中国科学 : 数学　2021 年　第 51 卷　第 1 期 : 3 ∼ 16
SCIENTIA SINICA Mathematica
综　述

零误差计算

献给张景中、杨路教授 85 华诞

冯勇[1,2], 陈经纬[1,2*]

1. 中国科学院重庆绿色智能技术研究院, 重庆 400714;
2. 自动推理与认知重庆市重点实验室, 重庆 400714
E-mail: yongfeng@cigit.ac.cn, chenjingwei@cigit.ac.cn

收稿日期: 2019-12-31; 接受日期: 2020-04-13; 网络出版日期: 2020-07-08; * 通信作者
国家自然科学基金 (批准号: 11671377, 11501540 和 11771421) 和中国科学院青年创新促进会资助项目

摘要 研究采用有误差的数值计算来获得无误差的准确值具有重要的理论价值和应用价值. 这种通过近似的数值方法获得准确结果的计算被称为零误差计算. 本文首先指出, 只有一致离散集合中的数才能够开展零误差计算, 即有非零隔离界的数集, 这也是 "数" 可以进行零误差计算的一个充要条件. 以此为基本出发点, 本文分析代数数零误差计算的最低理论精度, 该精度对应于恢复近似代数数的准确值时必要的误差控制条件, 但由于所采用恢复算法的局限性, 这一理论精度往往不能保证成功恢复出代数数的准确值. 为此, 本文给出采用 PSLQ (partial-sum-LQ-decomposition) 算法进行代数数零误差计算所需的精度控制条件, 与基于 LLL (Lenstra-Lenstra-Lovász) 算法相比, 该精度控制条件关于代数数次数的依赖程度由二次降为拟线性, 从而可降低相应算法的复杂度. 最后探讨零误差计算未来的发展趋势.

关键词 零误差计算　整数关系　误差控制　LLL 算法　PSLQ 算法

MSC (2010) 主题分类 11A05, 11Y16, 68W40

1 引言

无论计算控制多高, 近似计算的结果总是近似的, 它与准确结果之间始终有一个间隙. 能否跨越这一间隙, 实现近似值到准确值的飞跃? 零误差计算就是研究如何实现这种飞跃.

哪些实数适合零误差计算呢? 要回答这个问题, 需要引入一个概念: 一致离散集合. 所谓一致离散集合就是集合中的元素之间的隔离界是非零的, 即存在 $\delta > 0$ 使得对于集合中的任意两个元素 x 和 y, 要么 $x = y$, 要么 $d(x, y) \geqslant \delta$ (参见文献 [1]), 其中 $d(x, y)$ 表示 x 与 y 的距离, δ 称为这个一致离散集合的一个隔离界, 所有隔离界的上确界被称为最优隔离界.

可以进行零误差计算的实数首先要知道它属于实数空间中的某集合 Ω, 其次在这个集合 Ω 中, 这个数与其他数可通过距离的方式区别开来, 即这个集合中的相异元素之间要有非零的隔离界, 因而, 这

英文引用格式: Feng Y, Chen J W. On zero-error computation (in Chinese). Sci Sin Math, 2021, 51: 3–16, doi: 10.1360/SSM-2019-0336

个集合就是一个一致离散集合; 反之, 若事先知道需要计算的数属于某个一致离散集合, 其隔离界为 δ, 则当计算出来的近似值与准确值之间的误差小于 $\delta/2$ 时, 在该近似值的 $\delta/2$- 邻域内仅有唯一的一个该一致离散集合中的元素, 因此, 可以通过近似值获得准确值. 于是可以得到下面的定理:

定理 1.1 一个实数可以进行零误差计算的充要条件是该数属于某个一致离散集合.

通俗地讲, 一个一致离散集合的最优隔离界是这个集合的分辨极限. 只有近似计算的误差控制小于这个分辨极限的一半才有可能通过其近似值恢复出其准确值. 也就是说, 这个一致离散集合的最优隔离界的一半就是零误差计算中近似值误差控制的上界, 称之为理论控制界. 零误差计算无论采用何种计算方法, 其误差控制不可能大于这个理论控制界.

理论控制界是一种理想情形. 由于所使用的计算方法本身的原因, 对从近似值恢复出准确值所需要的误差控制界可能有更严格的要求. 在第 2 节中将会看到, 有理数的零误差计算的理论控制界为 $1/(2N(N-1))$, 采用连分数方法恢复其准确值需要的误差控制是 $1/(2N^2)$, 与理论控制界几乎一样, 其中 N 为有理数分母绝对值的上界. 但是, 高次代数数的零误差计算的理论控制界与所采用的计算方法需要的误差控制就不一样了 (第 3 节).

具体来讲, 第 2 节分析有理数的零误差计算, 给出可零误差计算的有理数的一致离散集合的最优隔离界 (必要条件), 这与 Zhang 和 Feng [2] 给出的基于连分数方法的有理数重构所需的误差控制 (充分条件) 几乎是一样的. 第 3 节分析高次代数的零误差计算问题, 给出可以进行零误差计算的代数数形成的一致离散集合的最优隔离界的下界 (定理 3.1), 因此, 在一定程度上可以看成是代数数可进行零误差计算的必要条件; 然后基于近期 PSLQ 算法的数值扰动分析的结果 (参见文献 [3]) 给出一个新的代数数零误差计算的误差控制条件 (3.15), 这一误差控制条件结果优于基于 LLL 算法的方案的误差控制条件 (3.10). 第 4 节探讨零误差计算中尚待研究的一些问题和潜在的应用.

记号 本文所有向量都用粗斜体的小写 (英文或希腊) 字母表示; 对向量 $\boldsymbol{x}$, $\|\boldsymbol{x}\|$ 表示 $\boldsymbol{x}$ 的 2- 范数, $\|\boldsymbol{x}\|_\infty$ 表示 $\boldsymbol{x}$ 的 ∞- 范数; 用 $\mathbb{Z}_d[X]$ 表示次数不超过 d 的单变元整系数多项式集合; 对于次数为 d 的多项式 $f(X)=f_0+f_1X+\cdots+f_dX^d$, 记其系数向量 $(f_0,f_1,\ldots,f_d)$ 为 $\boldsymbol{f}$.

2 有理数的零误差计算

显然, 若事先知道待求的准确数是整数, 那么当其近似值与准确值之间的误差小于 1/2 时, 便可通过四舍五入的方式恢复出这个整数的准确值.

若事先仅知道这个准确数是有理数, 则不能通过上述零误差计算的思想获得准确有理数. 这是因为有理数集合是稠密集, 对于给定的一个有理点, 总存在另外的有理点与这个给定的有理点的距离任意小. 但是, 若还知道这个准确有理数分母的绝对值不超过 N, 则该有理数便属于一个离散集合, 且有如下结果:

引理 2.1 设 N 为正实数, Ω_N 为分母的绝对值不超过 N 的有理数的集合, 则 Ω_N 为一个一致离散集合, 其最优隔离界为 $\frac{1}{N(N-1)}$.

证明 首先注意到对 Ω_N 中任意的两个不同的数 $\frac{m}{p}$ 和 $\frac{n}{q}$, 有

$$\left|\frac{m}{p}-\frac{n}{q}\right|=\frac{|qm-pn|}{|pq|}\geqslant\frac{1}{N(N-1)}.$$

这说明 Ω_N 的隔离界不小于 $\frac{1}{N(N-1)}$. 另一方面, 注意到 $\frac{1}{N}\in\Omega_N$ 和 $\frac{1}{N-1}\in\Omega_N$, 并且有 $|\frac{1}{N}-\frac{1}{N-1}|=\frac{1}{N(N-1)}$. 证毕. □

引理 2.1 指出 Ω_N 是一致离散集合, 其最优隔离界为 $\frac{1}{N(N-1)}$. 因此, 若准确有理数与它的某近似值的误差小于 $\frac{1}{2N^2} < \frac{1}{2N(N-1)}$, 则从理论上可以由该近似值恢复出准确有理数. 剩下的问题就是如何恢复. 首先能想到的就是采用连分数算法来获得. 具有如下形式的展开式称为连分数:

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cdots}}},$$

其中 a_0 为整数, $a_1, a_2, a_3, \ldots$ 为正整数. 通常将以上的连分数表示简记为 $[a_0; a_1, a_2, \ldots]$. 对于有限连分数, 注意到

$$[a_0; a_1, a_2, \ldots, a_n, 1] = [a_0; a_1, a_2, \ldots, a_n + 1].$$

利用连分数方法, 文献 [2] 给出如下的定理:

定理 2.1 (参见文献 [2, 定理 9]) 设 p 和 q 是使得 p/q 为最简分数的两个正整数, N 是满足 $N \geqslant \max\{q, 2\}$ 的正整数, r 是 p/q 的一个满足 $|r - p/q| < 1/(2N^2)$ 的近似值. 若 p/q 的连分数表示为 $p/q = [b_0; b_1, \ldots, b_L]$, 则 r 的连分数表示为 $r = [b_0; b_1, \ldots, b_M]$, 并且 $L \leqslant M$; 当 $L < M$ 时, 对任意满足 $L < T \leqslant M$ 的正整数 T, 有理数 $g = [b_0; b_1, \ldots, b_T]$ 的分母都大于 N.

假定对于一个未知的正有理数 p/q, 只知其分母绝对值的上界 N 和近似值 r. 由定理 2.1 知, 当 $|r - p/q| < 1/(2N^2)$ 时, p/q 的连分数 $[b_0; b_1, \ldots, b_L]$ 是近似值 r 的连分数 $[b_0; b_1, \ldots, b_M]$ 的前半部分. 虽然并不知道 L 是多少, 但有理数 $[b_0; b_1, \ldots, b_i]$ 的分母将提供有用的信息: 一旦 $[b_0; b_1, \ldots, b_i]$ 的分母超过 N, 则 $p/q = [b_0; b_1, \ldots, b_{i-1}]$. 具体参见算法 1.

算法 1 有理数恢复

输入: 正整数 N 和一个浮点数 $r \in (0, 1)$.
输出: 一个有理数 b.

令 t 为 r 的有理数表示, $h_0 := 0$, $h_1 := 1$, $k_0 := 1$, $k_1 := 0$; $a := \lfloor t \rfloor$, $b := t - a$; $k_2 := a \cdot k_1 + k_0$.
while $k_2 \leqslant N$ **do**
  $\binom{h_2}{k_2} := a \cdot \binom{h_1}{k_1} + \binom{h_0}{k_0}$, $\binom{h_0}{h_1} := \binom{h_1}{h_2}$, $\binom{k_0}{k_1} := \binom{k_1}{k_2}$.
  if $b = 0$ **then**
    $k_2 := N + 1$.
  else
    更新 $t := \frac{1}{b}$, $a := \lfloor t \rfloor$, $b := t - a$.
  end if
end while
return $b := h_0/k_0$.

算法 1 的输入是待求有理数分母的一个上界 N 和一个满足定理 2.1 中误差控制条件的近似值 r. 这里将 r 限制在 $(0, 1)$ 区间内是合理的, 因为有理数的整数部分总是精确的. 该算法通过计算 r 的连分数, 计算出一项就检查其分母是否超过 N, 一旦超过 N 就将上次连分数的值输出. 定理 2.1 保证了算法的正确性.

引理 2.1 和定理 2.1 显示, 采用连分数方法恢复准确值所需要的控制几乎达到了理论控制界, 在这种意义下, 这已是最优结果.

若记 r 的有理表示为 $t=m/n$, 其中 $m<n$ 为两个正整数, 则算法 1 本质上是对 n 和 m 进行扩展的 Euclid 算法, 因此可应用如文献 [4–8] 中的技术对算法 1 进行优化, 使得可以在不超过 $O(\mathcal{M}(s)\log s)$ 次位操作内计算出对应的准确值 b, 其中 $\max\{N,n,m\}\leqslant 2^s$, $\mathcal{M}(s)$ 表示两个不超过 2^s 的整数相乘的位复杂度.

3 代数数的零误差计算

关于实 (复) 数的零误差计算, 面临的首要问题是实 (复) 数的准确表示. 这里, 实 (复) 数的准确表示是指用有限的信息 (如有限长度的比特串) 来准确地表示实 (复) 数. 在这种意义下, 几乎所有的实 (复) 数都没有准确表示, 但代数数可以用有限信息表示. 本节将给出一种代数数表示方法, 并以此为基础阐述如何通过代数数的近似值获得准确值.

代数数是单变元整系数多项式的根. 给定代数数 α, 称以 α 为根的次数最低的整系数多项式 $P_\alpha(X)$ 为其极小多项式, $P_\alpha(X)$ 的次数称为代数数 α 的次数, $P_\alpha(X)$ 的高度 (系数向量的 ∞- 范数) 称为代数数 α 的高度. 对于实代数数, 文献 [9, 第 8.5 小节] 给出了以下 3 种表示.

- 序表示法: 实代数数 α 由以它为根的一个整系数多项式 $f(X)$ 和它在 $f(X)$ 的根从小到大的排序中的序号 j 表示.
- 符号表示法: 实代数数 α 由以它为根的一个整系数多项式 $f(X)$ 和 $f'(X)$ 在 α 处的符号序列 $\bar{s}$ 表示, 其中 $\bar{s}$ 表示 $f'(X)$ 的 Fourier 序列在 α 处取值的符号函数.
- 区间表示法: 实代数数 α 由以它为根的一个整系数多项式 $f(X)$ 和一个区间 $[a,b]$ 的两个端点组成, 而 α 是 $f(X)$ 在这个区间中唯一的根.

以上 3 种代数数表示都需要一个以代数数为根的多项式 $f(X)$ 和代数数的位置信息. 然而在实际计算中, 人们往往只能获得代数数的部分信息, 如该代数数的一个近似值. 当然, 仅仅知道代数数的近似值是不保证能恢复出准确代数数的, 因为代数数集在 $\mathbb{C}$ 中是稠密的. 但若还知道代数数的高度和次数的上界, 则被界定的所有代数数便形成一个 (有限) 离散集合, 在理论上, 只要近似值的误差控制在该一致离散集合的隔离界的一半范围内, 就能确定该近似值对应的准确代数数. 为此, 这里讨论第 4 种代数数表示方法: $(\bar{\alpha},d,N)$, 其中 $\bar{\alpha}$ 为待求准确代数数 α 的近似值, d 和 N 分别为 α 的次数和高度的上界. 与前面提及的几种代数数的表示方法相比, 这种表示最大的优势就是便于计算, 但想要获得其准确表示, 必须从 $(\bar{\alpha},d,N)$ 这个三元组中重构 α 的极小多项式 $P_\alpha(X)$.

第 3.1 小节首先给出由次数不超过 d 和高度不超过 N 的代数数构成的一致离散集合中最优隔离界的一个下界, 这个估计可以看成是代数数可进行零误差计算需要满足的必要条件. 第 3.2 和 3.3 小节分别讨论从代数数的近似值恢复准确值的两个具体算法和它们的控制误差条件.

3.1 代数数的隔离界

有理数是次数为 $n=1$ 的实代数数, 并且, 对于绝对值小于 1 的有理数, 其分母的上界就对应于其极小多项式高度的上界. 引理 2.1 已经指出, 对于由次数为 1、极小多项式高度上界为 N 的实代数数组成的一致离散集合, 最优隔离界为 $\frac{1}{N(N-1)}$. 下面将对 $n\geqslant 2$ 的情形, 给出最优隔离界的下界.

命题 3.1 (参见文献 [10, 命题 (1.6)]) 设 $h(X)$ 和 $g(X)$ 是次数分别为 n 和 m 的非零整系数多项式. 设 $\alpha\in\mathbb{C}$ 为 $h(X)$ 的根并且 $|\alpha|\leqslant 1$. 若 $h(X)$ 不可约且 $g(\alpha)\neq 0$, 则

$$|g(\alpha)|>\frac{1}{n\|\boldsymbol{h}\|^m\|\boldsymbol{g}\|^{n-1}}\geqslant\frac{1}{n(n+1)^{m/2}(m+1)^{(n-1)/2}\|\boldsymbol{h}\|_\infty^m\|\boldsymbol{g}\|_\infty^{n-1}},$$

其中 $\boldsymbol{h}$ 和 $\boldsymbol{g}$ 分别表示 $h(X)$ 和 $g(X)$ 的系数向量, $\|\cdot\|$ 表示向量的 2- 范数, $\|\cdot\|_\infty$ 表示 ∞- 范数.

上面命题表明对于两个不共轭的代数数 (具有不同的极小多项式), 若它们极小多项式的次数和高度有界, 则它们之间的距离也有非零的下界. 实际上, 若设 β 为 $g(x)$ 的根, 而且 $|\beta|\leqslant 1$, 则 $|g(\alpha)|=|g(\alpha)-g(\beta)|\leqslant|\alpha-\beta|\cdot\|\boldsymbol{g}\|_\infty m(m+1)/2$. 因而,

$$|\alpha-\beta|>\frac{2}{nm(m+1)^{\frac{n+1}{2}}(n+1)^{\frac{m}{2}}\|\boldsymbol{h}\|_\infty^m\|\boldsymbol{g}\|_\infty^n}.$$

设 d 和 N 分别为代数数次数和高度的上界, 则

$$|\alpha-\beta|>\frac{2}{d^2(d+1)^{d+\frac{1}{2}}N^{2d}}. \tag{3.1}$$

注意命题 3.1 中的限制条件 $|\alpha|\leqslant 1$ 是可以去掉的. 因为若 $|\alpha|>1$ 为 $h(X)$ 的根, 则 $1/\alpha$ 为 $H(X)=x^nh(1/X)$ 的根, 而 $H(X)$ 的次数和高度均与 $h(X)$ 一样, 而且有 $|\frac{1}{\alpha}-\frac{1}{\beta}|\leqslant\frac{|\alpha-\beta|}{|\alpha\beta|}\leqslant|\alpha-\beta|$, 此时便可以应用命题 3.1.

接下来分析极小多项式的共轭根的隔离界. 有下面的命题成立:

命题 3.2 (参见文献 [11, 第 262 页]) 设 $h(X)$ 为次数为 n 的无平方整系数多项式, 则

$$\mathrm{sep}(h)>\frac{\sqrt{3}}{n^{(n+2)/2}\|\boldsymbol{h}\|_2^{n-1}},$$

这里 $\mathrm{sep}(h)=\min_{z_i\neq z_j}|z_i-z_j|$, 其中 $z_1,\ldots,z_n$ 为 $h(X)$ 的根.

以上命题给出了同一多项式不同根之间距离的下界, 结合不等式 (3.1) 给出的是非共轭根之间的距离, 再由 $n\leqslant d$ 和 $\|\boldsymbol{h}\|_\infty\leqslant N$ 得

$$\begin{aligned}\frac{\sqrt{3}}{n^{(n+2)/2}\|\boldsymbol{h}\|_2^{n-1}}&>\frac{\sqrt{3}}{n^{(n+2)/2}(n+1)^{(n-1)/2}\|\boldsymbol{h}\|_\infty^{n-1}}\\&>\frac{\sqrt{3}}{n^2(n+1)^{(n-2)/2}(n+1)^{(n-1)/2}\|\boldsymbol{h}\|_\infty^{n-1}}\\&>\frac{2}{d^2(d+1)^{d+\frac{1}{2}}N^{2d}}.\end{aligned}$$

注意以上推导是针对 $n\geqslant 2$ 的情形. 当 $n=1$ 时, 最优隔离界的下界为 $1/N^2$, 因而可以去掉 $n\geqslant 2$ 的限制, 得到如下结论:

定理 3.1 设 E 是由次数不高于 d 和高度不大于 N 的代数数组成的集合, 其上的距离定义为 $d(\alpha,\beta)=|\alpha-\beta|$, 则 E 形成一个一致离散的集合. 特别地, 对任意的 $\alpha,\beta\in E$, 有 $d(\alpha,\beta)>2/(d^2(d+1)^{d+1/2}N^{2d})$.

一般来讲, 定理 3.1 中给出的隔离界不是最优的. 但是, 上述定理仍然表明, 若代数数 α 次数和高度的界分别为 d 和 N, 则当其近似值 $\bar{\alpha}$ 满足

$$|\bar{\alpha}-\alpha|<\frac{1}{d^2(d+1)^{d+1/2}N^{2d}} \tag{3.2}$$

时, 理论上可以通过其近似值 $\bar{\alpha}$ 获得准确值 α. 剩下的问题是对给定的满足 (3.2) 的三元组 $(\bar{\alpha},d,N)$, 设计高效算法计算出代数数的极小多项式, 从而完成代数数的零误差计算. 下面将分别介绍如何采用 LLL 算法和 PSLQ 算法来求解这一问题.

3.2 基于 LLL 算法的代数数零误差计算

文献 [10] 给出了通过代数数的近似值 $\bar{\alpha}$ 计算出代数数 α 的准确极小多项式的方法, 采用的主要工具是 LLL 格基约化算法 [12]. 作为准备, 需要简要回顾格的相关概念. 给定一组线性无关向量 $\boldsymbol{b}_1,\ldots,\boldsymbol{b}_n\in\mathbb{R}^m$, 称集合

$$\left\{\sum_{i=1}^{n} z_i\boldsymbol{b}_i : z_i\in\mathbb{Z}\right\}$$

为一个格, 常记为 L. 这组线性无关的向量 $\boldsymbol{b}_1,\ldots,\boldsymbol{b}_n$ 称为格 L 的一组基. 为方便起见, 也常称矩阵 $B=(\boldsymbol{b}_1,\ldots,\boldsymbol{b}_n)\in\mathbb{R}^{m\times n}$ 为格 L 的一组基. 对格 L, 总存在一个非零最短向量, 其最短向量的长度记为 $\lambda(L)$. 对格的研究中, 有一个非常重要的问题: 对给定的格, 寻找该格的一组高质量 (向量长度更短, 向量间更接近正交) 的基. LLL 约化基 [12] 就是一组质量较高的基, 能满足很多实际应用的需求, LLL 算法 [12] 就是用来计算给定格的一组 LLL 约化基的算法.

定义 3.1 [13] 设 $\Xi=(\delta,\eta,\theta)\in\mathbb{R}^3$. 若 $\eta\in[1/2,1)$, $\theta\in[0,1)$, $\delta\in(\eta^2,1]$, 则称 Ξ 为一组合法的 LLL 参数. 设 $B\in\mathbb{R}^{m\times n}$ 是一个非奇异的矩阵, 其 QR 分解为 $B=Q\cdot R$, 其中 Q 为正交矩阵, R 为对角线元素为正值的上三角矩阵. 称 B 是 Ξ-LLL 约化的, 若

(1) 对 $i<j$, 有 $|r_{i,j}|\leqslant\eta r_{i,i}+\theta r_{j,j}$;

(2) 对 i, 有 $\delta\cdot r_{i,i}^2\leqslant r_{i,i+1}^2+r_{i+1,i+1}^2$.

Ξ-LLL 约化基具有如下性质:

定理 3.2 [13] 给定一组合法的 LLL 参数 $\Xi=(\delta,\eta,\theta)$, 若 $B\in\mathbb{R}^{m\times n}$ 为格 L 的一组 Ξ-LLL 约化基, 则 $r_{j,j}\leqslant\tau\cdot r_{j+1,j+1}$ 且 $\|\boldsymbol{b}_1\|\leqslant\tau^{n-1}\lambda(L)$, 其中 $\tau=\frac{\theta\eta+\sqrt{(1+\theta^2)\delta-\eta^2}}{\delta-\eta^2}$.

上述定理显示, Ξ-LLL 约化基与经典的 LLL 约化基具有类似的性质. 事实上, 若 $\Xi=(3/4,1/2,0)$, 就是经典的 LLL 约化基 [12]. 为方便起见, 下面固定 $\Xi=(3/4,1/2,0)$, 从而 $\tau=\sqrt{2}$.

为了利用 LLL 约化基计算 $\bar{\alpha}$ 的极小多项式, 首先要解决的问题就是构造一个格, 并建立多项式与格之间的一一对应关系, 使得其极小多项式对应于该格 LLL 约化基的第一个向量, 从而将极小多项式的计算问题转化成格的 LLL 约化基计算问题. 假设极小多项式的次数为 n, 定义 $(n+3)\times(n+1)$ 矩阵

$$B=\begin{pmatrix} 1 & 0 & \cdots & 0\\ 0 & 1 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & 1\\ 2^s\mathrm{Re}(\bar{\alpha}_0) & 2^s\mathrm{Re}(\bar{\alpha}_1) & \cdots & 2^s\mathrm{Re}(\bar{\alpha}_n)\\ 2^s\mathrm{Im}(\bar{\alpha}_0) & 2^s\mathrm{Im}(\bar{\alpha}_1) & \cdots & 2^s\mathrm{Im}(\bar{\alpha}_n)\end{pmatrix}, \tag{3.3}$$

其中 $\bar{\alpha}_i=\overline{\alpha^i}$ 是 α^i 的近似值, s 为正整数. 记 (3.3) 中的列向量为 $\boldsymbol{b}_0,\boldsymbol{b}_1,\ldots,\boldsymbol{b}_n$, 它们生成的格记为 L_s, 则格 L_s 与次数不大于 d 的多项式集合 $\mathbb{Z}_d[X]$ 之间存在如下对应关系:

$$\begin{aligned}&\mathbb{Z}_d[X]\to L_s,\\ &g(X)=\sum_{i=0}^{d} g_iX^i\mapsto\tilde{\boldsymbol{g}}=\sum_{i=0}^{d} g_i\boldsymbol{b}_i.\end{aligned} \tag{3.4}$$

显然, 有

$$\|\tilde{\boldsymbol{g}}\|^2 = \|\boldsymbol{g}\|^2 + 2^{2s}|g(\bar{\alpha})|^2, \tag{3.5}$$

其中 $\boldsymbol{g}$ 表示多项式 $g(X)$ 的系数向量. 若 $|\alpha^i - \bar{\alpha}_i| < 2^{-s}$, 则 $|g(\bar{\alpha}) - g(\alpha)| < 2^{-s}dN$. 于是, 若 $h(X) \in \mathbb{Z}_d[X]$ 的高度不超过 N 且 $h(\alpha) = 0$, 则 $|h(\bar{\alpha})| < 2^{-s}dN$, 从而 $\|\tilde{\boldsymbol{h}}\|^2 = \|\boldsymbol{h}\|^2 + 2^{2s}|h(\bar{\alpha})|^2 \leqslant (d+1)N^2 + d^2N^2 < (d+1)^2N^2$. 令格 L_s 的一组 LLL 约化基的第一个向量为 $\tilde{\boldsymbol{v}}$, 则其满足 $\|\tilde{\boldsymbol{v}}\| \leqslant 2^d\lambda(L_s)$. 所以, 若格 L_s 包含了 $\tilde{\boldsymbol{h}}$, 则 $\|\tilde{\boldsymbol{v}}\| \leqslant 2^d\lambda(L_s) \leqslant 2^d(d+1)N$. 如果能证明 "在一定的控制下, 若格 L_s 不包含 $\tilde{\boldsymbol{h}}$, 则 $\|\tilde{\boldsymbol{v}}\| > 2^d(d+1)H$", 那么就可以通过计算格 L_s 的 LLL 约化基的方式获得极小多项式.

为此, 回顾命题 3.1: 设 α 为次数不超过 d 和高度不超过 N 的代数数. 若 $g(X) \in \mathbb{Z}_d[X]$ 使得 $g(\alpha) \neq 0$, 则 $|g(\alpha)| > 1/(d((d+1)N^2)^{d/2}\|\boldsymbol{g}\|^{d-1})$. 于是, 当 $g(\alpha) \neq 0$ 时, 若 $\|\boldsymbol{g}\| > 2^d(d+1)N$, 显然, $\|\tilde{\boldsymbol{g}}\| > \|\boldsymbol{g}\| > 2^d(d+1)N$; 若 $\|\boldsymbol{g}\| \leqslant 2^d(d+1)N$, 则有

$$\begin{aligned}\|\tilde{\boldsymbol{g}}\| &> 2^s|g(\bar{\alpha})| > 2^s(|g(\alpha)| - |g(\bar{\alpha}) - g(\alpha)|)\\ &> 2^s\left(\frac{1}{d((d+1)N^2)^{d/2}\|\boldsymbol{g}\|^{d-1}} - 2^{-s}d\|\boldsymbol{g}\|_\infty\right)\\ &= \frac{2^s}{d((d+1)N^2)^{d/2}\|\boldsymbol{g}\|^{d-1}} - d\|\boldsymbol{g}\|_\infty.\end{aligned} \tag{3.6}$$

再由 $\|\boldsymbol{g}\|_\infty \leqslant \|\boldsymbol{g}\| \leqslant 2^d(d+1)N$ 知,

$$\begin{aligned}\|\tilde{\boldsymbol{g}}\| &> \frac{2^s}{d((d+1)N^2)^{d/2}(2^d(d+1)N)^{d-1}} - d2^d(d+1)N\\ &= \frac{2^s}{2^{d(d-1)}d(d+1)^{3d/2-1}N^{2d-1}} - 2^dd(d+1)N.\end{aligned} \tag{3.7}$$

于是, 通过适当选取 s 使得 $2^s \geqslant 2^{d^2}d^2(d+1)^{3d/2}N^{2d}$ 便能保证 $\|\tilde{\boldsymbol{g}}\| > 2^d(d+1)N$. 从而, 得到以下定理:

定理 3.3 (参见文献 [10, 定理 (1.15)]) 设代数数 α 的次数和高度的界分别为 d 和 N, $\bar{\alpha}$ 为其近似值, 采用 $\Xi = (\delta, \eta, \theta)$ 为 LLL 的参数, 并且

$$|\alpha^i - \bar{\alpha}_i| \leqslant 2^{-s}, \quad i = 1, \ldots, d. \tag{3.8}$$

令 s 是使得

$$2^s \geqslant 2^{d^2} \cdot (d+1)^{(3d+4)/2} \cdot N^{2d} \tag{3.9}$$

成立的最小整数. 对于正整数 $n \leqslant d$, 构造如 (3.3) 的矩阵 B, 记其列向量生成的格为 L_s. 设格 L_s 的一组 LLL 约化基的第一个向量为 $\tilde{\boldsymbol{v}} = \sum_{i=0}^n v_i\boldsymbol{b}_i$, 其对应的多项式记为 $v(X) = \sum_{i=0}^n v_iX^i$, 则下列条件等价:

(1) $\|\tilde{\boldsymbol{v}}\| \leqslant 2^d \cdot (d+1) \cdot N$;

(2) $v(\alpha) = \sum_{i=0}^n v_i\alpha^i = 0$;

(3) α 的次数最多为 n,

并且, 若 n 为 α 的次数, 则 α 的极小多项式必为 $v(X)$.

以上定理说明, 当代数数的近似值 $\bar{\alpha}$ 达到定理中要求时, 如果极小多项式所对应的向量在格 L_s 中, 那么通过 LLL 算法得到的约化基的第一个向量的 2- 范数一定小于 $2^d(d+1)N$, 否则就大于 $2^d(d+1)N$. 也就是说, 对于 $n=1,\ldots,d$, 通过 (3.3) 构造格 L_s, 然后计算 L_s 的 LLL 约化基的第一个向量 $\tilde{\boldsymbol{v}}$, 检查其 2- 范数是否小于 $2^d(d+1)N$. 若成立, 则 $\tilde{\boldsymbol{v}}$ 对应的多项式 $v(X)$ 就是 α 的极小多项式; 否则, n 增加 1, 重复以上步骤. 而由 (3.9) 知, $s=O(d^2+d\log N)$.

定理 3.3 中要求代数数的近似值满足 (3.8). 在计算中可通过两个步骤实现 (3.8): 计算 $\bar{\alpha}$ 满足 $|\alpha-\bar{\alpha}|<1/(2^{s+2}d)$ 并用 $\bar{\alpha}^i$ 的近似值 $\bar{\alpha}_i$ 代替使其满足 $|\bar{\alpha}^i-\bar{\alpha}_i|<2^{-s-1/2}$. 可以验证, 在 $|\alpha|\leqslant 1$ 的假设下, 以上两条保证了定理中的要求. 于是, 得到基于浮点 LLL 算法的代数数极小多项式重构算法 2.

算法 2 基于 LLL 算法重构模不大于 1 的代数数的极小多项式

输入: 代数数次数的上界 d, 高度的上界 N, LLL 约化基合法参数 Ξ, 使得 (3.9) 成立的最小正整数 s, 以及满足

$$|\alpha-\bar{\alpha}|<\frac{1}{2^{s+2}d} \tag{3.10}$$

的代数数的近似值 $\bar{\alpha}$.

输出: 代数数 α 的极小多项式 $P_\alpha(X)$.

1: 对 $i=1,\ldots,d$ 计算 $\bar{\alpha}_i$ 使其满足 $\left|\bar{\alpha}^i-\bar{\alpha}_i\right|<2^{-s-1/2}$.
2: **for** n from 1 to d **do**
3: 构造格 L_s.
4: 调用任一 LLL 算法计算格 L_s 的一组 LLL 约化基, 基中的第一个向量记为 $\tilde{\boldsymbol{v}}$.
5: **if** $\|\tilde{\boldsymbol{v}}\|<2^{d/2}(d+1)N$ **then**
6: **return** $\tilde{\boldsymbol{v}}$ 对应的多项式 $v(X)$.
7: **end if**
8: **end for**

当代数数满足 $|\alpha|>1$ 时, 令 $\beta=1/\alpha$, 利用算法 2 求 β 的极小多项式 $g(X)$, 然后得到 α 的极小多项式 $P_\alpha(X)=X^ng(1/X)$, 其中 n 为极小多项式 $g(X)$ 的次数. 此时, 只需将相应的误差控制变为 $|\alpha-\bar{\alpha}|<1/(3\cdot 2^{s+2}d)$ 即可.

若在步骤 4 中采用不同的浮点 LLL 算法, 则可以得到相应算法 2 的位复杂度上界 (见表 1).

3.3 基于 PSLQ 算法的代数数零误差计算

PSLQ 是一个计算整数关系的算法 [18]. (PS 是指部分和 (partial sum), LQ 是指矩阵的 LQ 分解 (矩阵 B 的 LQ 分解等价于矩阵 B^{T} 的 QR 分解).) 给定一组数 $\boldsymbol{\alpha}=(\alpha_i)_{1\leqslant i\leqslant n}\in\mathbb{R}^n$, 若存在 $\boldsymbol{m}=(m_i)_{1\leqslant i\leqslant n}\in\mathbb{Z}^n$ 使得 $\langle\boldsymbol{\alpha},\boldsymbol{m}\rangle=0$, 则称 $\boldsymbol{m}$ 是 $\boldsymbol{\alpha}$ 的一组整数关系. 易知 $\boldsymbol{\alpha}$ 所有的整数关系形成一个格, 记为 $\Lambda_{\boldsymbol{\alpha}}$. 若 α 是一个 n 次代数数, 则其极小多项式的系数向量就是 $\boldsymbol{\alpha}=(1,\alpha,\ldots,\alpha^n)\in\mathbb{R}^{n+1}$ 的一组整数关系. 因此, 代数数的零误差计算本质上依赖于整数关系的计算.

表 1 基于浮点 LLL 算法的代数数极小多项式重构复杂度上界

算法	比特复杂度上界
L^2 (参见文献 [14, 定理 1])/H-LLL (参见文献 [15, 定理 4.4])	$O(d^{7+\varepsilon}+d^{6+\varepsilon}\log N+n^{5+\varepsilon}\log^2 N)$
渐近 LLL (参见文献 [16, 定理 5])	$O(d^{6+\varepsilon}+d^{4+\varepsilon}\log^2 N)$
L^1 (参见文献 [17, 定理 7])	$O(d^{6+\varepsilon}+d^{5+\varepsilon}\log N+d^{\omega+1+\varepsilon}\log^{1+\varepsilon} N)$

对 $\boldsymbol{\alpha}=(\alpha_i)_{1\leqslant i\leqslant n}\in\mathbb{R}^n$, PSLQ 算法首先通过 $\boldsymbol{\alpha}$ 来构造超平面矩阵 $H_{\boldsymbol{\alpha}}=(h_{i,j})\in\mathbb{R}^{n\times(n-1)}$:

$$h_{ij}=\begin{cases}0, & \text{若 } 1\leqslant i<j\leqslant n-1,\\ \dfrac{s_{i+1}}{s_i}, & \text{若 } 1\leqslant i=j\leqslant n-1,\\ \dfrac{-x_ix_j}{s_is_{j+1}}, & \text{若 } 1\leqslant j<i\leqslant n,\end{cases}\tag{3.11}$$

其中 $s_j^2=\sum_{k=j}^n\alpha_k^2$ $(j=1,\dots,n)$ 称为 $\boldsymbol{\alpha}$ 的部分和. 由构造可知 $H_{\boldsymbol{\alpha}}$ 是下梯形矩阵, 它的每个列向量为单位向量, 且不同列向量彼此正交, $\boldsymbol{\alpha}$ 与 $H_{\boldsymbol{\alpha}}$ 的列向量正交; 并且 $\boldsymbol{\alpha}$ 的任意一个整数关系 $\boldsymbol{m}$ 与超平面矩阵 $H_{\boldsymbol{\alpha}}$ 之间有如下的联系:

定理 3.4 (参见文献 [18, 定理 1]) 设 $\boldsymbol{m}\in\mathbb{Z}^n$ 为 $\boldsymbol{\alpha}\in\mathbb{R}^n$ 的任意一个非零整数关系. 对于任意的幺模矩阵 $A\in\mathrm{GL}(n,\mathbb{Z})$ 都存在正交矩阵 $Q\in\mathbb{R}^{(n-1)\times(n-1)}$ 使得 $H=AH_{\boldsymbol{\alpha}}Q=(h_{i,j})$ 是下梯形矩阵. 若 H 的所有对角元 $h_{j,j}\neq0$, 则

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

PSLQ 算法本质上是对 $H_{\boldsymbol{\alpha}}$ 进行迭代. 每次迭代分为两步: 通过 Hermite 约化 (参见文献 [18, 定义 3]) 和 Bergman 交换规则 [19] 来产生幺模矩阵 A, 然后再对 AH 进行 LQ 分解, 将 H 更新为对应的 L- 因子, 使其重新变为下梯形矩阵. PSLQ 算法对矩阵 H 的行进行 Bergman 交换, 使得每一次交换后 $\max_{1\leqslant j\leqslant n-1}|h_{j,j}|$ 不会增加; 并且, 当 Bergman 交换发生在 H 的第 $n-1$ 与 n 行之间时, 交换后的 $\max_{1\leqslant j\leqslant n-1}|h_{j,j}|$ 将严格减小. 由此, 若 $\boldsymbol{\alpha}$ 存在非零整数关系, 则定理 3.4 保证了 PSLQ 算法终能使得 $H=AH_{\boldsymbol{\alpha}}Q$ 满足

$$h_{n,n-1}=0,\tag{3.12}$$

此时矩阵 A^{-1} 的倒数第二列就是 $\boldsymbol{\alpha}$ 的一组整数关系, 并且能够证明如下定理:

定理 3.5 (参见文献 [18, 定理 2]) 假设 $\boldsymbol{\alpha}\in\mathbb{R}^n$ 有整数关系, 并记 $\lambda(\Lambda_{\boldsymbol{\alpha}})$ 为 $\boldsymbol{\alpha}$ 的最短非零整数关系的长度, 则 PSLQ 算法 (参见文献 [18, 第 3 节]) 将在不超过

$$\binom{n}{2}\frac{\log(\gamma^{n-1}\lambda(\Lambda_{\boldsymbol{\alpha}}))}{\log\tau}$$

次迭代后返回一组 $\boldsymbol{\alpha}$ 的整数关系 $\boldsymbol{m}\in\mathbb{Z}^n$, 并且 $\|\boldsymbol{m}\|\leqslant\gamma^{n-2}\lambda(\Lambda_{\boldsymbol{\alpha}})$, 其中 $\tau=1/\sqrt{1/\rho^2+1/\gamma^2}$, $\gamma>2/\sqrt{3}$, $\rho=2$.

需要指出的是, 上述结论是在输入数据 $\boldsymbol{\alpha}$ 是精确的且计算过程也是精确的假设下得到的. 但正如前面已经指出的那样, 在计算机上精确地表示实数尚存困难, 计算往往只能依赖于高精度的浮点算术. 因此, PSLQ 算法的数值稳定性分析是不可或缺的. 下面将介绍数值 PSLQ 算法的一项研究进展 (参见文献 [3]), 并讨论其在代数数零误差计算中的应用.

3.3.1 数值 PSLQ 算法的设计与分析

现假设输入不再是精确的, 即输入的是 $\boldsymbol{\alpha}\in\mathbb{R}^n$ 的一个近似向量 $\bar{\boldsymbol{\alpha}}$, 并且满足 $\|\boldsymbol{\alpha}-\bar{\boldsymbol{\alpha}}\|<\varepsilon_1$. 对于输入的 $\bar{\boldsymbol{\alpha}}$, 目标是找到一个非零的整数向量 $\boldsymbol{m}\in\mathbb{Z}^n$ 使其有望成为 $\boldsymbol{\alpha}$ 的一个整数关系. 为此, 需要解决以下两个问题:

- 由于输入数据的误差, 算法终止条件 (3.12) 将不能准确判定. 如何设计数值 PSLQ 算法的终止条件使其有限步终止?
- 假设目标是使得算法返回 $\boldsymbol{m}\in\mathbb{Z}^n$ 满足 $|\langle\boldsymbol{\alpha},\boldsymbol{m}\rangle|<\varepsilon$, 应该如何控制 $\bar{\boldsymbol{\alpha}}$ 的误差 ε_1?

为解决第一个问题, 需要将算法的终止条件由 (3.12) 更改为

$$|h_{n,n-1}|<\varepsilon_2,$$

并对原来的算法进行相应调整设计出数值算法 $\mathrm{PSLQ}_\varepsilon$ (参见文献 [3, 算法 5]). $\mathrm{PSLQ}_\varepsilon$ 算法与原始的 PSLQ 算法相比, 除了调整终止条件, 最大的区别还表现在: (1) 原始的 PSLQ 算法以 $\boldsymbol{\alpha}\in\mathbb{R}^n$ 为输入, 而 $\mathrm{PSLQ}_\varepsilon$ 算法以 $\boldsymbol{\alpha}$ 的近似超平面矩阵 $\overline{H}_{\boldsymbol{\alpha}}$ 为输入, 且满足

$$\|\overline{H}_{\boldsymbol{\alpha}}-H_{\boldsymbol{\alpha}}\|_F\leqslant\varepsilon_3, \tag{3.13}$$

其中 $\|\cdot\|_F$ 表示矩阵的 Frobenius 范数; (2) 原始 PSLQ 算法的输出是 $\boldsymbol{\alpha}$ 的一个精确的整数关系 (假设 $\boldsymbol{\alpha}$ 存在整数关系), 而 $\mathrm{PSLQ}_\varepsilon$ 算法的输出仅是一个整数向量 $\boldsymbol{m}\in\mathbb{Z}^n$, 该向量有望成为 $\boldsymbol{\alpha}$ 的整数关系, 即存在 $\varepsilon>0$ 使得 $|\langle\boldsymbol{\alpha},\boldsymbol{m}\rangle|<\varepsilon$. 由文献 [3, 定理 3.2] 知, $\mathrm{PSLQ}_\varepsilon$ 将在

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

次迭代后终止, 返回一个整数向量 $\boldsymbol{m}\in\mathbb{Z}^n$, 其中 $\gamma>\sqrt{3}/2$, $\tau=1/\sqrt{1/4+1/\gamma^2}$.

假设想要通过 $\mathrm{PSLQ}_\varepsilon$ 以 $|h_{n-1,n-1}|<\varepsilon_2$ 为终止条件来计算 $\boldsymbol{\alpha}$ 的整数关系, 对应的近似超平面矩阵 $\overline{H}_{\boldsymbol{\alpha}}$ 满足 (3.13), 算法输出 $\boldsymbol{m}\in\mathbb{Z}^n$. 下面的定理给出了 ε_2 和 ε_3 与 $|\langle\boldsymbol{m},\boldsymbol{\alpha}\rangle|$ 之间的关系:

定理 3.6 (参见文献 [3, 定理 3.8]) 给定 $\boldsymbol{\alpha}=(\alpha_i)_{1\leqslant i\leqslant n}$, 设 $H_{\boldsymbol{\alpha}}$ 为如 (3.11) 构造的超平面矩阵, $\overline{H}_{\boldsymbol{\alpha}}$ 是 $H_{\boldsymbol{\alpha}}$ 的近似矩阵且满足 $\|H_{\boldsymbol{\alpha}}-\overline{H}_{\boldsymbol{\alpha}}\|_F<\varepsilon_3<\frac{\alpha_n}{2\sqrt{(n-2)\alpha_n^2+1}}$. 假设幺模矩阵 A 和正交矩阵 Q 使得 $H=(h_{i,j})=A\overline{H}_{\boldsymbol{\alpha}}Q$ 是以 $|h_{n,n-1}|<\varepsilon_2$ 为终止条件的算法 $\mathrm{PSLQ}_\varepsilon$ 终止时的状态, 记 $\boldsymbol{m}$ 为 A^{-1} 的第 $(n-1)$ 列, 则 $|\langle\boldsymbol{\alpha},\boldsymbol{m}\rangle|<C\cdot(\|\boldsymbol{m}\|\varepsilon_3+\alpha_n\varepsilon_2)$, 其中 $C=\frac{2(\sqrt{(n-2)\alpha_n^2+1}+|\alpha_n|)}{|\alpha_n|}$.

若设 $H_{\bar{\boldsymbol{\alpha}}}$ 为 $\bar{\boldsymbol{\alpha}}$ 按 (3.11) 构造的超平面矩阵, 则有如下引理:

引理 3.1 (参见文献 [3, 引理 4.1]) 设 $\boldsymbol{\alpha}$ 为 n- 维单位向量, $\bar{\boldsymbol{\alpha}}$ 为 $\boldsymbol{\alpha}$ 的近似向量. 按照 (3.11) 来构造 $H_{\boldsymbol{\alpha}}$ 和 $H_{\bar{\boldsymbol{\alpha}}}$. 若 $\|\boldsymbol{\alpha}-\bar{\boldsymbol{\alpha}}\|<\frac{1}{8n}$, 则 $\|H_{\boldsymbol{\alpha}}-H_{\bar{\boldsymbol{\alpha}}}\|_F<8n^{\frac{3}{2}}\|\boldsymbol{\alpha}-\overline{\boldsymbol{\alpha}}\|$.

由此, 若进一步假设 $\overline{H}_{\boldsymbol{\alpha}}=H_{\bar{\boldsymbol{\alpha}}}$, 则将引理 3.1 应用到定理 3.6 便可得到如下定理:

定理 3.7 (参见文献 [3, 定理 4.2]) 设 $\boldsymbol{\alpha}\in\mathbb{R}^n$ 为单位向量, 并设 $\varepsilon>0$. 假设 $\boldsymbol{\alpha}$ 存在一个 2- 范数小于 M 的整数关系. 给定 $\boldsymbol{\alpha}$ 的近似向量 $\bar{\boldsymbol{\alpha}}$ 满足 $\|\boldsymbol{\alpha}-\bar{\boldsymbol{\alpha}}\|<\varepsilon_1\leqslant\frac{\varepsilon}{16MCn^{3/2}}$, 若以 $|h_{n-1,n-1}|<\varepsilon_2\leqslant\frac{\varepsilon}{2C|\alpha_n|}$ 为终止条件的 $\mathrm{PSLQ}_\varepsilon$ 算法返回的 $\boldsymbol{m}\in\mathbb{Z}^n$ 满足 $\|\boldsymbol{m}\|<M$, 则

$$|\langle\boldsymbol{\alpha},\boldsymbol{m}\rangle|<\varepsilon,$$

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

于是, 通过对超平面矩阵引入如 (3.13) 的扰动, 对数值的 PSLQ 算法进行扰动分析, 上述定理建立了输入的 ε_1 与输出质量 ε 之间的关系, 从而回答了本小节所提的第二个问题.

进一步地, 若对输入 $\boldsymbol{\alpha}$ 有如命题 3.1 中类似的结论, 辅以定理 3.7 中的误差控制, 便能保证可以通过 $\boldsymbol{\alpha}$ 的近似值计算出 $\boldsymbol{\alpha}$ 的一个精确整数关系. 由此, 从近似值重构代数数极小多项式这一问题便得到解决.

3.3.2 数值 PSLQ 在代数数零误差计算中的应用

设实代数数 α 的次数不超过 d, 高度不超过 N, 并且 $|\alpha| \leqslant 1$. 对 $n \leqslant d$, 设 $\boldsymbol{x}$ 是向量 $\boldsymbol{\alpha} = (\alpha^n, \alpha^{n-1}, \ldots, 1)$ 对应的单位向量, 即 $\boldsymbol{x} = \boldsymbol{\alpha}/\|\boldsymbol{\alpha}\|$. 此时定理 3.7 中对应的 $x_{n+1} = (\alpha^{2n} + \alpha^{2(n-1)} + \cdots + 1)^{-\frac{1}{2}} \geqslant (d+1)^{-\frac{1}{2}}$, 故 $C \leqslant 4(d+1)^{1/2}$. 若取

$$\varepsilon_1 = \frac{1}{64(d+1)^{d+7/2}N^{2d}}, \quad \varepsilon_2 = \frac{1}{8(d+1)^{d+3/2}N^{2d-1}}, \tag{3.14}$$

则由定理 3.7 知, 当以 $|h_{n,n-1}| < \varepsilon_2$ 为终止条件的算法 $\mathrm{PSLQ}_\varepsilon$ 返回的 $\boldsymbol{m}$ 满足 $\|\boldsymbol{m}\|_\infty < N$ 时, 有

$$|\langle \boldsymbol{x}, \boldsymbol{m}\rangle| < \frac{1}{(d+1)^{d+1}N^{2d-1}}.$$

记 $\boldsymbol{m}$ 对应的多项式为 $m(X)$, 则

$$|m(\alpha)| = |\langle \boldsymbol{\alpha}, \boldsymbol{m}\rangle| = \|\boldsymbol{\alpha}\| \cdot |\langle \boldsymbol{x}, \boldsymbol{m}\rangle| \leqslant \sqrt{d+1} \cdot |\langle \boldsymbol{x}, \boldsymbol{m}\rangle| < \frac{1}{(d+1)^{d+1/2}N^{2d-1}}.$$

由命题 3.1 知, $m(\alpha) = 0$, 从而得到如下算法.

算法 3 基于 $\mathrm{PSLQ}_\varepsilon$ 算法重构模不大于 1 的代数数的极小多项式

输入: 代数数次数的上界 d, 高度的上界 N, 满足

$$|\alpha - \bar{\alpha}| < \frac{1}{128(d+1)^{d+11/2}N^{2d}} \tag{3.15}$$

的代数数的近似值 $\bar{\alpha}$.

输出: 代数数 α 的极小多项式 $P_\alpha(X)$.

1: **for** n from 1 to d **do**
2: 令 $\bar{\boldsymbol{\alpha}} := (\bar{\alpha}^n, \bar{\alpha}^{n-1}, \ldots, 1)$, 并令 $\bar{\boldsymbol{x}} := \bar{\boldsymbol{\alpha}}/\|\bar{\boldsymbol{\alpha}}\|$.
3: 按 (3.11) 构造 $\bar{\boldsymbol{x}}$ 的超平面矩阵 $H := H_{\bar{\boldsymbol{x}}}$.
4: 按 (3.14) 设置 ε_2 的值, 并以 $|h_{n,n}| < \varepsilon_2$ 为终止条件调用 $\mathrm{PSLQ}_\varepsilon$ 算法 (参见文献 [3, 算法 5]) 返回非零整向量 $\boldsymbol{m} \in \mathbb{Z}^{n+1}$.
5: **if** $\|\boldsymbol{m}\| < N$ **then**
6: **return** $\boldsymbol{m}$ 对应的多项式 $m(X) = \sum_{i=0}^{n} m_i X^{n-i}$.
7: **end if**
8: **end for**

事实上, 容易验证满足 (3.15) 的 $\bar{\alpha}$ 将使得步骤 2 中单位化之前的 $\bar{\boldsymbol{\alpha}}$ 满足

$$\|\boldsymbol{\alpha} - \bar{\boldsymbol{\alpha}}\| < \frac{1}{128\sqrt{3}(d+1)^{d+4}N^{2d}},$$

在单位化之后, 新的 $\bar{\boldsymbol{x}}$ 满足

$$\|\boldsymbol{x} - \bar{\boldsymbol{x}}\| < \frac{1}{64(d+1)^{d+7/2}N^{2d}} = \varepsilon_1,$$

从而满足定理 3.7 中的要求, 保证算法 3 的正确性. 当 $|\alpha| > 1$ 时, 可以通过第 3.2 小节末尾讨论的方法来类似地处理.

与基于 LLL 算法的误差控制条件 (3.10) 相比, 基于 PSLQ 算法的误差控制条件 (3.15) 要宽松一些. 特别地, (3.10) 中要求近似值 $\bar{\alpha}$ 与准确值 α 之间的误差控制在 $2^{-O(d^2+d\log N)}$ 以内, 而 (3.15) 只要求误差控制在 $2^{-O(d\log d+d\log N)}$ 以内. 由此, 所需的浮点数精度关于 d 的依赖程度便从二次降为拟线性. 这与 Bailey[20] 提出的经验公式相符: 对 d 维向量求解高度不超过 N 的整数关系至少需要输入的数据具有 $d\log_{10} N$ 位的十进制精度. 表 2 比较了几个代数数极小多项式近似重构的误差控制条件.

表 2 代数数 $\alpha = 1/(\sqrt[r]{2} + \sqrt[s]{3})$ 极小多项式近似重构中的误差控制 (n 为次数, N 为高度)

r	s	n	N	(3.10) 中的输入误差	(3.15) 中的输入误差
2	4	8	104	3.9537×10^{-67}	7.2936×10^{-48}
2	6	12	552	7.6741×10^{-134}	5.2144×10^{-88}
4	6	24	32364	1.0465×10^{-445}	1.9748×10^{-260}
4	8	32	823984	1.2404×10^{-765}	2.8489×10^{-438}
6	8	48	400286016	1.7439×10^{-1647}	5.8168×10^{-919}

4 进一步的讨论

最后讨论零误差计算中尚待进一步研究的几个问题, 并探讨零误差计算更多的潜在应用.

误差分析与控制 正如前面已提到的, 给定次数和高度上界的代数数能够进行零误差计算的一个必要条件是近似值和准确值的误差需满足 (3.2). 从渐近的意义上讲, 这要求误差控制在 $2^{-O(d\log d + d\log N)}$, 与基于 PSLQ 的零误差算法中的误差控制条件 (3.15) 一致. 这说明在渐近意义下, 基于 PSLQ 的代数数零误差计算的误差控制已是最优的. 然而, 对表 2 中 $(r,s) = (2,4)$ 的代数数, 使用计算机代数系统 Maple (2015 版) 中 `LLL` 函数, 在 `Digits := 18` 时便可以恢复出准确的极小多项式. 这说明 LLL 方法在误差控制条件上仍存在改进空间.

零误差算法的优化与分析 在包括文献 [18, 21, 22] 等在内的文献中, 通过 PSLQ 算法恢复代数数的极小多项式时, 都是采用 $(1, \alpha, \ldots, \alpha^n)$ 的顺序, 当代数数的实际次数大于当前的 n 时, 算法会对 $n+1$ 的情形进行计算. 此时, 之前计算的中间过程和结果都被丢弃了. 文献 [23] 证明了将输入的顺序调整为 $(\alpha^n, \alpha^{n-1}, \ldots, 1)$ 后, 若对当前的 n 算法没有输出极小多项式的系数, 则对 $n+1$ 情形的计算可以利用当前已有的计算结果, 从而将算法所需的复杂度上界降低因子 n. 对于 PSLQ 算法本身, 也有着众多的改进、推广和应用. 文献 [24] 给出了 PSLQ 算法的几个变种, 包括一个并行的 PSLQ 算法, 但没有给出终止性证明; 文献 [25] 改进了该并行算法, 并证明了改进算法的终止性. 文献 [26] 试图将 PSLQ 算法推广到代数数域上的代数整数关系计算. 在 PSLQ 的应用方面, 可以参见文献 [21]. 需要提及的是, 文献 [27] 给出了目前代数数恢复的 "世界纪录": 使用 64,000 位十进制成功恢复出次数达 512、高度达 10^{229} 的代数数的极小多项式. 但是, PSLQ 算法的位复杂度分析至今仍未完成. 文献 [3] 给出的数值 PSLQ 算法的扰动分析是解决这一问题的必要步骤. 但为完成浮点 PSLQ 算法的设计与分析, 还需要分析算法的舍入误差, 并研究如何有效控制算法中产生的整数矩阵的规模.

另外, 作为实现零误差计算的两种不同的算法, PSLQ 算法与 LLL 算法之间有着密切联系. 例如, 文献 [28] 从格约化的观点给出了 PSLQ 算法的全新解释. 文献 [29] 利用 PSLQ 中的 Bergman 交换规则设计出计算格的 LLL 约化基的算法. 在与零误差计算相关的 LLL 算法研究方面, 文献 [30] 对形如 (3.3) 的整数格进行了研究, 将 LLL 算法对该类型格所需的迭代次数关于维数 n 的依赖降低了一个因子 n. 因此, 对于一些 (带结构的) 特殊格, LLL 格约化算法的效率也还有改进空间.

零误差计算范围的扩展 本文重点讨论的是从数的浮点近似值获得准确值或准确表示的方法, 但也可以考虑更多数据类型和更多的近似类型. 例如, 文献 [7, 8, 31, 32] 讨论了在模算术意义下的有理数近似重构问题, 而文献 [33, 34] 在模算术的意义下考虑了有理向量的近似重构问题. 因此, 一个自然的问题是对代数数的零误差计算能否扩展到代数数向量的零误差计算?

将零误差计算从代数数扩展到更广泛的对象也是一个有趣的课题. 例如, 有理函数平方和表示中的零误差计算问题 [35]、整系数指数多项式的根如何进行零误差计算的问题 [36]、文献 [37] 给出的关于 π 和 e 的诸多等式的零误差计算问题, 这类研究也是 "实验数学" [38,39] 和 "符号 - 数值混合计算" [40] 的核心课题. 更一般地, 能否在抽象距离空间中来研究零误差计算问题? 能否给出一个算法支持抽象一致离散集合的零误差计算, 从而对前面提及的各种情形给出一个统一的解决方案, 也值得进一步探究.

致谢 感谢吴文渊研究员和两位匿名审稿人对本文提出的有益建议.

参考文献

1 Xia D, Wu Z, Yan S, et al. Real Variable Function and Functional Analysis (II) (in Chinese), 2nd ed. Beijing: Higher Education Press, 2010 [夏道行, 吴卓人, 严邵宗, 等. 实变函数论与泛函分析 (下册). 第二版修订本. 北京: 高等教育出版社, 2010]
2 Zhang J, Feng Y. Obtaining exact value by approximate computations. Sci China Ser A, 2007, 50: 1361–1368
3 Feng Y, Chen J, Wu W. The PSLQ algorithm for empirical data. Math Comp, 2019, 88: 1479–1501
4 Lehmer D H. Euclid's algorithm for large numbers. Amer Math Monthly, 1938, 45: 227–233
5 Knuth D. The analysis of algorithms. In: Actes du Congrès International des Mathématiciens. Paris: Gauthier-Villars, 1970, 269–274
6 Schönhage A. Schnelle Berechnung von Kettenbruchentwicklungen. Acta Inform, 1971, 1: 139–144
7 Wang X, Pan V Y. Acceleration of Euclidean algorithm and rational number reconstruction. SIAM J Comput, 2003, 32: 548–556
8 Pan V Y, Wang X. On rational number reconstruction and approximation. SIAM J Comput, 2004, 33: 502–503
9 Mishra B. Algorithmic Algebra. New York: Springer, 1993
10 Kannan R, Lenstra A K, Lovász L. Polynomial factorization and nonrandomness of bits of algebraic and some transcendental numbers. Math Comp, 1988, 50: 235–250
11 Mignotte M. Some useful bounds. In: Computer Algebra: Symbolic and Algebraic Computation. Heidelberg: Springer, 1982, 259–263
12 Lenstra A K, Lenstra Jr H W, Lovász L. Factoring polynomials with rational coefficients. Math Ann, 1982, 261: 515–534
13 Chang X W, Stehlé D, Villard G. Perturbation analysis of the QR factor R in the context of LLL lattice basis reduction. Math Comp, 2012, 81: 1487–1511
14 Nguyen P Q, Stehlé D. Floating-point LLL revisited. In: Advances in Cryptology—Annual International Conference on the Theory and Applications of Cryptographic Techniques. Heidelberg: Springer, 2005, 215–233
15 Morel I, Stehlé D, Villard G. H-LLL: Using Householder inside LLL. In: Proceedings of the 2009 International Symposium on Symbolic and Algebraic Computation. New York: ACM, 2009, 271–278
16 van Hoeij M, Novocin A. Gradual sub-lattice reduction and a new complexity for factoring polynomials. In: Proceedings of the 2010 Latin American Symposium on Theoretical Informatics. Heidelberg: Springer, 2010, 539–553
17 Novocin A, Stehlé D, Villard G. An LLL-reduction algorithm with quasi-linear time complexity: Extended abstract. In: Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing. New York: ACM, 2011, 403–412
18 Ferguson H R P, Bailey D H, Arno S. Analysis of PSLQ, an integer relation finding algorithm. Math Comp, 1999, 68: 351–369
19 Bergman G M. Notes on Ferguson and Forcade's generalized Euclidean algorithm. http://math.berkeley.edu/~gbergman/papers/unpub/FF_Euc.pdf, 1980
20 Bailey D H. Integer relation detection. Comput Sci Eng, 2000, 2: 24–28
21 Borwein J M, Lisoněk P. Applications of integer relation algorithms. Discrete Math, 2000, 217: 65–82
22 Chen J, Feng Y, Qin X, et al. Reconstructing minimal polynomial from approximate algebraic numbers (in Chinese). J Systems Sci Math Sci, 2011, 31: 903–912 [陈经纬, 冯勇, 秦小林, 等. 代数数极小多项式的近似重构. 系统科学与数学, 2011, 31: 903–912]
23 Feng Y, Chen J, Wu W. Incremental PSLQ with application to algebraic number reconstruction. ACM Commun Comput Algebra, 2014, 47: 112–113
24 Bailey D H, Broadhurst D J. Parallel integer relation detection: Techniques and applications. Math Comp, 2001, 70: 1719–1736

25 Feng Y, Chen J, Wu W. Two variants of HJLS-PSLQ with applications. In: Proceedings of the 2014 Symposium on Symbolic-Numeric Computation. New York: ACM, 2014, 88–96
26 Skerritt M P, Vrbik P. Extending the PSLQ algorithm to algebraic integer relations. In: From Analysis to Visualization. Cham: Springer, 2020, 407–421
27 Bailey D H, Borwein J M, Kimberley J S, et al. Computer discovery and analysis of large Poisson polynomials. Experiment Math, 2017, 26: 349–363
28 Chen J, Stehlé D, Villard G. A new view on HJLS and PSLQ: Sums and projections of lattices. In: Proceedings of the 38th International Symposium on Symbolic and Algebraic Computation. New York: ACM, 2013, 149–156
29 Chen J, Feng Y, Wu W. Reducing lattice bases with Bergman exchange. In: Proceedings of the IEEE 9th International Conference on Communication Software and Networks (ICCSN), Volume II. Piscataway: IEEE, 2017, 630–634
30 Chen J, Stehlé D, Villard G. Computing an LLL-reduced basis of the orthogonal lattice. In: Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation. New York: ACM, 2018, 127–133
31 Böhm J, Decker W, Fieker C, et al. The use of bad primes in rational reconstruction. Math Comp, 2015, 84: 3013–3027
32 Abbott J. Fault-tolerant modular reconstruction of rational numbers. J Symbolic Comput, 2017, 80: 707–718
33 Olesh Z, Storjohann A. The vector rational function reconstruction problem. In: Computer Algebra 2006: Latest Advances in Symbolic Algorithms. Singapore: World Scientific, 2007, 137–149
34 Bright C, Storjohann A. Vector rational number reconstruction. In: Proceedings of the 36th International Symposium on Symbolic and Algebraic Computation. New York: ACM, 2011, 51–58
35 Kaltofen E L, Li B, Yang Z, et al. Exact certification in global polynomial optimization via sums-of-squares of rational functions with rational coefficients. J Symbolic Comput, 2012, 47: 1–15
36 Moreno C J. The zeros of exponential polynomials (I). Compos Math, 1973, 26: 69–78
37 Bailey D H. Numerical results on the transcendence of constants involving π, e, and Euler's constant. Math Comp, 1988, 50: 275–281
38 Bailey D H, Borwein J M, Calkin N J, et al. Experimental Mathematics in Action. New York: CRC Press, 2007
39 Bailey D H, Borwein J M. Exploratory experimentation and computation. Notices Amer Math Soc, 2011, 58: 1410–1419 [Bailey D H, Borwein J M. 探索性实验和计算. 陆汝钤, 译. 数学译林, 2012, 31: 1–14]
40 Zhi L. Symbolic-numeric algorithms for computing validated results. In: Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation. New York: ACM, 2014, 25–26

On zero-error computation

Yong Feng & Jingwei Chen

Abstract It is important both in theory and in practice to study how to obtain exact results via numeric computation, which we call zero-error computation. In this paper, we firstly indicate which kind of numbers are suitable for zero-error computation: One can compute the exact value from its approximate values for every element in a uniformly discrete set, in which there exists a nonzero separation bound between two distinct elements. Based on this observation, we give such a separation bound for algebraic numbers, which can be seen as a necessary condition on error control for zero-error computation of algebraic numbers. However, this condition may not be sufficient, depending on different algorithms. For the PSLQ (partial-sum-LQ-decomposition)-based algorithm, we give a sufficient condition on the precision that is quasi-linear in the degree of the algebraic number to be recovered, while the corresponding condition for the LLL (Lenstra-Lenstra-Lovász)-based algorithm is quadratic. We also suggest several potential research areas in the future.

Keywords **zero-error computation, integer relation, error-controlling, LLL algorithm, PSLQ algorithm**

MSC(2010) **11A05, 11Y16, 68W40**

doi: 10.1360/SSM-2019-0336