Integer Relations among Real Numbers

Daria Romanova

June 11, 2007

Abstract

A lot of interesting and important results in various areas of mathematics were obtained with the help of the algorithms for finding integer relations among real numbers.

We will consider two mostly used types of such algorithms and present a couple of their applications.

1 Introduction: Integer Relations

Let X be a mathematical expression, that can be approximated numerically. (For example a definite integral.) Suppose we know, that X is *rational*.

Example. If $X \approx 2.33333333333333333\ldots$ then we make a conclusion that $X = 7/3$.

Example. But what if $X \approx 0.14127423822714681440443 21\ldots$?

Let us expand X into continuous fraction:

$$0.14127423822714681440\ldots \approx \cfrac{1}{7+\cfrac{1}{12+\cfrac{1}{1+\cfrac{1}{2+\cfrac{1}{1+\cfrac{1}{24987506246876561,719}}}}}} \Rightarrow$$

$$X = \cfrac{1}{7+\cfrac{1}{12+\cfrac{1}{1+\cfrac{1}{2+\cfrac{1}{1}}}}} = \frac{51}{361}.$$ (Actually the period of $\frac{51}{361}$ is 342.)

Now suppose we do not know if X is rational. But we do know that it is a *quadratic irrationality*. (That is a root of an equation $ax^2 + bx + c = 0$ with a, b, c rational.)

Theorem 1 (Lagrange)**.** *X is a quadratic irrationality* $\Leftrightarrow$ *its continuos fraction is periodic.*

Example. $\sqrt{3} = 1+(\sqrt{3}-1) = 1+\dfrac{2}{\sqrt{3}+1} = 1+\dfrac{1}{\frac{\sqrt{3}+1}{2}} = 1+\dfrac{1}{\frac{2+(\sqrt{3}-1)}{2}} =$
$1+\dfrac{1}{1+\frac{\sqrt{3}-1}{2}} = 1+\dfrac{1}{1+\frac{2}{2(\sqrt{3}+1)}} = 1+\dfrac{1}{1+\frac{1}{\sqrt{3}+1}} = 1+\dfrac{1}{1+\frac{1}{2+(\sqrt{3}-1)}} =$
$1+\cfrac{1}{1+\cfrac{1}{2+\cfrac{1}{1+\cfrac{1}{2+\ldots}}}}.$

Let us generalize the problem.

Definition 2. α is an *algebraic number* if there exist $a_0, \ldots, a_n \in \mathbb{Z}$ such that $a_n\alpha^n + \ldots + a_1\alpha + a_0 = 0$ and $a_n \neq 0$. The *degree* of α is the smallest of such n.

Remark. α is algebraic of degree $\leq n \Leftrightarrow (1, \alpha, \alpha^2, \ldots, \alpha^n)$ posess an *integer relation* [see below].

Definition 3. An *integer relation* for n-tuple $(x_1, \ldots, x_n) \in \mathbb{R}^n$ is an n-tuple $0 \neq (a_1, \ldots, a_n) \in \mathbb{Z}^n$ such that $a_1x_1 + \ldots + a_nx_n = 0$.

The problem of finding an integer relation for two numbers (x_1, x_2) can be solved by applying the Euclidian algorithm to x_1, x_2, or, equivalently, by computing the continued fraction expansion of x_1/x_2.

The generalization for $n \geq 3$ was attempted by Euler, Jacobi, Minkowski, Perron, Bernstein, among others.

The best known and most used algorithms at the present time are either algorithms based on the lattice basis reduction algorithm by Lenstra, Lenstra, Jr. and Lovász (*LLL*) or *PSLQ* algorithm based on ideas of Ferguson and Forcade. (Both discovered in 1970-s – 1980-s.)

2 Algorithms for Finding Integer Relations

2.1 Preliminaries

Let us recall some definitions.

Let $\mathbb{R}^n$ be the n-dimensional real vector space ($n > 1$) with inner product: $\langle \mathbf{x}, \mathbf{y} \rangle = \sum\limits_{j=1}^{n} x_j y_j$. $\|\mathbf{y}\| = \sqrt{\langle \mathbf{y}, \mathbf{y} \rangle}$ is *the length* of $\mathbf{y} \in \mathbb{R}^n$. $\mathbf{x}$ and $\mathbf{y}$ are *orthogonal* if $\langle \mathbf{x}, \mathbf{y} \rangle = 0$.

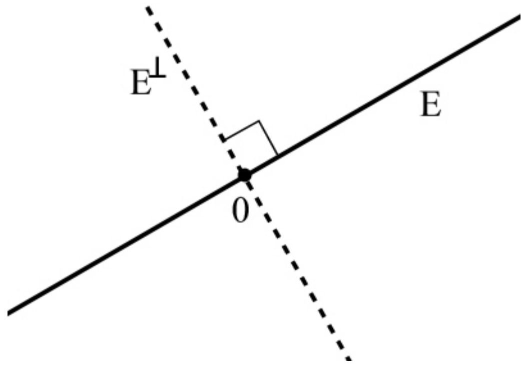

Figure 1: E and $E^\perp$

For a linear subspace $E \subset \mathbb{R}^n$ we denote by $E^\perp \subset \mathbb{R}^n$ the *orthogonal complement* of E (i.e., the subspace consisting of all vectors that are orthogonal to E).

If $\mathbf{b}_1, \ldots, \mathbf{b}_r \in \mathbb{R}^n$ then $[\mathbf{b}_1, \ldots, \mathbf{b}_r]$ will denote $n \times r$ matrix which has $\mathbf{b}_1, \ldots, \mathbf{b}_r$ as columns.

span$(\mathbf{b}_1, \ldots, \mathbf{b}_r)$ is the linear space, spanned on $\mathbf{b}_1, \ldots, \mathbf{b}_r$: span$(\mathbf{b}_1, \ldots, \mathbf{b}_r) = \left\{ \sum_{j=1}^{r} a_j \mathbf{b}_j \, | a_j \in \mathbb{R} \right\}$.

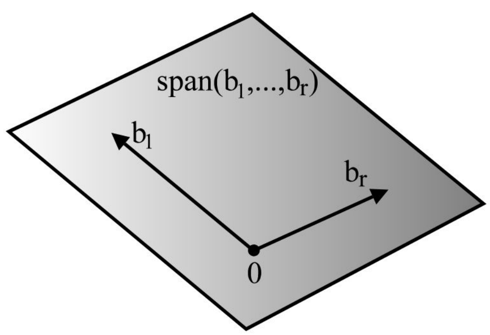

Figure 2: span$(\mathbf{b}_1, \ldots, \mathbf{b}_r)$

With $\mathbf{b}_0 = \mathbf{x},\ \mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^n$ we associate the orthogonal system $\mathbf{b}_0^*, \ldots, \mathbf{b}_n^*$ that are defined inductively:
$\mathbf{b}_0^* = \mathbf{x}$,
$\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=0}^{i-1} \frac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle}{\langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle} \mathbf{b}_j^*,\ i = 1, \ldots, n.$

This process is called *Gram-Schmidt orthogonalization*.

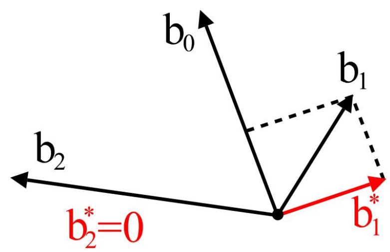

Figure 3: Gram-Schmidt orthogonalization

Remark. $\mathbf{b}_i^*$ is orthogonal to span$(\mathbf{b}_0^*, \ldots, \mathbf{b}_{i-1}^*)$=span$(\mathbf{b}_0, \ldots, \mathbf{b}_{i-1})$.

Definition 4. A *lattice* $L \subset \mathbb{R}^n$ is an additive closure of some linear independent $\mathbf{b}_1, \ldots, \mathbf{b}_r \in \mathbb{R}^n : L = \left\{ \sum_{i=1}^{r} m_i \mathbf{b}_i \,| m_i \in \mathbb{Z} \right\}$.

Such $\mathbf{b}_1, \ldots, \mathbf{b}_r$ are called the *basis* of L. Of course they are not defined uniquely.

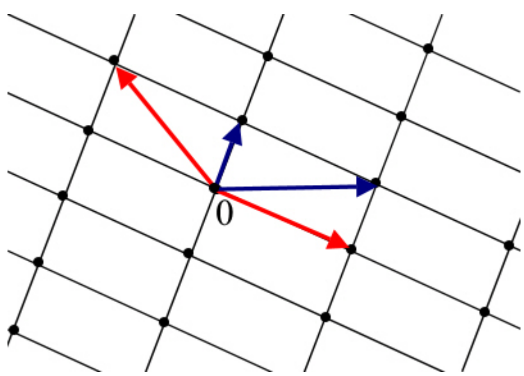

Figure 4: Lattice and its Bases

An important example: the lattice $L_{\mathbf{x}} \subset \mathbb{Z}^n$ of all integer relations for $\mathbf{x}$ together with $\mathbf{0} : L_{\mathbf{x}} := \{ \mathbf{m} \in \mathbb{Z}^n \,| \langle \mathbf{x}, \mathbf{m} \rangle = 0 \}$.

We will perform two types of elementary basis exchange operations on a current basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$ of a given lattice:

Exchange steps: interchange $\mathbf{b}_i$ and $\mathbf{b}_{i+1}$ for some i;

Size-reduction steps: replace $\mathbf{b}_i$ with $\mathbf{b}_i - p\mathbf{b}_j$ where $p \in \mathbb{Z}$ for some $1 \leq j < i$.

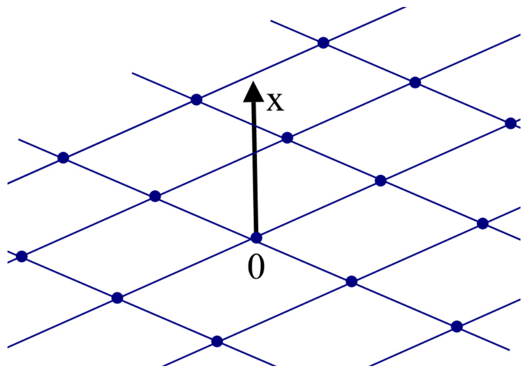

Figure 5: $L_{\mathbf{x}}$

With every basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$ there is the *dual basis* $\mathbf{c}_1, \ldots, \mathbf{c}_n$: $[\mathbf{c}_1, \ldots, \mathbf{c}_n]^T = [\mathbf{b}_1, \ldots, \mathbf{b}_n]^{-1} \Leftrightarrow [\mathbf{c}_1, \ldots, \mathbf{c}_n]^T [\mathbf{b}_1, \ldots, \mathbf{b}_n] = \mathrm{Id} \Leftrightarrow \langle \mathbf{b}_j, \mathbf{c}_k \rangle = \delta_{jk} = \begin{cases} 0, & j \neq k \\ 1, & j = k \end{cases}$.

Remark. If $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{Z}^n$ and $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ is unimodular ($\det B = \pm 1$) then $\mathbf{c}_1, \ldots, \mathbf{c}_n \in \mathbb{Z}^n$.

2.2 LLL-based Algorithms: HJLS

HJLS is a variation of LLL-algorithm by Hastad, Just, Lagarias and Schnorr. (In [HJLS1989] it is called "Small Integer Relation Algorithm.")

We will use the following arithmetic operations on real numbers at unit cost: addition, subtraction, multiplication, division, comparison, the nearest integer($\lceil \rfloor$).

Let's introduce some notation. μ_{ij} will denote the Gram-Schmidt quantities $\frac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle}{\langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle}$. We define $\lambda(\mathbf{x})$ as the length of the shortest integer relation for $\mathbf{x}$. If there are no relations then we let $\lambda(\mathbf{x}) = \infty$.

The algorithm

1. Input: $\mathbf{x} \in \mathbb{R}^n, k \in \mathbb{N}$.
2. Initiation: $\mathbf{b}_0 := \mathbf{x}$; $\mathbf{b}_1, \ldots, \mathbf{b}_n :=$ standard basis of $\mathbb{Z}^n$.
Compute μ_{ij} and $B_i = \langle \mathbf{b}_i^*, \mathbf{b}_i^* \rangle$.
3. Termination test:
If $B_n \neq 0$ then an integer relation is found. Compute $[\mathbf{c}_1, \ldots, \mathbf{c}_n]^T = [\mathbf{b}_1, \ldots, \mathbf{b}_n]^{-1}$ and output the integer relation $\mathbf{c}_n$. Stop.
If $\sqrt{B_j} \leq 1/2^k$ for $1 \leq j \leq n$ then no small integer relation exist. Output "$\lambda(\mathbf{x}) \geq 2^k$" and stop.

4. Exchange step:
Choose from $1 \le i \le n$ that i that maximizes $2^i B_i$.
Size-reduce $\mathbf{b}_{i+1}$: $\mathbf{b}_{i+1} := \mathbf{b}_{i+1} - \lceil \mu_{i+1,i} \rfloor \mathbf{b}_i$.
Update $\mu_{i+1,j}$ for $j = 1, \dots, i$.
Exchange $\mathbf{b}_i$ and $\mathbf{b}_{i+1}$.
Update $B_\nu,\ \mu_{\nu j},\ \mu_{j\nu}$ for $\nu = i, i+1,\ 1 \le j \le n$. Go to (2).

Remark. We list here explicit formulae for step 4.
$\mathbf{b}_{i+1} := \mathbf{b}_{i+1} - \lceil \mu_{i+1,i} \rfloor \mathbf{b}_i \Rightarrow \mu_{i+1,j} := \mu_{i+1,j} - \lceil \mu_{i+1,i} \rfloor \mu_{ij}$ for $j = 1, \dots, i$.
Updating $B_\nu,\ \mu_{\nu,j},\ \mu_{j,\nu}$ for $\nu = i, i+1,\ 1 \le j \le n$ in the case $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$:
$\mu := \mu_{i+1,1}$; $B := B_{i+1} + \mu^2 B_i$.
If $B \ne 0$ then $B_{i+1} := B_i B_{i+1}/B$, $\mu_{i+1,i} := \mu B_i / B$;
else $B_{i+1} := B_i$, $\mu_{i+1,i} := 0$.
$B_i := B$.
$\begin{pmatrix} \mu_{ij} \\ \mu_{i+1,j} \end{pmatrix} := \begin{pmatrix} \mu_{i+1,j} \\ \mu_{ij} \end{pmatrix}$ for $j = 1, \dots, i-1$.
$\begin{pmatrix} \mu_{ji} \\ \mu_{j,i+1} \end{pmatrix} := \begin{pmatrix} 1 & \mu_{i+1,i} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & -\mu \end{pmatrix} \begin{pmatrix} \mu_{ji} \\ \mu_{j,i+1} \end{pmatrix}$ for $j = i+2, \dots, n$.

Remark. The matrix $[\mathbf{c}_1, \dots, \mathbf{c}_n]$ can be computed incrementally.
Initially $[\mathbf{c}_1, \dots, \mathbf{c}_n] = \mathrm{Id}_n$.
$\mathbf{b}_{i+1} := \mathbf{b}_{i+1} - \lceil \mu_{i+1,i} \rfloor \mathbf{b}_i \Rightarrow \mathbf{c}_i := \mathbf{c}_i + \lceil \mu_{i+1,i} \rfloor \mathbf{c}_{i+1}$.
$\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1} \Rightarrow \mathbf{c}_i \leftrightarrow \mathbf{c}_{i+1}$.

Theorem 5. *The output $\mathbf{c}_n$ is an integer relation for $\boldsymbol{x}$.*
For every basis $\boldsymbol{b}_1, \dots, \boldsymbol{b}_n$ of $\mathbb{Z}^n$ $\lambda(\boldsymbol{x}) \ge 1 / \max\limits_{1 \le j \le n} \left\| \boldsymbol{b}_j^* \right\|$. So the algorithm claims "$\lambda(\boldsymbol{x}) \ge 2^k$" correctly.
The output $\boldsymbol{c}_n$ satisfies $\left\| \boldsymbol{c}_n \right\|^2 \le 2^{n-2} \min \left\{ \lambda(\boldsymbol{x})^2, 2^{2k} \right\}$.
The algorithm halts after at most $O(n^3(k+n))$ arithmetic steps on real numbers.

Proof. (Only two first statements.)
Since $\mathbf{b}_n^* \ne \mathbf{0}$ there exists i such that $\mathbf{b}_i^* = \mathbf{0}$. Then $\mathbf{0} = \mathbf{b}_i^* = \mathbf{b}_i - \sum\limits_{j=0}^{i-1} \frac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle}{\langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle} \mathbf{b}_j^*$. So there exist a_j such that $\sum\limits_{j=0}^{i} a_j \mathbf{b}_j = \mathbf{0}$. But $\mathbf{b}_1, \dots, \mathbf{b}_i$ are linearly independent, so $a_0 \ne 0$ and $\mathbf{x} = \mathbf{b}_0 = \sum\limits_{j=1}^{i} \frac{a_j}{a_0} \mathbf{b}_j$. Since $\langle \mathbf{b}_j, \mathbf{c}_k \rangle = 0$ for $k > j$ we have $\langle \mathbf{x}, \mathbf{c}_k \rangle = 0$ for $k > i$, in particular $\langle \mathbf{x}, \mathbf{c}_n \rangle = 0$.

Let $\mathbf{m}$ be any integer relation for $\mathbf{x}$. Since $\mathbf{m} \in (\mathbf{x}\mathbb{R})^\perp = \mathrm{span}(\mathbf{b}_1^*, \dots, \mathbf{b}_n^*)$ there exists i such that $\langle \mathbf{m}, \mathbf{b}_i^* \rangle \ne 0$. For the smallest such i we have $\langle \mathbf{m}, \mathbf{b}_i^* \rangle = \left\langle \mathbf{m}, \mathbf{b}_i - \sum\limits_{j=0}^{i-1} \mu_{ij} \mathbf{b}_j^* \right\rangle = \langle \mathbf{m}, \mathbf{b}_i \rangle - \sum\limits_{j=0}^{i-1} \mu_{ij} \left\langle \mathbf{m}, \mathbf{b}_j^* \right\rangle = \langle \mathbf{m}, \mathbf{b}_i \rangle \in \mathbb{Z}$, and hence $|\langle \mathbf{m}, \mathbf{b}_i^* \rangle| \ge 1$. But $|\langle \mathbf{m}, \mathbf{b}_i^* \rangle| \le \|\mathbf{m}\| \left\| \mathbf{b}_i^* \right\|$. So $\|\mathbf{m}\| \ge \frac{1}{\left\| \mathbf{b}_i^* \right\|}$. □

For details see [LLL1982] and [HJLS1989].

2.3 PSLQ

The name "PSLQ" comes from *partial sums* of squares and LQ (lower-diagonal — orthogonal) matrix decomposition.

We will use the same model of computation as with previous algorithm.

Let $\mathbf{x} = (x_1, \ldots, x_n), \|\mathbf{x}\| = 1, x_j \neq 0$.

Definition 6. Let for $1 \leq j \leq n$ $s_j^2 := \sum_{k=j}^{n} x_k^2$.

Definition 7. Let $H_{\mathbf{x}} = (h_{i,j})$ be $n \times (n-1)$ lower-trapezoidal matrix defined by:

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

The Algorithm

1. Input: $\mathbf{x} \in \mathbb{R}^n; \gamma \geq \sqrt{4/3}$.
2. Initiation: $\mathbf{s} := (s_1, \ldots, s_n);\ \mathbf{y} := \mathbf{x}/s_1;\ H := H_{\mathbf{x}};\ B := \mathrm{Id}_n$.

Reduce H:

for $i := 2$ **to** n
  for $j := i-1$ **to** 1 **step** -1
    $t := \lceil h_{ij}/h_{jj} \rfloor$
    $y_j := y_j + t y_i$
    for $k := 1$ **to** j
      $h_{ik} := h_{ik} - t h_{jk}$
    endfor
    for $k := 1$ **to** n
      $b_{kj} := b_{kj} + t b_{ki}$
    endfor
  endfor
endfor

3. Exchange step:

Choose r that maximizes $\gamma^r |h_{rr}|$.

Exchange $y_r \leftrightarrow y_{r+1}$, corresponding rows of H and corresponding columns of B.

4. Corner:

$\delta := \sqrt{h_{rr}^2 + h_{r,r+1}^2};\ \alpha := h_{rr}/\delta;\ \beta := h_{r,r+1}/\delta$.

if $r \leq n-2$ **then**
  for $i := r$ **to** n
    $h_0 := h_{ir};\ h_1 := h_{i,r+1};$
    $h_{ir} := \alpha h_0 + \beta h_1;\ h_{i,r+1} := -\beta h_0 + \alpha h_1$
  endfor

endif

5. Reduce H.

6. Norm bound: Compute $M := 1/\max_{1 \le j \le n} h_{jj}$. Then $\lambda(\mathbf{x}) \ge M$.

7. Termination: Goto (3) unless $y_j = 0$ for some $1 \le j \le n$ or $h_{ii} = 0$ for some $1 \le i \le n-1$.

Theorem 8. *The integer relation* $\mathbf{m}$ *for* $\boldsymbol{x}$ *appears as one of the columns of* B.

The following holds at each step: $\lambda(\boldsymbol{x}) \ge 1/\max_{1 \le j \le n} h_{jj}$.

The output satisfies $\|\mathbf{m}\| \le \gamma^{n-2}\lambda(\boldsymbol{x})$.

The algorithm halts after at most $O(n^4 + n^3 \log \lambda(\boldsymbol{x}))$ *arithmetic steps on real numbers.*

For details see [FBA1999].

3 Usage

It is important to note that since a computer can operate only with rational numbers, the discovery of an integer relation by a computer does not constitute a proof. However, in many cases the numerically discovered relations received afterwards rigorous mathematical proofs. Moreover, many complicated relations would probably never never be found without the help of computer.

It should be also emphasized that for all integer relation finding algorithms a very high precision arithmetic must be used. As a rule of thumb if $\mathbf{x}$ has n entries and D is the maximal number of digits in the relation we hope to find then we should work with nD digits precision.

LLL vs PSLQ

LLL-based algorithms are available in almost any computer algebra system (Maple, Mathematica). PSLQ implementation are less directly available.

PSLQ is more stable, because it uses a stable matrix reduction procedure. Unfortunately, HJLS is not stable. The cause of this instability is not known, but is believed to derive from its reliance on Gram-Schmidt orthonormalization, which is known to be numerically unstable.

Let us compare the two algorithms on a simple example.

Example. Consider $\mathbf{x} = (11, 27, 31)$.

PSLQ with $\gamma = \sqrt{2}$ for successive iterations $N = 0, 1, 2, 3, 4$ yields the five matrices:

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

$$\begin{pmatrix} \boxed{\begin{matrix} -1 \\ 5 \\ -4 \end{matrix}} & \boxed{\begin{matrix} -8 \\ 9 \\ -5 \end{matrix}} & \begin{matrix} -2 \\ 2 \\ -1 \end{matrix} \end{pmatrix}.$$

It found two relations (the outlined columns).

HJLS for successive iterations $N = 0, 1, 2, 3, 4, 5, 6$ yields the seven matrices:

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

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

It found one relation.

4 Applications

4.1 "BBP" Formula for Pi

Perhaps one of the best known applications of PSLQ is the 1995 discovery, by means of PSLQ computation, of the "BBP" (Bailey, Borwein, Plouffe) formula for π:

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

This formula permits one to compute directly hexademical digits of π without computing previous ones.

It was found by applying PSLQ to $(X_1, \ldots, X_n, \pi)$ where

$$X_j = \sum_{k=0}^{\infty} \frac{1}{16^k (8k+j)}.$$

4.2 Bifurcation Points in Chaos Theory

The chaotic iteration $x_{n+1} = rx_n(1 - x_n)$ ("logistic iteration") has been studied since the beginning of the chaos theory.

For $1 < r < B_1 = 3$ iterates converge to some nonzero point. If $B_1 < r < B_2 = 1 + \sqrt{6} = 3.449489\ldots$ then we have two distinct limit points. When $B_2 < r < B_3$ iterates choose between four distinct limit points. For $B_3 < r < B_4$ we have eight distinct limit points. And so on.

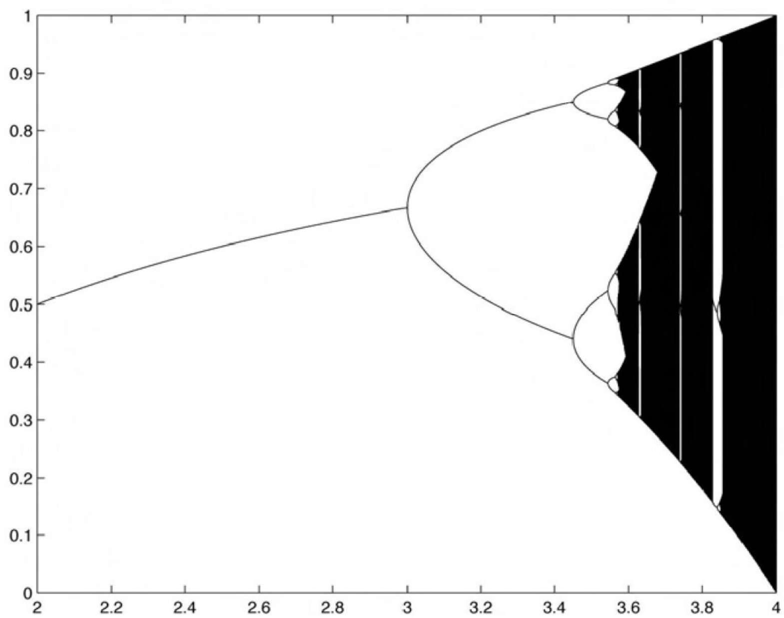

Figure 6: Bifurcation in Chaotic Iteration

All the B_j are algebraic numbers, so one can try to find their minimal polynomials, using integer relations founding algorithms.

Using PSLQ with $n = 13$ we get that B_3 satisfies:
$r^{12} - 12r^{11} + 48r^{10} - 40r^9 - 193r^8 + 392r^7 + 44r^6 + 8r^5 - 977r^4 - 604r^3 + 2108r^2 + 4913 = 0$.

The much more difficult problem for finding B_4 was studied in [BB2000].
It was conjectured that B_4 might satisfy a 240-degree polynomial, and, in addition, $\alpha = -B_4(B_4 - 2)$ might satisfy a 120-degree polynomial. Then an advanced PSLQ implementation was employed, and a relation with coefficients descending from 257^{30} to 1 was found.
4 year later the result was confirmed in large symbolic computation in [KK2004].

We refer to [BB2006] for more applications of integer relation finding algorithms.

References

[LLL1982] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász.

Factoring Polynomials with Rational Coefficients.

Math. Ann., Vol.261, 1982, pp.515-534.

[HJLS1989] J. Hastad, B. Just, J. C. Lagarias, and C. P. Schnorr.

Polynomial Time Algorithms for Finding Integer Relations among Real Numbers.

SIAM J. Comput., Vol.18, 1989, pp.859-881.

[FBA1999] H. R. P. Ferguson, D. H. Bailey, S. Arno.

Analysis of PSLQ, an Integer Finding Algorithm.

Mathematics of Computation, Vol.68, 1999, pp.351-369.

[BB2000] D. H. Bailey and D. J. Broadhurst.

Parallel Integer Relation Detection: Techniques and Applications.

Mathematics of Computation, Vol.70, 2000, pp.1719-1736.

[KK2004] I. Kotsireas and K. Karamanos.

Exact Computation of the Bifurcation Point B4 of the Logistic Map and the Bailey-Broadhurst Conjectures.

Internat. J. Bifurcation and Chaos, Vol.14, 2004, pp.2417-2423.

[BB2006] D. H. Bailey, J. M. Borwein, N. J. Calkin, R. Girgensohn, D. R. Luke, and V. H. Moll.

"Integer Relation Detection": §2.2 in Experimental Mathematics in Action.

Natick, MA: A. K. Peters, pp.29-31, 2006.