{t} = A\mathbf{t} \tag{4.1}$$

Since we assume that $\mathbf{t}$ is non-trivial, $\mathbf{t} \neq 0$, and A is invertible we know $A\mathbf{t} \neq 0$. We can find a lattice basis vector a_j^T that is not orthogonal to $\mathbf{t}$. Let j be the vector with the smallest such $1 \le j \le n$. Then,

$$\forall 1 \le i < j, \quad a_i^T \cdot \mathbf{t} = 0 \tag{4.2}$$

We claim eq. (4.1) and eq. (4.2) together imply it must be the case that

$$\forall 1 \le i < j, (H^T\mathbf{t})_i = 0 \tag{4.3}$$

Assume $j > 1$, otherwise the claim trivially follows. $H'(H^T\mathbf{t})_1 = (A\mathbf{t})_1$ using eq. (4.1) so $H'(H^T\mathbf{t})_1 = 0$ by eq. (4.2). H' is lower trapezoidal so $H'_{11} \cdot (H^T\mathbf{t})_1 = 0$. The diagonal entries of H' are non-zero so $(H^T\mathbf{t})_1 = 0$. If $j > 2$, we can continue onto $i = 2$. By the same reasoning, and using $(H^T\mathbf{t})_1 = 0$, $H'_{22} \cdot (H^T\mathbf{t})_2 = 0$ so $(H^T\mathbf{t})_2 = 0$. We can continue this process inductively for all $1 \le i < j$. By this

claim, eq. (4.3) tells us $(A\mathbf{t})_j = a_j^T\mathbf{t}$ has a non-zero term contribution only from diagonal entry H'_{jj} scaled by $(H^T\mathbf{t})$. Therefore,

$$(h_j^T\mathbf{t}) \cdot (H')_{jj} = a_j^T\mathbf{t}$$

Vector $a_j^T\mathbf{t}$ is non-zero and in $\mathbb{Z}^n$ so

$$\left|(h_j^T\mathbf{t}) \cdot (H')_{jj}\right| = \left|a_j^T\mathbf{t}\right| \geq 1$$

As $h_j^T\mathbf{t}$ is the projection onto an orthonormal vector,

$$1 \leq \left|(h_j^T\mathbf{t}) \cdot (H')_{jj}\right| \leq \left|(h_j^T\mathbf{t})\right| |(H')_{jj}| \leq ||\mathbf{t}|| \, |(H')_{jj}|$$

as desired. □

The above theorem is crucial in proving the termination of the PSLQ algorithm and in showing reducing the diagonals of the above lower trapezoidal matrix via $\mathrm{GL}_n(\mathbb{Z})$ transformations is fundamentally related to integer relations. This also shows an additional important property of PSLQ: either an integer relation is found *or* one can provably show that an integer relation of magnitude at most M does not exist!

Initial Algorithm Attempt. Theorem 1 suggests a blueprint of a potential integer relation algorithm.

- Begin with the canonical basis of $\mathbb{Z}_n$ given by the rows of $A_0 := I \in \mathrm{GL}_n(\mathbb{Z})$, and construct an $H_0 := H_x \in \mathbb{R}^{n\times(n-1)}$ that is already lower trapezoidal (and whose column vectors spans $\mathbf{x}^\perp$) – more on how to construct such a matrix directly in section 4.1.
- Repeatedly decrease the diagonal entries by iteratively multiplying H by $A_k \in \mathrm{GL}_n(\mathbb{Z})$ on the left.

This does not quite get us there. The issue is that finding an unimodular integer matrix $A_k \in \mathrm{GL}_n(\mathbb{Z})$ reducing the diagonal entries is challenging and potentially impossible directly. Instead, we will decrease the diagonal entries by changing H iteratively as well. Observe in Theorem 1 we just assumed that H was lower trapezoidal (with non-zero diagonal entries), but we were free to choose H. We will perform two steps, described in section 4.3.1 and section 4.3.2, which will keep the desired properties of H invariant but modify it (after a known number of steps) to *reduce the diagonal entries*. Then, we will left-multiply by matrix $A_k \in \mathrm{GL}_n(\mathbb{Z})$ in order to keep the diagonal entries the same but ensure that the off-diagonals are small compared to the diagonal – this is described in section 4.2. We can show that this reduces $\left|\left|proj_{\mathbf{x}^\perp} a_i^T\right|\right|^2$ which is the geometric justification for this step. Discussion of the technical justification is in section 4.3.

4.1. **Partial Sum of Squares Matrix.** Knowing Theorem 1, we want to be able to efficiently construct an initial lower trapezoidal H with the desired properties given any $\mathbf{x}$. A common technique to accomplish this uses the sums of squares of the $\mathbf{x}$ entries

$$s_k = \sqrt{\sum_{i=k}^{n} x_i^2}$$

DEFINITION 1. *For vector* $\mathbf{x} \in \mathbb{R}^n$*, define* partial sum of squares (PSOS) matrix $H \in \mathbb{R}^{n\times(n-1)}$ *via*

$$H_{ij} = \begin{cases} 0 & 1 \le i < j \le n-1 \\ \frac{s_{i+1}}{s_i} & i = j \\ \frac{-x_i x_j}{s_j s_{j+1}} & 1 \le j < i \le n \end{cases}$$

We can prove that the columns indeed span $\mathbf{x}^\perp$.

LEMMA 1. *For PSOS matrix* H *from Definition 1,* $\mathbf{x}^T \cdot H = 0 \in \mathbb{R}^n$.

Proof. Let $\mathbf{h}_k \in \mathbb{R}^n$ be the kth column of H.
Then,

$$\langle \mathbf{h}_k, \mathbf{x} \rangle = \sum_{1\le t<k} H_{t,k} \cdot \mathbf{x}_t + H_{k,k} \cdot \mathbf{x}_k + \sum_{k+1\le t\le n} H_{t,k} \cdot \mathbf{x}_t$$

Recall $H_{t,k}$ for $t < k$ is zero, $H_{k,k} = \frac{s_{k+1}}{s_k}$, and $H_{t,k} = \frac{-\mathbf{x}_t \cdot x_k}{s_k s_{k+1}}$ for $t \ge k+1$. Therefore, the above simplifies to

$$\begin{aligned} \langle c, \mathbf{x} \rangle &= \frac{s_{k+1}}{s_k} \cdot \mathbf{x}_k + \sum_{k+1\le t\le n} \frac{-\mathbf{x}_t \cdot \mathbf{x}_k}{s_k s_{k+1}} \cdot \mathbf{x}_k = \frac{s_{k+1}}{s_k} \cdot \mathbf{x}_k + \sum_{k+1\le t\le n} \frac{-\mathbf{x}_t^2 \cdot \mathbf{x}_k}{s_k s_{k+1}} \\ &= \frac{s_{k+1}}{s_k} \cdot \mathbf{x}_k + \frac{\mathbf{x}_k}{s_k s_{k+1}} \cdot \sum_{k+1\le t\le n} -\mathbf{x}_t^2 = \frac{s_{k+1}}{s_k} \cdot \mathbf{x}_k - \frac{\mathbf{x}_k}{s_k s_{k+1}} \cdot s_{k+1}^2 = 0 \end{aligned}$$

□

It can also be quickly checked that all the columns of H will be orthogonal to each other and have magnitude 1. Therefore, the columns will form an orthonormal basis for the span of $\mathbf{x}^\perp$.

4.2. **Hermite Reduction.** Let H be a $n \times (n-1)$ lower trapezoidal matrix with non-zero diagonals whose columns are linearly independent. It would be convenient for AH, where $A \in \mathrm{GL}_n(\mathbb{Z})$, to be diagonal with the same diagonals as H. Minimizing the diagonals would directly make $\left|\left|proj_{\mathbf{x}^\perp} a_i^T\right|\right|^2$ smaller:

$$\left|\left|proj_{\mathbf{x}^\perp} a_i^T\right|\right|^2 = (AB)_{ii}^2$$

For a matrix A' in $\mathbb{R}^{n\times n}$ instead, we can construct a matrix with the above property directly using the following recursive construction.

$$A'_{ij} = \begin{cases} 0 & 1 \le i < j \le n \\ 1 & i = j \\ -\frac{1}{h_{jj}} \cdot \left(\sum_{k=j+1}^{i} A'_{ik} \cdot \frac{h_{kj}}{h_{jj}} \right) & \end{cases}$$

To construct this sequentially, iterate across i from 2 to n and then across j from $i-1$ to 1 in that order. It can be verified that A' does have the desired properties. Nevertheless, for our purposes we want A' to be a matrix in $GL_n(\mathbb{Z})$. To this end, we can simply round all the entries to the nearest integer. Define

$$A_{ij} = \lfloor A'_{ij} + 1/2 \rfloor$$

REMARK 1. *For* $x \in \mathbb{R}$ *the function* $\lfloor x + 1/2 \rfloor$ *rounds* x *to the nearest integer.*

This matrix can be shown to be invertible by routine calculation[3]. With rounding, we keep the diagonal the same after left multiplication, but will lose that the new matrix will be diagonal. However, $A'H$ will still maintain the important property that small diagonals imply small projection norm via the following lemma.

LEMMA 2. [FBA99, Lemma 4] *Let $H \in \mathbb{R}^{n\times(n-1)}$ be a lower trapezoidal matrix with non-zero diagonals. Let $A \in \mathrm{GL}_n(\mathbb{Z})$ be as above. Define $H' = AH$. Then, for all $i > j$,*

$$\left|h'_{ij}\right| \leq \frac{1}{2}\left|h'_{jj}\right|$$

We omit the proof of the above lemma, but it follows from the fact that non-rounded version of A, A', will exactly diagonalize H and using that $\left|a'_{ij} - a_{ij}\right| \leq 1/2$ (this is where the $1/2$ factor comes in). By multiplying A' with H and using the above facts, one can quickly derive Lemma 2.
The lemma implies the projection reduction claim:

$$\left|\left|proj_{\mathbf{x}^\perp} a_i^T\right|\right|^2 = \sum_{j=1}^{n}(h'_{ij})^2 = \sum_{j=1}^{n-1}(h'_{ij})^2 + (h_{ii'})^2 \leq \sum_{j=1}^{i-1}(h'_{jj}/2)^2 + (h'_{ii})^2 = \sum_{j=1}^{n}(h'_{jj})^2$$

Minimizing the ℓ_2 norm of the diagonal minimizes the norm of the projection. Matrix A is called the "reducing matrix". The matrix $H' := AH$ is the "reduced matrix".

4.3. **Diagonal Reduction.** We will do the following two steps in sequence to reduce the ℓ_1-norm (and hence the ℓ_2-norm) of the diagonal entries.

4.3.1. *Bergman Swap.* Let B be a $n \times m$ matrix, and $\gamma > 2/\sqrt{3}$ be a free parameter. We will essentially swap two rows if their corresponding entries in the diagonals are not properly ordered, even accounting for some slack dependent on γ. We can define $r := r_{B,\gamma}$ to be the j such that $\gamma^j \cdot |B_{jj}|$ is maximized. The Bergman swap will swap rows r and $r+1$. Observe that $\gamma^r \cdot |B_{rr}| \geq \gamma^{r+1} \cdot |B_{r+1,r+1}|$, or equivalently $|B_{rr}| \geq \gamma\,|B_{r+1,r+1}|$.

4.3.2. *LQ^T update.* After performing a Bergman swap with $r < n-1$ or left-multiplication by A to get $n \times m$ matrix B', we potentially lose the invariant that B' is lower trapezoidal. In order to maintain it, we will perform an LQ^T decomposition on matrix B'.

The LQ^T decomposition of B factors it into a lower trapezoidal matrix L and an orthogonal matrix Q^T. First, we recall the Gram–Schmidt orthonormalization process. Given a matrix B', we can output an upper trapezoidal matrix R and orthogonal matrix Q such that $B' = QR$. These matrices simply compactly express the Gram-Schmidt process which repeatedly subtracts from current vector $\mathbf{v}_i$ the projection of $\mathbf{v}_i$ onto the orthogonal set of previous $i-1$ vectors. Given RQ, we take $L \leftarrow R^T$ so that $B' = LQ^T$.

Given this decomposition, we can multiply the matrix B' on the right by Q to get a lower trapezoidal matrix L. In the PSLQ algorithm, B will be of the form AH where H's columns form an orthonormal basis. Right-multiplying by Q will

[3]For this calculation, first express the above matrix as a iterative transformation on the vectors in the basis formed by A and show this keeps linear independence assuming the necessary conditions about H.

give us an $H_1 := HQ$ that is still orthonormal, as desired.

It is possible to equivalently express the above two transformations more succinctly. By noticing only 2 rows are swapped, and carefully tracking the LQ^T update to see only the 2×2 sub-matrix from rows $r, r+1$ and columns $r, r+1$ changes, we can analytically express the update directly in the following way. Restrict to this sub-matrix. Before any updates, this is

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

Afterwards, this will be

$$\begin{pmatrix} \alpha' & 0 \\ \beta' & \theta' \end{pmatrix}$$

where

$$\begin{aligned} \alpha' &= \sqrt{\beta^2 + \theta^2} \\ \beta' &= \beta \cdot (\alpha/\alpha') \\ \theta' &= \theta \cdot (\alpha/\alpha') \end{aligned}$$

It is also possible to equivalently *define* this as the update, for r following the maximality condition from section 4.3.1[4]. When does this reduce the diagonal? We will first look at α. Recall that by the condition imposed on r, the value $\theta \leq \alpha/\gamma$. Additionally, if matrix B has been size reduced by the step in section 4.2 then $\beta \leq \alpha/2$. Therefore,

$$|\alpha'| \leq \sqrt{\alpha^2/4 + \theta^2} \leq \sqrt{\alpha^2/4 + \alpha^2/\gamma^2} = |\alpha| \cdot \sqrt{1/4 + 1/\gamma^2}.$$

As long as $\gamma^2 > 4/3$, or $\gamma > 2/\sqrt{3}$, this will make α' smaller in magnitude than α. Unfortunately, β' and α' will potentially increase in magnitude. However, the Bergman swap ensures that large entries are pushed towards the bottom-right corner of the matrix. If $r = n - 1$, this diagonal entry will decrease by a factor of 2. This follows since the size reduction condition will imply $\beta \leq \alpha/2$ so one row swap will lead to $|\alpha'| \leq |\alpha|/2$ (LQ is not performed since trapezoidal condition is kept). After sufficiently many steps, roughly speaking, these large diagonal entries will be reduced so the ℓ_1 norm of the entries is reduced. Proving that only a manageable (polynomial) number of steps are needed to reduce the diagonal entries requires examining a more sophisticated function of the diagonal entries and prove it decreases by a constant factor each step. This occurs in the proof of Theorem 3, which we omit.

4.4. **Full Algorithm.** We are now ready for the full algorithm. The initialization is the following.

(1) Begin with the canonical basis of $\mathbb{Z}_n$ given by the rows of $A_0 := I \in \mathrm{GL}_n(\mathbb{Z})$ and $H_0 := H_x$ from Definition 1.
(2) Perform a size reduction, from section 4.2, on $H' = A_0 H_0$. Still denote this matrix H'.

Then, we perform many iterations of the following, until a termination condition $H'_{n,n-1} = 0$ (which we explain soon) is satisfied.

[4]In fact, this replacement is the one used to yield a more numerically stable algorithm.

(3) Perform on H' both the Bergman swap and LQ^T steps from section 4.3 in order. Record the value of r. Let Q_k be the orthogonal matrix from the decomposition, $Q_k = I$ if no decomposition is necessary ($r = n - 1$).
(4) Perform on H' the size reduction from section 4.2. Let A_k be the reducing matrix, and $A \leftarrow A_k \cdot A$.

When the termination condition is satisfied, return the $(n-1)$-th column of A^{-1} which will be an integer relation for $\mathbf{x}$ (explained in Theorem 2).

We now analyze why this gives the desired integer relation algorithm. The H' matrix after k steps will be

$$H'^{(k)} = A^{(k)} H Q^{(k)},$$

where $A^{(k)} = \prod_{i=k}^{1} A_i$ is the product of all the reducing matrices $A_i \in \mathrm{GL}_n(\mathbb{Z})$, and $Q = \prod_{i=1}^{k} Q_i$ is the product of all orthogonal matrices Q_i.

The *termination condition* is that $H'_{n,n-1} = 0$. Using Theorem 1, we can prove this condition *must* be reached if an integer relation of magnitude at most M exists. If we keep reducing the diagonal entries, then we will prove ever increasing lower bounds on the magnitude of *any* integer relation for $\mathbf{x}$. Therefore, this cannot occur indefinitely if one exists with magnitude at most M. We cannot continue further when we cannot keep doing the swap and LQ step without making the diagonal entries non-zero (and incurring a division by zero error). In particular, this will imply that $H'_{k,k-1} = 0$ as the swap leads to a diagonal being zero. It must be that $k = n$. If we assume the diagonals are non-zero in the previous step, then a diagonal reduction from section 4.3 for $k < n$ will construct non-zero diagonals α', θ' by inspection. Therefore, without loss of generality, we have $H'_{n,n-1} = 0$ when the algorithm cannot continue further.

When the termination condition occurs, we will use the inverse of $A^{(k)} \in \mathrm{GL}_n(\mathbb{Z})$ to recover an integer relation in the following way.

THEOREM 2. *Let $H'^{(k)} = A^{(k)} H Q^{(k)} \in \mathbb{R}^{n \times (n-1)}$ be from the PSLQ algorithm, and let $A := A^{(k)}$. If $H'^{(k)}_{n,n-1}$ is zero, then the $(n-1)$th column of $B := A^{-1}$ is an integer relation $\mathbf{t}$ of $\mathbf{x}$.*

Proof. Assume that iteration $k+1$ is the first time when any entry of the diagonal was zero. Then, the entry $H'^{(k)}_{n-1,n-1}$ is non-zero and $H'^{(k)}_{n,n-1}$ is zero. Using Lemma 1 we know $\mathbf{x}^T H = 0$ (H was defined to have columns spanning $\mathbf{x}^\perp$), so

$$(\mathbf{x}^T A^{-1})(AH) = 0 \Leftrightarrow (\mathbf{x}^T B)(A^{(k)} H Q^{(k)}) = (\mathbf{x}^T B) H'^{(k)} = 0.$$

Thus, the only non-zero contributing term to $\left((\mathbf{x}^T B) \cdot H'^{(k)}\right)_{n-1} = 0$ is

$$(\mathbf{x}^T b_{n-1}) \cdot H'^{(k)}_{n-1,n-1},$$

using that $H'^{(k)}$ is lower trapezoidal and $H'^{(k)}_{n,n-1} = 0$. Finally, using $H'^{(k)}_{n-1,n-1}$ is non-zero, this implies $\mathbf{x}^T b_{n-1} = 0$. That is, the $(n-1)$th column of B (in $\mathbb{Z}^n$) is an integer relation. □

Theorem 2 is the second crucial theorem in the PSLQ algorithm since it shows that once the termination condition is reached (which *must* be reached using Theorem 1 by the prior discussion) we can obtain an integer relation.

It can be shown that the PSLQ algorithm will have runtime polynomial in the dimension n and in $\log M$. This follows from a careful analysis which we omit for brevity. We give a rough proof outline below.

THEOREM 3. [FBA99, Theorem 2] *The PSLQ algorithm will terminate with an integer relation of* $\mathbf{x}$*, or prove such a relation with magnitude at most* M *does not exist, in number of steps at most*

$$\binom{n}{2} \cdot \frac{(n-1)\log\gamma + \log M}{\log\tau}$$

where $\tau = 1/\sqrt{1/4 + 1/\gamma^2}$.

Proof. Proof Sketch.
One can show a particular quantity relating to the absolute value of the diagonal entries will decay exponentially. Thus, each time that $H'_{n,n-1}$ is non-zero (so the diagonals entries are non-zero) we can prove an exponentially improving lower bound. Thus, only steps proportional to $\log M$ will be required. The quantity is

$$\Pi(k) := \prod_{i=1}^{k-1} \left[\min(\gamma^{n-1}M, \frac{1}{\left|h'^{(k)}_{i,i}\right|}) \right]^{n-i}$$

For the theorem [FBA99] proves two important lemmas: $\Pi(k)$ is bounded absolutely by $(\gamma^{n-1}M)^{\binom{n}{2}}$, and Π increases by a factor of at least τ when going from k to $k+1$. The first fact essentially follows from bounding $\min(\gamma^{n-1}M, 1/\left|h'^{(k)}_{i,i}\right|)$ by $\gamma^{n-1}M$. This second fact follows from carefully tracking the effect of the diagonal reduction from section 4.2 on $\Pi(k)$. These two lemmas clearly imply the quantitative claims of the theorem. □

The above theorem is extremely useful since it implies that the PSLQ algorithm can be used in practice. We note a subtle point in the guarantees given by the algorithm. If the algorithm does not find an integer relation, then no integer relation of magnitude at most M exists. However, if an integer relation is returned we *do not* necessarily get the smallest magnitude integer relation – or even one with magnitude at most M. The algorithm will return *some* integer relation (it will turn out the magnitude of this relation is at most M multiplied by a factor exponential in n).

REMARK 2. *Since we just used elementary arithmetic operations and simple linear algebra in the PSLQ algorithm above, it can also be shown it generalizes to the case where* $\mathbf{x}$ *is over the quaternionic number system! The algorithm and the analysis remain almost exactly the same (except for slightly weaker bounds), but more careful analysis of the proofs is required.*

REFERENCES

[BBBP97] D. H. Bailey, J. M. Borwein, P. B. Borwein, and S. Plouffe. The quest for pi. *Math. Intelligencer*, 19(1):50–57, 1997.

[BJMadH88] László Babai, Bettina Just, and Friedhelm Meyer auf der Heide. On the limits of computations with the floor function. *Inform. and Comput.*, 78(2):99–107, 1988.

[BLe00] Jonathan M. Borwein and Petr Lisonˇ ek. Applications of integer relation algorithms. volume 217, pages 65–82. 2000. Formal power series and algebraic combinatorics (Vienna, 1997).

[Bor02] Peter Borwein. *Computational excursions in analysis and number theory*, volume 10 of *CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC*. Springer-Verlag, New York, 2002.

[FBA99] Helaman R. P. Ferguson, David H. Bailey, and Steve Arno. Analysis of PSLQ, an integer relation finding algorithm. *Math. Comp.*, 68(225):351–369, 1999.

[FF79] H. R. P. Ferguson and R. W. Forcade. Generalization of the Euclidean algorithm for real numbers to all dimensions higher than two. *Bull. Amer. Math. Soc. (N.S.)*, 1(6):912–914, 1979.

[HsJLS89] J. Hå stad, B. Just, J. C. Lagarias, and C.-P. Schnorr. Polynomial time algorithms for finding integer relations among real numbers. *SIAM J. Comput.*, 18(5):859–881, 1989.

[Sch80] Wolfgang M. Schmidt. *Diophantine approximation*, volume 785 of *Lecture Notes in Mathematics*. Springer, Berlin, 1980.

Massachusetts Institute of Technology
Email address: adelmo@mit.edu