arXiv:2301.05847v1 [cs.SC] 14 Jan 2023

混合三角多项式的有界与无界实根隔离

Rizeng Chen[a]、Haokun Li[a]、Bican Xia[a]、Tianqi Zhao[a]、Tao Zheng[b]

[a]北京大学数学科学学院，中国北京[b]中国科学院数学与系统科学研究院数学机械化重点实验室，中国北京

摘要

混合三角多项式（MTP）是形如 $f(x,$ 正弦 $x,$ 余弦 $x)$ 且满足 $f \in \mathbb{Q}[x_1, x_2, x_3]$ 的函数。本文提出并实现了一种“隔离”MTP 所有实根的算法。该算法自动将实根分为两部分：一部分是在区间 $[\mu_-, \mu_+]$ 内有限多个“有界”根，另一部分是在 $\mathbb{R}\backslash[\mu_-, \mu_+]$ 内可能可数无限多个“周期”根。对于有界根，算法返回隔离区间及相应的重数；对于周期根，算法返回有限个两两不相交的小区间 $I_i \subset [-\pi, \pi]$、整数$c_i > 0$ 以及根重数多重集 $\{m_{j,i}\}_{j=1}^{c_i}$，使得任何周期根$t > \mu_+$ 都属于集合 $(\sqcup_i \cup_{k\in\mathbb{N}} (I_i + 2k\pi))$，并且任何区间 $I_i + 2k\pi \subset (\mu_+, \infty)$ 分别包含恰好 c_i 个重数为 $m_{1,\,i}$，……，$m_{c_i,\,i}$ 的周期根。实验表明，该算法有效且高效。此外，用于隔离 $[\mu_-, \mu_+]$ 中根的方法也适用于任何其他有界区间。该算法利用弱傅里叶序列技术，逐周期处理区间，而不缩放坐标，从而保持序列长度较短。这些新方法很容易修改，用以判定形式为 $(-\infty, a)$ 或 (a, ∞) 的无界区间中是否存在根，或是否存在无穷多个根，其中 $a \in \mathbb{Q}$。

关键词：混合三角多项式、实根隔离、周期根、无界根、混合三角多项式不等式

电子邮箱地址：xiaxueqaq@stu.pku.edu.cn (Rizeng Chen), haokunli@pku.edu.cn (Haokun Li), xbc@math.pku.edu.cn (BicanXia), zhaotq@pku.edu.cn (Tianqi Zhao), taozheng@amss.ac.cn (Tao Zheng)

提交至《符号计算杂志》的预印本 *2023*年*1*月*18*日

1. 引言

实根隔离被视为一种符号方法，用于计算一元多项式或函数的实根。多项式的实根隔离在计算实代数几何中具有基础性的重要意义（例如，它是柱面代数分解的一种路由子算法 [1]），而且计算机代数系统中有许多高效算法和流行工具，例如 Maple 和 Mathematica，可用于隔离多项式的实根。

此外，也有一些关于非多项式函数实根隔离的研究：在 [2], 中，区间算术已被用于隔离一大类解析函数的实根。该算法主要考虑判定某个区间内是否排除或包含一个根。近年来，Strzebo´nski 提出了一种针对单变量指数-对数-反正切函数的实根隔离过程 [3, 4]。该过程基于弱傅里叶序列和半傅里叶序列技术。同时，Strzebo´nski 利用伪导数的概念，研究了所谓温和初等函数的实根隔离问题 [5]。Strzebo´nski 的这些算法的完备性依赖于 Schanuel 猜想。在 [6, 7], 为涉及整系数多项式和某个特定解析超越函数 trans(x) 的一类一阶逻辑句子开发判定过程时，Achatz、McCallum 和 Weispfenning 提出了一种递归算法，用于隔离 trans(x) 中某类广义整系数多项式的实根：f(x, trans(x))（其中 $f(x, y)$ 是一个给定的多项式，其变量为 y，系数属于关于某个特定整系数多项式 $d(x)$ 的幂构成的分式环 $\mathbb{Z}[x]$ ），该算法采用了伪微分和罗尔定理，遵循 [8], 的思路，并运用了 Lindemann [9] 的一项经典结果。他们的判定方法可直接应用于的超越函数包括 exp(x)、ln(x) 和反正切(x)。当 trans(x) $=$ exp(x) 时，该判定过程已实现，且相应的实根隔离算法被认为比 [10] 中提出的算法更高效。受 [6] 和 [11], 的启发，Xu等人研究了一类扩展多项式公式的带量词指数多项式公式 [12]。其中所用的指数多项式隔离算法改编自 [6]。在 [13], ，Huang等人开发了一些算法，用于隔离一类称为多重幂函数的函数的正根；而在 [14], ，Wang 和 Xu 专注于隔离

某类根式的实根。

混合三角多项式 (MTP) 是指具有以下形式的函数$f(x,$ 正弦 $x,$ 余弦 $x)$, 其中 f是一个具有有理系数的三元多项式。本文提出一种算法，用于“隔离”此类 MTP 的所有实根（它们可能有无穷多个）。虽然所有反三角函数都是指数-对数-反正切函数，但 MTP 并非如此。事实上，由于指数-对数-反正切函数只有有限个实根（[4, 命题 4]），任何具有无穷多个实根的 MTP 都不是指数-对数-反正切函数（例如，正弦 x、余弦 x、正弦$x+1$等）。另一方面，由于变量遍历整个 $\mathbb{R}$，MTP 通常不是温和初等函数。因此，我们的结果不属于 Strzebo´nski 的研究范畴。不过，在第4节中，当隔离 MTP 在有界区间内的实根时，我们利用了 [3, 4] 中引入的函数的弱傅里叶序列（也可以使用 [5] 中的伪导数技术，因为任何限制在有界区间上的 MTP 都是温和的）。尽管如此，我们不需要 Schanuel 猜想，因为我们只考虑单变量三角函数。请注意， [4] 和 [7] 中使用的方法都适用于隔离以下形式的函数的实根： f(反正切 $x,$ x)，其中 f是一个具有有理系数的二元多项式。这等价于隔离函数 f(ˆ$x,$ 正切 ˆx) 的实根，其中 ˆ$x \in (-\frac{\pi}{2},$ $\frac{\pi}{2}$)（参见例如 [7,第 6]节）。本文的目标是隔离 MTP 的所有实根，这是一个非平凡的推广。

自动证明涉及三角函数的不等式这一主题与本文主题密切相关。近年来，由于这类不等式在工程问题和科学的不同分支中的应用，越来越受到关注 [15]。通常，“MTP不等式”指形如 $g(x) > 0$ 或 $g(x) \geq 0$的不等式，其中 g 是一个 MTP。Mortici 在 [16]中提出的所谓“自然方法”，通过泰勒多项式近似正弦 x和余弦x，并将证明 MTP不等式的问题转化为证明某些多项式（或有理函数）不等式的问题。该方法推动了一系列后续研究 [17, 18, 19, 20, 21], ，其中大量涉及三角函数的不等式得以自动求解。之后，在 [15], 中，Chen 和 Liu 给出了一个证明 MTP不等式的完整算法，并首次系统地讨论了算法的终止性。然而，这些算法只能证明有界区间上的 MTP不等式（例如，$(0, \frac{\pi}{2})$ 或 $(0, \frac{\pi}{4}]$，等等）。直到最近，Chen 和 Ge [22]

尝试开发一种程序，以自动证明无界区间 $(0,\infty)$ 上的 MTP不等式。遗憾的是，我们发现其中存在一些错误。因此，证明 MTP不等式以及在无界区间中隔离实根的问题至今仍未解决。

本文提出了一种用于“隔离”MTP 所有实根的算法（算法 7）。该算法会自动将实根分为两部分：一部分是在区间 $[\mu_-,\mu_+]$with $\mu_- < 0 < \mu_+$ 中有限个“有界”根，另一部分则是在 $\mathbb{R}\backslash[\mu_-,\mu_+]$ 中可能有可数无穷多个“周期”根。对于有界根，算法会返回若干隔离区间以及每个根的重数。对于大于 μ_+ 的周期根，算法会返回有限个两两不相交的小区间 $I_i \subset [-\pi,\pi]$、整数 $c_i > 0$ 和根重数多重集$\{m_{j,i}\}_{j=1}^{c_i}$，使得任意周期根 $t > \mu_+$ 都属于集合 $(\sqcup_i \cup_{k\in\mathbb{N}}(I_i + 2k\pi))$，并且任意区间 $I_i + 2k\pi \subset (\mu_+, \infty)$ 恰好包含 c_i 个周期根，其重数分别为 $m_{1,\,i}$，…，$m_{c_i,\,i}$，。对于小于 μ_-的周期根，算法也会返回类似结果。该算法使用 `Maple2021` 实现。我们测试了大量示例（包括文献中的示例和随机生成的示例），以展示其有效性和效率。

特别是，这是首个能够判定给定 MTP 是否有无穷多个实根的算法：当且仅当算法返回至少一个周期根时，MTP 才有无穷多个实根。当然，该算法还能够判定 MTP 是否有实根（若有实根，我们可以计算出所有实根的重数）。这使我们能够证明或反驳关于有理数集 $\mathbb{R}$ 上的 MTP 不等式。我们的结果还首次表明，MTP 的根在“周期”中的“分布”$(-\pi,\ \pi]+2k\pi$，且 $k > 0$ (分别，$k < 0$) 足够大）具有相同的模式，这似乎是关于 MTP 实根的一条基本规律。

此外，用于在 $[\mu_-,\mu_+]$ 中隔离根的方法同样适用于任何其他有界区间。该方法利用弱傅里叶序列技术，逐周期处理各区间，而不缩放坐标（参见例如 [22, 算法4.1，第 2] 步），从而保持序列长度较短。一个有趣的问题是，比较伪导数技术与弱傅里叶技术在有界区间内隔离 MTP 实根时的效率；这超出了本文的范围。

作为直接应用，我们在论文末尾简要讨论如何修改我们提出的新方法，以证明或证伪在形式为 (a,b), $[a,b)$, $(a,b]$, $[a,b]$, 且满足 $a,b\in\mathbb{Q}\cup\{\pm\infty\}$ 的区间上的 MTP不等式。

本文其余部分安排如下。下一节介绍问题以及一些将不时用到的基本结果。第3节介绍用于“隔离”周期根的算法，第4节则给出计算有界根隔离区间的过程。第5节介绍用于“隔离”MTP所有实根的主算法。第6节提供实验结果，以表明该算法的有效性和效率。最后，第7节总结全文，并就证明 MTP不等式这一主题给出一些注。

2. 问题

2.1. 一些基本结果

分别用 $\mathbb{N}$、$\mathbb{Z}$、$\mathbb{Q}$、$\mathbb{R}$和 $\mathbb{C}$表示自然数集、整数集、有理数集、实数集和复数集。对于任意子集 $A \subset \mathbb{R}$，令 $A_{>0} = \{a \in A \mid a > 0\}$。符号 $A_{\geq 0}$、 $A_{\leq 0}$和 $A_{<0}$的含义类似。对于任意 $\mathbb{A} \in \{\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}\}$，$\mathbb{A}[x_1, \ldots, x_m]$表示变量 x_1、.....、x_m上的相应多项式环，其系数属于 $\mathbb{A}$。对于任意 $f \in \mathbb{A}[x_1, \ldots, x_m]$，$\mathsf{deg}(f, x_i)$、$\mathtt{LC}(f, x_i)$和 $\mathtt{discrim}(f, x_i)$分别表示 f 关于 x_i的次数、首项系数和判别式。此外，$\mathtt{tdeg}(f, [x_{i_1}, \ldots, x_{i_t}])$表示变量子序列$[x_{i_1}, \ldots, x_{i_t}]$的总次数。

本文研究的问题是求解以下方程的实根：

$$f(x, \sin x, \cos x) = 0, \text{ where } f \in \mathbb{Q}[x, y, z]. \tag{2.1}$$

下面给出一些基本观察。

引理 2.1. 对于任意非零的 $f \in \mathbb{Q}[x, y, z]$，$f(x,$ 正弦 $x,$ 余弦 $x)$的实根集在 $\mathbb{R}$ 中没有聚点。

证明。 复值函数 $h(z) = f(z,$ 正弦 $z,$ 余弦 $z)$ 在$\mathbb{C}$上解析。根据定义，$h|_{\mathbb{R}} = f(x,$ 正弦 $x,$ 余弦 $x)$。如果 f 的实根集在 $\mathbb{R}$中有聚点，那么 h 的复根集在 $\mathbb{C}$ 中也有聚点，这表明 $h \equiv 0$ 在 $\mathbb{C}$ 上恒为零。但 h 不可能在 $\mathbb{C}$ 上恒等于 0，因为 f 在 $\mathbb{R}$ 上并非如此。 □

引理 2.2. 对于任意 $q(x) \in \mathbb{Q}[x]$，“$r\pi$ 是任意 $r \in \mathbb{Q} \setminus \{0\}$ 的 $q(x)$ 实根”当且仅当“$q(x)$ 是关于 $\mathbb{Q}[x]$ 的零多项式”。

证明. “若”部分显然。“仅若”部分由以下事实得出：对于任意 $r \in \mathbb{Q} \setminus \{0\}$，$r\pi$ 都是超越的。 □

定理 2.3。 *[23, 第 2 章。3]* 若 $\alpha \in \mathbb{C}$ 且 $\alpha \neq 0$，则每个数对$(\alpha$, 正弦 $\alpha)$、$(\alpha$, 余弦$\alpha)$、$(\alpha$, 正切 $\alpha)$、$(\alpha$, 余切$\alpha)$中至少有一个超越数。

□

定义 2.4。 对于非零的 $g \in \mathbb{Q}[x, y]$，定义其倒数多项式 $\mathcal{R}(g)$ 为 $g(x, \frac{1}{y})$ $y^{\deg(g,y)}$。此外，令 $\mathcal{R}(0) = 0$。

由定义显然可知

1) $\mathcal{R}(g_1 g_2) = \mathcal{R}(g_1)\mathcal{R}(g_2)$，对任意 g_1， $g_2 \in \mathbb{Q}[x, y]$；2) 若 $g = \sum_{i=0}^{\deg(g,y)} a_i(x)y^i$ 且 $a_0(x) \neq 0$ (即， $y \nmid g$)，则 $\mathcal{R}(\mathcal{R}(g)) = g$；3) 对任意非零 g， y 不整除 $\mathcal{R}(g)$。由此得到 $\mathcal{R}$ 的一些重要性质如下。

命题 2.5. 假设对于任意 $c \in \mathbb{Q}$， $g \in \mathbb{Q}[x, y]$ 和 $g \neq cy$。那么，“g在 $\mathbb{Q}[x, y]$ 中不可约”蕴含“$\mathcal{R}(g)$ 在 $\mathbb{Q}[x, y]$中不可约”。

证明。对 g 的总次数作归纳：若 $\mathtt{tdeg}(g, [x, y]) = 0$，则结论显然成立，因为 g 不可约性不成立。假设$\mathtt{tdeg}(g, [x, y]) = n > 0$，并且对于任意总次数小于 n 的多项式 $(\neq cy)$，该蕴含关系成立。设 $\mathcal{R}(g)$ 可约，且其因式分解为 $c_0 h_1^{p_1} \cdots h_t^{p_t}$，其中 h_i 不可约， $\sum_{j=1}^{t} p_j \geq 2$ 且 $c_0 \in \mathbb{Q}\setminus\{0\}$。此外，对于任意非零 $c \in \mathbb{Q}$，每个 $h_i \neq cy$ 都成立，因为 $y \nmid \mathcal{R}(g)$。由于 g不可约，且对于任意 $c \in \mathbb{Q}$， $g \neq cy$，因此 $y \nmid g$。于是 $g = \mathcal{R}(\mathcal{R}(g)) = c_0 \mathcal{R}(h_1)^{p_1} \cdots \mathcal{R}(h_t)^{p_t}$。根据归纳假设，每个 $\mathcal{R}(h_i)$ 都不可约。由于 $\sum_{j=1}^{t} p_j \geq 2$， g可约，这就产生了矛盾。

□

对于任意总次数为正的多项式 $g \in \mathbb{Q}[x, y]$，若 g 的因式分解为 $cg_1 \cdots g_t$，其中 $c \in \mathbb{Q}\setminus\{0\}$、 $t \in \mathbb{Z}_{\geq 1}$ 均为不可约多项式，且对于任意 $1 \leq i \neq j \leq t$， g_i 与 g_j 互素，则称 g 为无平方因子多项式。根据我们的定义，任何常数多项式都不是无平方因子的。

命题 2.6。 假设对于任意 $c \in \mathbb{Q}$， $g \in \mathbb{Q}[x, y]$ 且 $g \neq cy$。那么，“g 是无平方因子的”蕴含“$\mathcal{R}(g)$ 是无平方因子的”。

证明。 将 g因式分解为 $c_0 y^{\ell} h_1 \cdots h_t$，其中 $c_0 \in \mathbb{Q}$ 非零，或为 1（分别对应于情形 $y \nmid g$和 $y \mid g$）， $t \geq 1$（因为 $g \neq cy$和$g \notin \mathbb{Q}$），且每个 h_i 都不可约。

首先，$\mathcal{R}(g) = c_0\mathcal{R}(h_1)\cdots\mathcal{R}(h_t)$，因为 $\mathcal{R}(y^\ell) = 1$。其次，对每个 h_i，任意 $c \in \mathbb{Q}$ 都有$h_i \neq cy$，因为 $y \nmid h_i$。因此，根据命题 2.5，$\mathcal{R}(h_i)$不可约。最后，对于 $1 \leq i \neq j \leq t$，$\mathcal{R}(h_i)$和 $\mathcal{R}(h_j)$互素。假设它们不互素，那么对于某个 $c' \in \mathbb{Q}\backslash\{0\}$，$\mathcal{R}(h_i) = c'\mathcal{R}(h_j)$。由于 $y \nmid h_i$和 $y \nmid h_j$，$h_i = \mathcal{R}(\mathcal{R}(h_i)) = c'\mathcal{R}(\mathcal{R}(h_j)) = c'h_j$，这与$h_i$ 和 h_j 在 g的因式分解中互素这一事实矛盾。 □

2.2. 将问题分解为子问题

因为

$$\begin{cases} \sin x &= \frac{2\tan\frac{x}{2}}{1+\tan^2(\frac{x}{2})} \\ \cos x &= \frac{1-\tan^2\frac{x}{2}}{1+\tan^2(\frac{x}{2})} \end{cases}, \quad x \neq 2k\pi+\pi \ (k \in \mathbb{Z}),$$

方程（2.1）等价于

$$\begin{cases} f(x, \sin x, \cos x) = 0, & x = 2k\pi+\pi, \qquad (2.2a) \\ f(x, \frac{2\tan\frac{x}{2}}{1+\tan^2(\frac{x}{2})}, \frac{1-\tan^2(\frac{x}{2})}{1+\tan^2(\frac{x}{2})}) = 0, & x \in (2k\pi-\pi, 2k\pi+\pi), \qquad (2.2b) \end{cases}$$

其中，方程（2.2a）等价于 $f(x, 0, -1) = 0$，$x = (2k+1)\pi$。根据引理 2.2，$(2k+1)\pi$是 $f(x, 0, -1)$ 的一个根，当且仅当 $f(x, 0, -1)$ 是 $\mathbb{Q}[x]$ 中的零多项式。通过约去分母，我们定义如下多项式：

$$g(x,t) = f(2x, \frac{2t}{1+t^2}, \frac{1-t^2}{1+t^2}) \cdot (1+t^2)^{\mathtt{tdeg}(f,[y,z])} \in \mathbb{Q}[x,t]. \qquad (2.3)$$

由于 $1+$正切$^2(\frac{x}{2}) > 0$，求解方程（2.2b）的实根等价于求解方程

$$g(x, \tan x) = 0, \ x \in (k\pi - \frac{\pi}{2}, k\pi + \frac{\pi}{2}). \qquad (2.4)$$

例 2.7。 考虑以下 *MTP*

$$f(x, \sin x, \cos x) = x\cos^4 x - 2\sin x\cos^3 x + \frac{4}{5}\sin^3 x.$$

由方程 (2.3)，可得

$$g(x,t) = 10xt^8 + 20t^7 - 40xt^6 - 28t^5 + 60xt^4 + 92t^3 - 40xt^2 - 20t + 10x.$$

在方程 (2.4) 中，当 k 在 $\mathbb{Z}$ 中变化时，我们必须处理无穷多个区间 $(k\pi-\frac{\pi}{2},\ k\pi+\frac{\pi}{2})$。幸运的是，下面的命题 2.8 和推论 2.11，以及下一节中的定理 3.6，将使我们不必这样做。

命题 2.8. 设非零多项式 $g \in \mathbb{Q}[x,y]$, $\delta > 0$ 和 $U \subset \mathbb{R}$ 构成一个有界集。若对任意 $y \in U$，都有 $|\mathrm{LC}(g,x)| > \delta$，则存在一个整数$N > 0$，使得 $x > N$ 蕴含 $\forall y \in U$, $g(x,y) \neq 0$。

证明。设 $\deg(g,x) = n$，且 $a_i(y) \in \mathbb{Q}[y]\,(i = 0,\ldots,n)$ 是 g 关于 x 的系数。由于 U 有界，存在 $B_i > 0$，使得对任意值都有 $|a_i\ y\ | < B_i$ $y \in U$ x $\delta|x|^n >$ ( )。当 足够大时，有 。$\sum_{i=0}^{n-1} B_i|x|^i$ $y \in U$
因此，对任意 ,

$$|a_n(y)x^n| > \delta|x|^n > \sum_{i=0}^{n-1} B_i|x|^i \geq |\sum_{i=0}^{n-1} a_i(y)x^i|.$$

这表明，对任意 $y \in U$，都有 $g(x,y) \neq 0$。 □

根据命题 2.8，$g(x,\ \tan\ x)$ 的每个“周期性”实根处的正切函数值，都出现在 $\mathrm{LC}(g,\ x)$ 的某个实根附近。为了定位这些“周期性”实根，我们引入以下定义。

定义 2.9。对于任意满足 $\deg(g,y) \geq 1$ 的 $g \in \mathbb{Q}[x,y]$，设 I 为一个由开区间组成的有限列表 $[(a_0,b_0)\ldots,(a_{s+1},b_{s+1})],$，其中 $s \in \mathbb{Z}_{\geq 0}$, $a_0 = -\infty, b_{s+1} = +\infty$, $b_0 < 0$, $a_{s+1} > 0$, $a_j \in \mathbb{Q}\ (1 \leq j \leq s+1)$ 和 $b_j \in \mathbb{Q}\ (0 \leq j \leq s)$，使得

(i) 对于每个 $j\ (1 \leq j \leq s)$，$a_j < b_j$ 和 $b_j \leq a_{j+1}$,

(ii) 对于 $\mathrm{LC}(g,x)$ 的每个实根，都存在一个包含该根的区间 $(a_j,b_j)\ (1 \leq j \leq s)$（($\mathrm{LC}(g,x)$ 在 (a_0,b_0) 和 (a_{s+1},b_{s+1})) 中没有根），

(iii) 每个区间$(a_j,b_j)\ (1 \leq j \leq s)$ 恰有一个 $\mathrm{LC}(g,x)$的实根。

我们称 I 为 g的潜在周期区间集。

一般而言，如果我们将(iii) 中的“恰好”替换为“至多”，就称 I 为 g 的一般潜在周期区间集 。

示例 2.10。 考虑例 *2.7* 中的多项式:

$$g(x,y) = 10xy^8 + 20y^7 - 40xy^6 - 28y^5 + 60xy^4 + 92y^3 - 40xy^2 - 20y + 10x.$$

我们有 $\mathtt{LC}(g,x) = 10y^8 - 40y^6 + 60y^4 - 40y^2 + 10$。它有两个实根 -1和1，它们的隔离区间分别为$(-\frac{215}{128}, -\frac{19}{32})$和$(\frac{19}{32}、\frac{215}{128})$。因此，

$$\left[\left(-\infty, -\frac{63}{16}\right), \left(-\frac{215}{128}, -\frac{19}{32}\right), \left(\frac{19}{32}, \frac{215}{128}\right), \left(\frac{63}{16}, +\infty\right)\right]$$

是 g 的一个潜在周期区间集。

推论 2.11。 设 $g \in \mathbb{Q}[x,y]$ 为满足 $\deg(g,y) \geq 1$, $[(a_0,b_0)\ldots,(a_{s+1},b_{s+1})]$ 的一般潜在周期区间集，且

$$U := \mathbb{R} \setminus \bigcup_{j=0}^{s+1}(a_i, b_i). \tag{2.5}$$

那么，存在一个正整数 N，使得

$$\forall x \in \mathbb{R}\ \forall y \in U\ (x > N \Rightarrow g(x,y) \neq 0).$$

证明。注意，$|\mathtt{LC}(g,x)|$ 在 $y \in U$ 上有下界，且 U 有界。因此，结论由命题 2.8 得出。 □

对于方程 (2.4) 中的多项式 $g \in \mathbb{Q}[x,y]$，令 U, a_j, b_j 与推论 2.11 中相同。取 $N_+, N_- \in \mathbb{Z}_{\geq 1}$，使得 $\forall x \in \mathbb{R}\ \forall y \in U\,(x > N_+ \Rightarrow g(x,y) \neq 0)$ 且 $\forall x \in \mathbb{R}\ \forall y \in U(x < -N_- \Rightarrow g(x,y) \neq 0)$。存在 $k_+ \in \mathbb{Z}_{\geq 0}$ 和 $k_- \in \mathbb{Z}_{\leq 0}$，使得它们分别满足 $k_+\pi - \frac{\pi}{2} < N_+ < k_+\pi + \frac{\pi}{2}$ 和$k_-\pi - \frac{\pi}{2} < -N_- < k_-\pi + \frac{\pi}{2}$。因此，方程 (2.4) 可等价地转化为以下三种情形:

(2.4a)

$$g(x, \tan x) = 0, \quad x \in (k\pi - \frac{\pi}{2}, k\pi + \frac{\pi}{2})$$

其中 $k_- \leq k \leq k_+$ $(k \in \mathbb{Z})$;

(2.4b)

$$g(x, \tan x) = 0, \quad \begin{aligned} & x \in (k\pi - \frac{\pi}{2}, k\pi + \arctan(b_0)) \\ & x \in (k\pi + \arctan(a_i), k\pi + \arctan(b_i)),\ i = 1, \ldots, s \\ & x \in (k\pi + \arctan(a_{s+1}), k\pi + \frac{\pi}{2}) \end{aligned}$$

其中 $k > k_+$ ($k \in \mathbb{Z}$);

(2.4c)

$$g(x, \tan x) = 0, \quad \begin{aligned} & x \in (k\pi - \frac{\pi}{2}, k\pi + \arctan(b_0)) \\ & x \in (k\pi + \arctan(a_i), k\pi + \arctan(b_i)),\ i = 1, \ldots, s \\ & x \in (k\pi + \arctan(a_{s+1}), k\pi + \frac{\pi}{2}) \end{aligned}$$

其中 $k < k_-$ ($k \in \mathbb{Z}$)。

求解 $k_- \le k \le k_+$ 的情形 (2.4a) 的一种可能方法，是回到原始 MTP $f(x,$ 正弦 $x,$ 余弦 $x)$，并考察区间 $(2k_-\pi - \pi, 2k_+\pi + \pi)$ 内的根。将 $f(x,$ 正弦 $x,$ 余弦 $x)$ 中的 x 替换为 ℓx，其中 $\ell \in \mathbb{Z}_{>0}$ 足够大，使得区间 $(2k_-\pi - \pi, 2k_+\pi + \pi)$ 内的根“收缩”为区间 $(-\pi/2, \pi/2)$ 内 $f(\ell x,$ 正弦 $\ell x,$ 余弦 $\ell x)$ 的根。然后，在 $f(\ell x,$ 正弦 $\ell x,$ 余弦 $\ell x)$ 中用正切函数统一正弦函数和余弦函数，得到某个 $\hat{}g(x,$ 正切 $x)$。最后，处理区间 $(0, \pi/4)$ 内 $\hat{}g(-x, -$ 正切 $x)$ 和 $\hat{}g(x,$ 正切 $x)$ 的根。我们在第4节中不采用这种方法，是因为 ℓ 可能很大，而且 $\hat{}g(x,$ 正切 $x)$ 中正切 x 的“次数”可能很高。相反，我们将按周期逐一处理情形 (2.4a)，以避免这个问题。

通过将 $g(x,$ 正切 $x)$ 替换为 $g(-x, -$ 正切 $x)$，最后一种情况可以归结为倒数第二种情况。下一节中，我们将重点讨论对应于情况 (2.4b) 的实根的渐近性质。

3. 根的渐近性质

本节将处理情形 (2.4b)，即隔离 $g(x,$ 正切 $x)$ 的“周期性”根，其中 g 是关于 $\mathbb{Q}[x, y]$ 的非零多项式。

3.1. 检测并定位周期根

$g(x,$ 正切 $x)$ 有可数无穷多个“周期性”根，因此无法用传统实根隔离方法定位。不过，可以借助一般潜在周期区间集（回顾定义 2.9）将这些“周期性”根划分为有限个簇。每个簇中的根呈周期性出现，这就是我们称它们为“周期性”根的原因。本小节提出一种算法，利用一般潜在周期区间集隔离这些“周期性”根。

首先，我们介绍解析可刻画的定义，以及 McCallum [24] 的著作中关于它的一项重要结果。

定义 3.1 ([24])。令 $\bar{x}$ 表示$(r-1)$元组$(x_1, \cdots, x_{r-1})$。若实数域上的 r 元多项式 $f(\bar{x}, x_r)$ 在子流形 S (通常为连通的) 上关于 $\mathbb{R}^{r-1}$解析可刻画，则

(*1*) f 的实代数簇中位于S 上方柱体 $S \times \mathbb{R}$ 内的部分，由若干个从 S到 $\mathbb{R}$的 $k \geq 0$ 解析函数$\theta_1 < \cdots < \theta_k$ 的图形之并组成，并且

(*2*) 存在正整数 $m_1, \cdots, m_k$，使得对每个 $a \in S$，根 $\theta_i(a)$ 在 $f(a, x_r)$ (视为关于 x_r 的一元多项式时) 的重数为 m_i。

此外，若一个 r 元多项式 $f(\bar{x}, x_r)$ 定义在 $\mathbb{R}$ 上，则称其在 $\mathbb{R}^{r-1}$ 的子集 S 上是次数不变的，如果对于 S 中的每个点 p， $f(p, x_r)$ (作为关于 x_r) 的一元多项式的总次数都相同。

命题 *3.2* ($[24, \text{定理 } 2]$) 。设 $r \geq 2$ 为整数，且 $f(\bar{x}, x_r)$ 是 $\mathbb{R}[\bar{x}, x_r]$ 中一个总次数为正的多项式。假设 $D(\bar{x}) := \texttt{discrim}(f, x_r)$ 是一个非零多项式。设 S 是 $\mathbb{R}^{r-1}$ 的一个连通子流形，满足 f 是次数不变的，并且在 S 上不恒等于零；此外， $D(\bar{x}) > 0$ 或 $D(\bar{x}) < 0$ 在 S 上。则 f 在 S 上是解析可刻画的。□

我们在推论 3.3 中重述一个众所周知的结果（该结果可由命题 3.2 推出），并在定义 3.4 中重新定义“根函数”的概念。

推论 3.3. 令 g 为有理数集 $\mathbb{Q}[x, y]$ 上的无平方因子多项式，满足 $\deg(g, y) \geq 1$，并令 M 为多项式 $\texttt{LC}(g, y) \cdot \texttt{discrim}(g, y)$ 的最大实根。那么，

1. g 在（$M, +\infty$）上解析可刻画，并且

2. 存在一些解析函数 $\theta_1(x) < \cdots < \theta_r(x)$ $(r \geq 0)$，它们定义在 $(M, +\infty)$ 上，使得对于任意 $x_0 \in (M, +\infty)$， $g(x_0, \theta_i(x_0)) = 0$ 和 $\theta_1(x_0)$ ..... $\theta_r(x_0)$ 都是 $g(x_0, y)$ 的实根。 □

定义 3.4. 沿用推论 *3.3* 中的符号说明。我们将这些解析函数 $\theta_1(x), \ldots, \theta_r(x)$ 称为 g 的终端根函数，并将 g 的终端根函数的起始点记为 M。

例 3.5. 考虑例 *2.10* 中的多项式 $g(x, y)$。我们有 $\mathtt{LC}(g, y) = -10x$，且 $\mathtt{discrim}(g, y) = -70368744177664000 \cdot (163840000x^8 + 261529600x^6 + 83912256x^4 - 13333725x^2 - 10857025)$。将它们的所有实根隔离后，我们看到 $\mathtt{discrim}(g, y)$ 的最大实根 M （位于 $(0,1)$ 内）是 g 的终端根函数的起始点。g 有两个终端根函数，它们定义在 $(M, +\infty)$ (见图 *1*) 上。

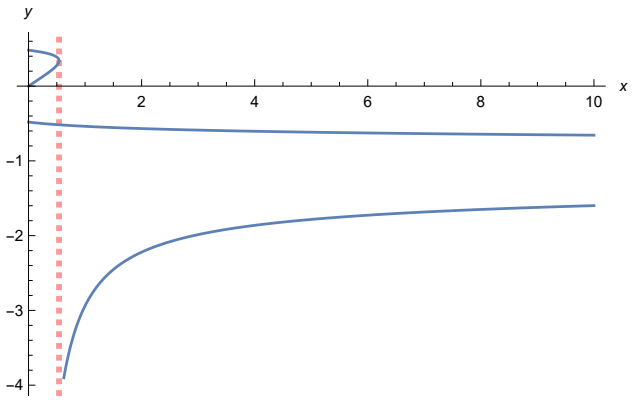

图 1: g 的终端根函数图。红色点线表示起始点 $x = M$ 的图像，终端根函数的图像以蓝色曲线绘出。

在下文的定理 3.6 中，我们计算一个界，以确保终端根函数定义良好（如推论 3.3 所述）；这些函数对于定位无界根至关重要。

定理 3.6. 设 g 是 $\mathbb{Q}[x, y]$ 中的一个无平方因子多项式，满足 $\deg(g, y) \geq 1$, $[(a_0, b_0), \ldots, (a_{s+1}, b_{s+1})]$ 是 g 的一般潜在周期区间集，k' 是一个自然数，使得 $M' = (k' + 1/2)\pi$ 是

终端根函数 g 的起始点 M 以及该多项式的所有实根的上界

$$\prod_{j=0}^{s} g(x, b_j) g(x, a_{j+1}),$$

且 $\theta_1(x) \ldots, \theta_r(x)$ 是 g 的终端根函数。则：

(1) 对于每个 $i\,(1 \le i \le r)$，存在 $j_i\,(0 \le j_i \le s+1)$，使得当 $x \in (M', +\infty)$ 时，$\theta_i(x) \in (a_{j_i}, b_{j_i})$。

(2) 对于 $g(x,$ 正切 $x)$ 的每个大于 M' 的实根 ξ，存在 $k > k'\,(k \in \mathbb{N})$ 和 $i\,(1 \le i \le r)$，使得

$$\xi \in (\arctan(a_{j_i}) + k\pi, \arctan(b_{j_i}) + k\pi)\,.$$

(3) 对于每个 $i\,(1 \le i \le r)$ 以及任意 $k > k'\,(k \in \mathbb{N})$，存在

$$\xi \in (\arctan(a_{j_i}) + k\pi, \arctan(b_{j_i}) + k\pi)$$

使得 正切 $\xi = \theta_i(\xi)$。

例3.7。 考虑例2.10中的多项式 $g(x, y)$。在例2.10中，我们计算了 $g(x, y)$ 的潜在周期区间集：

$$\left[\left(-\infty, -\frac{63}{16}\right), \left(-\frac{215}{128}, -\frac{19}{32}\right), \left(\frac{19}{32}, \frac{215}{128}\right), \left(\frac{63}{16}, +\infty\right)\right].$$

多项式的实根的隔离区间为

$$g(x, -\frac{63}{16}) \cdot g(x, -\frac{215}{128}) \cdot g(x, -\frac{19}{32}) \cdot g(x, \frac{19}{32}) \cdot g(x, \frac{215}{128}) \cdot g(x, \frac{63}{16})$$

分别是 $(-8, -7)$、$(-4, -3)$、$(-1, 0)$、$(0, 1)$、$(3, 4)$ 和 $(7, 8)$。在例 3.5 中，我们看到 g 的终端根函数的起始点 M 小于 1。因此，我们取 $k' = 3$，使得 $(k' + 1/2)\pi$ 是 M 及上述计算出的实根的一个上界。注意，当 $x \in ((k' + 1/2)\pi, \pm\infty)$ 时，g 的两个终端根函数都由 $(-\frac{215}{128}, -\frac{19}{32})$ 界定（见图2）。因此，对于每个 $k > 3 (k \in \mathbb{Z})$，$g(x,$ 正切 $x)$ 在 $(k\pi -$ 反正切 $215\ 128, k\pi -$ 反正切 $19\ 32)$ 中至少有两个实根。

注3.8。 对于每个区间（a_j, b_j）（$0 \le j \le s+1$），可能没有任何一个终端根函数 θ_i（x）在（a_j, b_j）内有界，此时

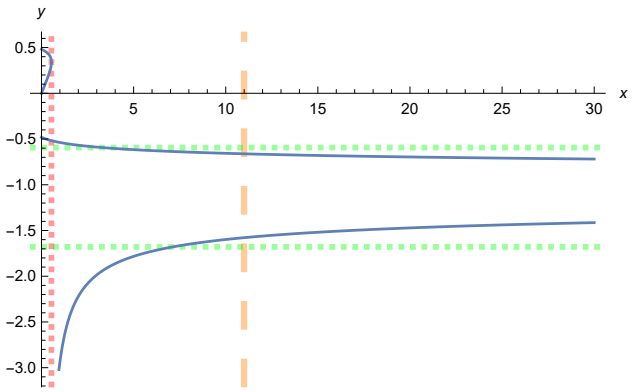

图2： g在（$-\frac{215}{128}$， $-\frac{19}{32}$）内有界的终端根函数图。红色点线和橙色虚线（均为竖直线）分别表示起始点 $x = M$ 和 $x =$ （$k' + 1/2$）π的图像。两条绿色点线（均为水平线）分别表示 $y = -\frac{215}{128}$和 $y = -\frac{19}{32}$的图像。当 $x >$（$k' + 1/2$）π时，终端根函数（绘制为蓝色曲线）均被区间（$-\frac{215}{128}$， $-\frac{19}{32}$）所界定。

$x > M'$。也可能有一个或多个终端根函数 $\theta_{i_1}(x) \ldots, \theta_{i_t}(x)$，满足 $t \in \mathbb{Z}_{\geq 1}$，在 $x > M'$时于(a_j, b_j)内有界。根据定理 *3.6*中的*(2)*和*(3)*，前一种情况下，集合$(M', \infty) \cap$正切$^{-1}(a_j, b_j)$中没有实根；后一种情况下，包含在(M', ∞)中的每个区间(反正切 $a_j + k\pi$, 反正切 $b_j + k\pi$)内至少有 t 个实根（在算法*1*第*11*行中，我们区分这两种情况）。将这些根记为 $x_k^{(1)} < \cdots < x_k^{(t'_k)}$，其中某个 $t'_k \geq t$。由于定理 *3.6*对于任意小“反正切长度”（即反正切 b_j − 反正切 a_j）的隔离区间(a_j, b_j)以及任意足够大的 M' 均成立，因此我们有

$$\lim_{k\to\infty} (x_k^{(t'_k)} - x_k^{(1)}) = 0.$$

这表明，不可能找到实数 c_1、 $c_2 \in$ （$-\pi/2$， $\pi/2$），使它们能够统一地隔离例如 $x_k^{(1)}$ 、 $x_k^{(2)}$ 和 $x_k^{(3)}$ （，如果 $t > 2$）。也就是说，不存在满足以下条件的 c_1、c_2 ：

$$x_k^{(1)} < c_1 + k\pi < x_k^{(2)} < c_2 + k\pi < x_k^{(3)}$$

对所有充分大的整数 k。这就是为什么我们只能在区间 (反正切a_j, 反正切 b_j) $+ k\pi$ 内对它们进行“隔离”，也就是说，每个 $x_k^{(\iota)}$ 都包含在其中。尽管如此，在下一小节中，我们将证明，对于

充分大的 k，这意味着我们可以计算距离零足够远的区间 (反正切 a_j, 反正切 b_j) $+ k\pi$ 内实根的确切数量 t 。

定理 3.6 的证明。(1) 注意， $g(x, a_j)$ $(1 \leq j \leq s+1)$ 和 $g(x, b_j)$ $(0 \leq j \leq s)$ 在 $(M', +\infty)$ 中没有根。于是，对于每个 i $(1 \leq i \leq r)$ 以及任意 $x \in (M', +\infty)$， $\theta_i(x) \notin \{b_0, a_1, b_1, \ldots, b_s, a_{s+1}\}$。根据命题 3.2， $\theta_i(x)$ 连续。因此，存在 j_i $(0 \leq j_i \leq s+1)$ 使得 $\theta_i((M', +\infty)) \subseteq (a_{j_i}, b_{j_i})$，或者存在 j_i $(0 \leq j_i \leq s)$ 使得 $\theta_i((M', +\infty)) \subseteq (b_{j_i}, a_{j_i+1})$。根据推论 2.11，当 x 充分大时，$\theta_i(x) \in \cup_{j=0}^{s+1} (a_j, b_j)$。因此，对于某个 $0 \leq j_i \leq s+1$， $\theta_i((M', +\infty)) \subseteq (a_{j_i}, b_{j_i})$。

(2) 若 ξ 是 $g(x,$ 正切 $x)$ 的实根，且满足 $\xi > M'$，则存在 i $(1 \leq i \leq r)$，使得 $\theta_i(\xi) =$ 正切 ξ。由 (1)， $\theta_i((M', +\infty)) \subseteq (a_{j_i}, b_{j_i})$。因此，正切 $\xi \in (a_{j_i}, b_{j_i})$。于是，存在 k $(k > k')$，使得 $\xi \in$ (反正切 $(a_{j_i}) + k\pi$，反正切 $(b_{j_i}) + k\pi$)。

(3) 由 (1)，对于每个 i $(1 \leq i \leq r)$， $\theta_i((M', +\infty)) \subseteq (a_{j_i}, b_{j_i})$。因此，对于任意 k $(k > k')$，我们有

$$\theta_i(\arctan(a_{j_i}) + k\pi) > a_{j_i} = \tan(\arctan(a_{j_i}) + k\pi),$$
$$\theta_i(\arctan(b_{j_i}) + k\pi) < b_{j_i} = \tan(\arctan(b_{j_i}) + k\pi).$$

因此，存在 $\xi \in$ (反正切 $(a_{j_i}) + k\pi$，反正切 $(b_{j_i}) + k\pi$)，使得正切 $\xi - \theta_i(\xi) = 0$。
□

定义 3.9。 设 g, k' 与定理 *3.6* 中相同，且 $\theta(x)$ 是 $g(x, y)$ 的一个终端根函数。我们将集合 $\{x_0 \in ((k'+1/2)\pi, +\infty) \mid \theta(x_0) =$ 正切 $x_0\}$ 的一簇周期根 $g(x,$ 正切 $x)$。

综上所述，我们提出算法1，以隔离 $g(x,$ 正切 $x)$ 的“周期”根。给定一个次数为 ≥ 1 的无平方因子多项式 $g \in \mathbb{Q}[x, y]$，其 deg(g, y)，以及 g 的一般潜在周期区间集 I，算法1计算出一个自然数 k' 和一组数对 $((x_i^-, x_i^+), c_i)$，使得对于任意 $k > k'$ $(k \in \mathbb{N})$， $g(x,$ 正切 $x)$ 在区间 $(x_i^- + k\pi, x_i^+ + k\pi)$ 内至少有 c_i 个实根。

定理3.10。 算法*1*正确。

证明。 算法1的执行过程基于定理 3.6。由定理 3.6 的证明可知，算法1是正确的。 □

算法1：IsolatingFarZero

输入 ： g， $\mathbb{Q}[x,y]$ 中的一个无平方因子多项式，使得 $\deg(g,\ y)\ \geq 1$;
$I=[(a_0,\ b_0)\ldots,\ (a_{s+1},\ b_{s+1})]$，一个由 g构成的一般潜在周期区间集。
输出： k'，一个自然数;
ret，一个有限对集 $((x_j^-,\ x_j^+)^{c_j})$，使得对于任意 $k>k'$ $(k\in\mathbb{N})$，都有 AT $LEAST$ c_i 在区间（$x_j^-+k\pi,\ x_j^+ +k\pi$）内， $g(x,$ 正切 $x)$ 的根。

求出 $k'\in\mathbb{N}$，使得 $(k'+\frac{1}{2})\pi$–是所有实根的上界
$\mathtt{LC}(g,\ y)\cdot\mathtt{discrim}(g,\ y)\cdot\prod_{j=0}^{s}g(x,\ b_j)g(x,\ a_{j+1})$。
选取一个整数 x_0，使得 $x_0>(k'+\frac{1}{2})\pi$。 –
$c_j\leftarrow 0$ 对于 $j=0,\ldots,s+1$
对于 *each real root* r *of* $g(x_0,\ y)$ **执行**
    对 j *from* 0 *to* $s+1$ **执行**
        如果 $r\in\ (a_j,\ b_j)$，**则**
            $c_j\leftarrow c_j+1$
            跳出
$ret\leftarrow\emptyset$
对 j *from* 0 *to* $s+1$ **执行**
    如果 $c_j>0$，**则**
        $x_j^-\leftarrow$ 反正切$^{a_j},\ \ x_j^+\leftarrow$ 反正切$\,b_j$
        $ret\leftarrow ret\cup\{((x_j^-,\ x_j^+)^{c_j})\}$
返回 $k',\ \ ret$

3.2. 确保“隔离”区间内周期根的精确数量

在本小节中，我们将加强定理 3.6：对于所有充分大的 k， $\theta_i(x)$ 与正切 x 在区间（反正切$\,a_{j_i}+k\pi$，反正切$\,b_{j_i}+k\pi$）内恰好相交一次。并且，可以高效地确定此类 k 的一个下界。

根据导数 $\theta_i'(x)$ $(x>M')$ 是否严格小于 1，讨论分为两种情况。对于有界情况，关键观察是：正切 x 的导数为 $1/$余弦$^2\,x$，它始终为 ≥ 1。因此，由于 $\theta_i'(x)$ 小于 1， $y=$ 正切$\,x$ 在一个周期（$k\pi-\pi/2,\ \ k\pi+\pi/2$）内不可能与 $\theta_i(x)$ 相交两次，其中 $k\in\mathbb{N}$ 且 $(k+1/2)\pi>M'$。对于无界情况，我们考虑 $1/\theta_i(x)$，而不是 $\theta_i(x)$。

我们首先证明一个引理。

引理 3.11. 设 $g\in\mathbb{Q}[x,y]$ 是无平方因子的，且 $\deg(g,y)\ \geq 1$。存在一个有限集 $S\subset\mathbb{Q}$，使得对于任意 $\delta_0\in\mathbb{Q}\setminus S,\ \ \mathtt{res}(\frac{\partial g}{\partial x}+\delta_0\frac{\partial g}{\partial y},g,y)$ 在$\mathbb{Q}[x]$中不是零多项式。

证明。 将 δ视为变量。我们声称， $\mathtt{res}(\frac{\partial g}{\partial x}+\delta\frac{\partial g}{\partial y},g,y)$ 是$\mathbb{Q}[\delta,x]$ 中的非零多项式。注意， $\frac{\partial g}{\partial y}$ 不是非零多项式，因为

$\deg(g,y) \geq 1$。于是，$q := \gcd(\frac{\partial g}{\partial x} + \delta\frac{\partial g}{\partial y}, g)$ 是 $\mathbb{Q}[x,y]$ 中的一个非零多项式。我们还注意到，$q \mid \left(\frac{\partial g}{\partial x} + \delta\frac{\partial g}{\partial y}\right)$ 意味着 $q \mid \frac{\partial g}{\partial y}$。由于 g 无平方因子，$q \mid g$ 和 $q \mid \frac{\partial g}{\partial y}$ 意味着 $\deg(q,y) = 0$。因此，$\mathrm{res}(\frac{\partial g}{\partial x} + \delta\frac{\partial g}{\partial y}, g, y)$ 不是 $\mathbb{Q}[\delta,x]$ 中的零多项式。于是，我们可以令 S 为 $\mathbb{Q}$ 中 $\mathrm{res}(\frac{\partial g}{\partial x} + \delta\frac{\partial g}{\partial y}, g, y)$ 关于 x 的系数的公共根集合。证明至此完成。

□

引理 *3.12.* 设 $g \in \mathbb{Q}[x,y]$ 是一个无平方因子多项式，$\deg(g,y) \geq 1$，δ_0 是一个有理数，使得 $\mathrm{res}(\frac{\partial g}{\partial x} + \delta_0\frac{\partial g}{\partial y}, g, y)$ 不是零多项式，$\hat{\theta}_1(x), \ldots, \theta_r(x)$ 是 g 的终端根函数，并且 $k \in \mathbb{N}$ 是一个自然数，使得 $M = (k+1/2)\pi$ 是 g 的终端根函数起始点以及 $\mathrm{res}(\frac{\partial g}{\partial x} + \delta_0\frac{\partial g}{\partial y}, g, y)$ 的所有实根的上界。那么，每个 $\theta_i(x)$ 的导数均满足以下性质之一：

(1) $\forall x > \hat{M} : \theta_i'(x) < \delta_0$;

(2) $\forall x > \hat{M} : \theta_i'(x) > \delta_0$.

我们称 k 为 g 的 δ_0-界点。特别地，如果 $\delta_0 < 1$ 且性质 (1) 对某个 $\theta_i(x)$ 成立，那么对于任意 $k > k$ $(k \in \mathbb{N})$，存在唯一的 $\xi \in (k\pi - \pi/2, k\pi + \pi/2)$，使得 $\theta_i(\xi) =$ 正切 ξ。

证明。对于每个 $\theta_i'(x)$，我们证明这两种情况之一成立。注意，M 大于 g 的终端根函数的起始点（回忆定义 3.4）。因此，当 $x \in (M, +\infty)$ 时，$\mathrm{LC}(g,y)(x) \neq 0$, $\mathrm{discrim}(g,y)(x) \neq 0$，且 $g(x, \theta_i(x)) = 0$。因此，$\frac{\partial g}{\partial y}(x, \theta_i(x)) \neq 0$，并且根据隐函数定理，$\theta_i'(x) = -\frac{\partial g/\partial x}{\partial g/\partial y}(x, \theta_i(x))$。根据 M $\mathrm{LC}\ g\ y\ x \neq$ 的定义，对 0 $\mathrm{res}\ \frac{\partial g}{\partial x} + \delta_0\frac{\partial g}{\partial y}\ g\ y\ x \neq 0$ $x \in M\ +\infty$ $g\ x\ \theta_i\ x =$ 于 (^ ,)，我们有 (,)( ) 和 (, ,)( )。然后，由于 (, ( )) 0，显然 $(\frac{\partial g}{\partial x} + \delta_0\frac{\partial g}{\partial y})(x, \theta_i(x)) \neq 0$。因此，我们有 $\theta_i'(x) \neq \delta_0$。所以，根据 $\theta_i'(x)$ 的连续性，这两种情况中恰有一种成立。

至于最后一个断言，其存在性可由中间值定理得出：对区间（$k\pi - \pi/2, k\pi + \pi/2$）上的 $\tan x - \theta_i(x)$ 应用该定理即可。其唯一性则源于 $\tan x - \theta_i(x)$ 在该区间上严格递增，因为 $1/\cos^2 x - \theta_i'(x) > 1 - \delta_0 > 0$。

□

沿用引理 3.12 中的符号说明。根据引理 3.12 的“特别地”部分，当 $\theta_i'(x) < \delta_0$ 在区间 (M, ∞) 上时，对于某个选定的 $\delta_0 < 1$，我们可以保证存在且仅存在 1 个由 θ_i 诱导的周期根，其

$g(x, \text{正切}\, x)$ 位于 $(k\pi - \pi/2, k\pi + \pi/2)$ 中，其中 k 足够大。为处理 $\theta_i'(x) > \delta_0$的情形，我们考虑 $1/\theta_i(x)$，而不是 $\theta_i(x)$。事实上，$1/\theta_i(x)$ 也是倒数多项式h 的终端根函数：$= \mathcal{R}(g)$（见定义 2.4）。如果 $y \nmid g$，根据定义，有 $\deg(h, y) = \deg(g, y) \geq 1$。此外，由于 g 无平方因子，并且显然对于任意 $c \in \mathbb{Q}$，h 也无平方因子（回顾命题 2.6）。

最后，我们得到以下定理，它加强了定理 3.6。

定理 *3.13.* 设 $g \in \mathbb{Q}[x, y]$ 无平方因子，且 $\deg(g, y) \geq 1$ 且 $y \nmid g$。令 $h = \mathcal{R}(g)$，并取 $\delta_1 \in \mathbb{Q}\ (0 < \delta_1 < 1)$、$\delta_2 \in \mathbb{Q}\ (-1 < \delta_2 < 0)$，使得 $\operatorname{res}(\frac{\partial g}{\partial x} + \delta_1 \frac{\partial g}{\partial y}, g, y)$ 和 $\operatorname{res}(\frac{\partial h}{\partial x} + \delta_2 \frac{\partial h}{\partial y}, h, y)$ 均不是零多项式。设 k' 与定理 *3.6* 中相同，k'' 是 g 的一个 δ_1-界点，k''' 是引理 *3.11* 保证存在的 h 的一个 δ_2-界点。令 $k^* := \max\{k', k'', k'''\}$，并设 $\theta_1(x), \ldots, \theta_r(x)$ 为 g 的终端根函数。那么，对于任意 $k > k^*\ (k \in \mathbb{Z})$，存在唯一的 $\xi \in (k\pi - \pi/2, k\pi + \pi/2)$，使得 $\theta_i(\xi) = \text{正切}\, \xi$。

证明。由于 $y \nmid g$、$\deg(g, y) \geq 1$，且 g 无平方因子，我们可知 $\deg(h, y) \geq 1$，且 h 无平方因子。因此，k''' 定义良好。设 $\mu := (k^* + 1/2)\pi$。根据引理 3.12，假设 $\theta_1(x), \ldots, \theta_t(x)$ 属于情形（1）（即，对于 $x \in (\mu, +\infty)$，$\theta_i'(x) < \delta_1 < 1$），而 $\theta_{t+1}(x), \ldots, \theta_r(x)$ 属于情形（2）（即，对于$x \in (\mu, +\infty)$，$\theta_i'(x) > \delta_1$），其中 $0 \leq t \leq r$。那么，根据引理 3.12 中的“特别地”部分，对于情形（1）中的每个 θ_i $(1 \leq i \leq t)$，我们已经得到了结论。下文将处理满足 $t + 1 \leq i \leq r$ 的那些 $\theta_i(x)$。

首先，我们证明对于每个 $\theta_i(x)\ (t + 1 \leq i \leq r)$，$\theta_i(x)$ 在 $(\mu, +\infty)$ 上 > 0：由于根据引理 3.12，$\theta_i'(x)$ 大于一个正数 δ_1，$\theta_i(x)$ 因而 $\to +\infty$，其中 $x \to +\infty$。根据定理 3.6 的性质 (1)，θ_i 在 $(\mu, +\infty)$ 上的像是 $(a_{s+1}, +\infty)$ 的子集，且其中有 $a_{s+1} > 0$（回顾定义 2.9）。

其次，我们证明对于每个 $\theta_i(x)\ (t + 1 \leq i \leq r)$，$\xi$ 的存在性。令 k 为任意一个大于 k^* 的整数。将介值定理应用于 $\tan x - \theta_i(x)$ 在 $(k\pi - \pi/2, k\pi + \pi/2)$ 上的情形，可知存在 $\xi \in (k\pi - \pi/2, k\pi + \pi/2)$，使得 $\tan \xi = \theta_i(\xi)$。

最后，我们证明对于每个 $\theta_i(x)$，ξ 都是唯一的（$t + 1 \leq i \leq r$）：假设对于某个 $\xi \in (k\pi - \pi/2, k\pi + \pi/2)$，$\tan \xi = \theta_i(\xi)$。那么，由于对于所有 $x \in (\mu, +\infty)$，$\theta_i(x) > 0$，可得 $\xi \in (k\pi, k\pi + \pi/2)$。注意，$1/\theta_i(x)$ 在 $(\mu, +\infty)$ 上定义良好；只要 $x \in (\mu, +\infty)$，$h(x, 1/\theta_i(x)) = 0$；并且 μ 大于 h 的终端根函数的起始点。因此，当 $x \in (\mu, +\infty)$ 时，$1/\theta_i(x)$ 是 h 的终端根函数之一。由引理 3.12，对于某个 $-1 < \delta_2 < 0$，当 $x \in (\mu, +\infty)$ 时，有 $(1/\theta_i(x))' < \delta_2$，或 $(1/\theta_i(x))' > \delta_2$。

由于 $\lim_{x\to+\infty}\theta_i(x) = +\infty$，且 $\lim_{x\to+\infty}1/\theta_i(x) = 0$，我们有 $(1/\theta_i(x))' > \delta_2$。因为 $-1/\text{正弦}^2 x - (1/\theta_i(x))' < -1-\delta_2 < 0$，函数 cot $x - 1/\theta_i(x)$ 在区间 $(k\pi, k\pi+\pi/2)$ 上严格递减。因此，ξ 是唯一的。 □

例*3.14.* 考虑例*2.10*中的多项式 $g(x,y)$。在例*3.7*中，我们取了 $k' = 3$。现在还需计算定理 *3.13* 中的 k'' 和 k'''。我们取 $\delta_1 = \frac{999}{1000}$ 和 $k'' = 3$，它是 g的 δ_1-界点。注意，$h = \mathcal{R}(g)$
$= 10xy^8 - 20y^7 - 40xy^6 + 92y^5 + 60xy^4 - 28y^3 - 40xy^2 + 20y + 10x$.我们取 $\delta_2 = -\frac{999}{1000}$ 和 $k''' = 0$，它是 h的 δ_2-界点。因此，我们有 $k^* = max\{k', k'', k'''\} = 3$。根据例*3.7*和定理 *3.13*，对于每个 $k > 3\,(k \in \mathbb{Z})$，$g(x,$ 正切 $x)$ 在区间 $(k\pi -$ 反正切 *215 128*, $k\pi -$ 反正切 *19 32*) 内恰有两个实根。

我们提出算法 2，用于计算更好的下界，以确保 $g(x, \tan x)$ 的“周期”根具有唯一性。给定一个无平方因子的多项式 $g \in \mathbb{Q}[x,y]$，使得 $\deg(g,y) \geq 1$ 且 $y \nmid g$，以及一般潜在周期区间集 I，算法 2 会计算一个自然数 k^* 和一组数对 $((x_i^-, x_i^+)\ c_i)$，使得对于任意 $k > k^*\,(k \in \mathbb{N})$，区间 $(x_i^- + k\pi, x_i^+ + k\pi)$ 中恰有 c_i 个 $g(x, \tan x)$ 的实根。算法 2 的过程如下。首先，通过算法1，我们计算$k' \in \mathbb{N}$ 和一组隔离区间 ret，使得对于任意 $k > k'$ ($k \in \mathbb{N}$) 以及 ret 中的任意数对 $((x_i^-, x_i^+)\ c_i)$，区间 $(x_i^- + k\pi, x_i^+ + k\pi)$ 中至少有 c_i 个 $g(x, \tan x)$ 的实根。然后，我们分别在第 2–4 行和第 5–8 行利用引理 3.12 计算 $k'' \in \mathbb{N}$ 和 $k''' \in \mathbb{N}$。最后，算法返回自然数 $k^* := \max\{k', k'', k'''\}$ 和集合 ret。定理 3.13 保证了算法 2 的正确性，其终止性显然。

4. 有界实根隔离

本节讨论第 2 节中的情形 (2.4a)，即给定 $k_+,\ k_- \in \mathbb{Z}$，隔离以下函数的实根:

$$g(x, \tan x) = 0, \quad x \in (k\pi - \frac{\pi}{2}, k\pi + \frac{\pi}{2}) \tag{4.1}$$

其中 $g \in \mathbb{Q}[x,y] \setminus \{0\}$、$k \in \mathbb{Z}$ 和 $k_- \leq k \leq k_+$。

我们在四个区间 $(k\pi - \frac{\pi}{2}, k\pi - \frac{\pi}{4})$，$(k\pi - \frac{\pi}{4}, k\pi)$，$(k\pi, k\pi + \frac{\pi}{4})$ 和 $(k\pi + \frac{\pi}{4}, k\pi + \frac{\pi}{2})$ 上求解 (4.1)。在每个区间内，该问题等价于隔离

算法 2：CompleteIsolatingFarZero

输入：g，$\mathbb{Q}[x,y]$ 中的一个无平方因子多项式，使得 $\deg(g,\ y) \geq 1$ 且 $y \nmid g$；
$I = [(a_0,\ b_0) \ldots,\ (a_{s+1},\ b_{s+1})]$，一个由 g构成的一般潜在周期区间集。

输出： k^*，一个自然数；
ret，一个有限对集 $((x_i^-,\ x_i^+)\ c_i)$，使得对于任意 $k > k^*$ $(k \in \mathbb{N})$，存在 $EXACTLY\ c_i$ $g(x,$ 正切 $x)$在区间 $(x_i^- + k\pi,\ x_i^+ + k\pi)$ 内的根。

$k',\ ret \leftarrow$ **IsolatingFarZero**(g, I)
$d_1 \leftarrow \mathtt{res}\ (\frac{\partial g}{\partial x} + \delta \frac{\partial g}{\partial y},\ g,\ y)$ *#In this step, δ is a variable and $d_1 \in \mathbb{Q}[\delta, x]$.*
选择 $\delta_1 \in (0,\ 1) \cap \mathbb{Q}$，使得 $d_1(\delta_1,\ x)$ 不是 $\mathbb{Q}[x]$中的零多项式。
求 $k'' \in \mathbb{N}$，使得 $(k'' + \frac{1}{2})\ \pi$ 是该多项式所有实根的上界
$$d_1\ (\delta_1,\ x)\ \cdot \mathtt{LC}\ (g,\ y)\ \cdot \mathtt{discrim}\ (g,\ y)\ 。$$
$h \leftarrow g(x,\ \frac{1}{y}) \cdot y^{\deg(g,y)}$
$d_2 \leftarrow \mathtt{res}(\frac{\partial h}{\partial x} + \delta \frac{\partial h}{\partial y}, h, y)$
选择 $\delta_2 \in (-1,\ 0) \cap \mathbb{Q}$，使得 $d_2(\delta_2,\ x)$不是 $\mathbb{Q}[x]$中的零多项式。
求 $k''' \in \mathbb{N}$，使得$(k''' + \frac{1}{2})\pi$ 是该多项式所有实根的上界
$$d_2(\delta_2,\ x)\ \cdot \mathtt{LC}(h,\ y)\ \cdot \mathtt{discrim}(h,\ y)。$$
$k^* \leftarrow$ 最大值$\{k', k'', k'''\}$
返回 k^* ，ret

区间 $(0,\ 1)$ 上某个 $G(r\pi +$ 反正切 $y,\ y)$ 的根，其中 $G \in \mathbb{Q}[x,y]$ 和 $r \in \mathbb{Q}$，因此我们可以采用 $[4,$ 算法 $47]$ 的框架来处理。在第 4.1 节中，我们使用算法 3 计算 $G(r\pi +$ 反正切 $y,\ y)$ 的弱傅里叶序列。事实上，算法 3 是针对我们特定需求对 [4,算法 40] 的简化版本。在第 4.2 节中，我们提出算法 4 来隔离方程 (4.1) 的实根。

4.1. 弱傅里叶序列

我们先回顾弱傅里叶序列的定义，以及一个关于它的重要结果。

定义 4.1. *[4,* 定义 *10]* 设 $I \subseteq \mathbb{R}$ 为一个开区间，且 $G_1, \ldots, G_k : I \to \mathbb{R}$ 是定义在 I 上的一列可微函数。若对所有 $y \in I$，序列 $G_1, \ldots, G_k$ 是 I 上的弱傅里叶序列，

1. *for all* $i\ (1 \leq i \leq k-1)$, $\mathtt{sign}(G_{i+1}(y)) = \mathtt{sign}(G_i'(y))$, *and*
2. $G_k(y) \neq 0$.

若 $G_1, \ldots, G_k$ 是 G 在 I上的弱傅里叶序列，则称其为.....，当且仅当 $G_1, \ldots, G_k$ 是 I 上的弱傅里叶序列，且对所有 $y \in I$，$G_1(y) = G(y)$ 。

符号说明 4.2. 设 $a, b \in \mathbb{R}$ 和 $G_1, \ldots, G_k$ 是开区间 (a, b) 上的弱傅里叶序列。那么，对于任意 $i, j\ (1 \leq i \leq j \leq k)$,

1. $\mathtt{sign}(G_i(x^+)) := \mathtt{sign}(G_i(c))$, 其中 $x \in [a,b)$, $c < b$ 和 $\forall y \in (x, c]\ G_i(y) \neq 0$, 2. $\mathtt{sign}(G_i(x^-)) := \mathtt{sign}(G_i(d))$, 其中 $x \in (a,b]$, $a < d$ 和 $\forall y \in [d,x)\ G_i(y) \neq 0$, 3. $\mathtt{sgc}_{i,j}(x)$, 对于 $x \in (a,b)$, 是序列 $G_i(x) \ldots, G_j(x)$（去除等于零的项）中的符号变化次数, 4. $\mathtt{sgc}_{i,j}(x^+)$, 对于 $x \in [a,b)$, 是序列$G_i(x^+) \ldots, G_j(x^+)$中的符号变化次数, 5. $\mathtt{sgc}_{i,j}(x^-)$, 对于 $x \in (a,b]$, 是序列 $G_i(x^-) \ldots, G_j(x^-)$ 中的符号变化次数。

定理 4.3. [4, 定理 13] 设 $I \subseteq \mathbb{R}$ 为一个开区间, 且 $G_1, \ldots, G_k : I \to \mathbb{R}$ 为一个弱傅里叶序列。那么对于任意 $x \in I$, $\mathtt{sgc}_{1,k}(x^+) = \mathtt{sgc}_{1,k}(x)$ 和 $\mathtt{sgc}_{1,k}(x^-) = \mathtt{sgc}_{1,k}(x) + r + 2s$, 其中 $r, s \in \mathbb{N}$ 且 $G_1(x) = \cdots = G_r(x) = 0, G_{r+1}(x) \neq 0$。此外, 除非存在一个 $t > r+1$ 使得 $G_t(x) = 0$, 否则 $s = 0$。 □

设 a, b 为两个实数, 使得 $(a,b) \subseteq I$。注意, 根据 [4, 命题 15], 的证明, 若 $\mathtt{sgc}_{1,k}(a^+) = \mathtt{sgc}_{1,k}(b^-)$, 则 G_1 在 (a,b) 中没有实根; 若 $\mathtt{sgc}_{1,k}(a^+) = \mathtt{sgc}_{1,k}(b^-) + 1$, 则 G_1 在 (a,b) 中恰有一个单根。

为说明算法 3, 我们引入反正切导数的定义。

定义 4.4。对于 $G \in \mathbb{Q}(y)[x]$, 令

$$\tilde{G} := \frac{\partial G}{\partial x} \cdot \frac{1}{1+y^2} + \frac{\partial G}{\partial y} \in \mathbb{Q}(y)[x].$$

$\tilde{G}$ 的反正切导数定义为 $G^\star := G \cdot h(y)$, 其中 $h(y) \in \mathbb{Q}[y]$ 是 $\mathtt{LC}(G,x) \in \mathbb{Q}(y)$的分母。特别地, 如果 $\mathtt{LC}(G,x)$ 是一个多项式, 则 $h(y) = 1$。我们称 $h(y)$ 为 $G^\star$的公因子。

注 4.5. 对于任意 $G \in \mathbb{Q}(y)[x]$, $G^\star \in \mathbb{Q}(y)[x]$ 和 $\mathtt{LC}(G^\star, x)$, 它都是关于 $\mathbb{Q}[y]$ 的多项式。

给定 $G(x,y) \in \mathbb{Q}[x,y] \setminus \{0\}$ 和 $r \in \mathbb{Q}$，算法 3 的过程如下：令 G_1 为 $G(x,y)$。若 G_1 不是非零有理数，则计算 G_1 的反正切导数 $G_1^\star$，并令 $G_2 = G_1^\star$。若 G_2 不是非零有理数，则令 $G_3 = G_2^\star$。重复此过程，直到某个 $G_k (k \geq 1)$ 是非零有理数。于是 $G_1(r\pi + \text{反正切}\, y, y), \ldots, G_k(r\pi + \text{反正切}\, y, y)$ 是 $G(r\pi + \text{反正切}\, y, y)$ 在 $\mathbb{R}$ 上的弱傅里叶序列。

算法 3：弱傅里叶序列

输入：$G(x, y)$，关于 $\mathbb{Q}[x,y] \setminus \{0\}$ 的多项式；r，一个有理数
输出：$G_1(r\pi + \text{反正切}\, y, y)$。。。，$G_k(r\pi + \text{反正切}\, y, y)$，一个弱傅里叶序列，适用于 $G(r\pi + \text{反正切}\, y, y)$ 在 $\mathbb{R}$ 上，其中 $G_i \in \mathbb{Q}(y)[x]$

```
G_1 ← G(x, y)
i ← 1
当 G_i(x, y) is not a nonzero rational number 时
    G_{i+1}(x, y) ← G_i^⋆(x, y)
    i ← i + 1
返回 G_1(rπ + 反正切 y, y)..., G_k(rπ + 反正切 y, y)
```

注意，我们可以根据 [4, 命题 15]，轻松地从半傅里叶序列（定义见 [4, 定义 14]）得到一个弱傅里叶序列。因此，在为 $G(r\pi + \text{反正切}\, y, y)$ 计算弱傅里叶序列时，算法 3 和 [4, 算法 40] 本质上相同，因此算法 3 的终止性和正确性显而易见。然而，为了解释后续将用到的一些性质，我们准备了若干引理，并重新证明算法 3 的终止性和正确性。

引理 4.6。若 $G \in \mathbb{Q}(y)[x] \setminus \mathbb{Q}$ 使得 $G(x,y)$ 在整个平面 $\mathbb{R}^2$上有定义，则 $G^\star \in \mathbb{Q}(y)[x] \setminus \{0\}$。

证明。假设 $G^\star$ 是 $\mathbb{Q}(y)[x]$中的零多项式。根据定义 4.4，G 也是零多项式。于是，$G(\text{反正切}\, y, y) \equiv 0$。注意，$\tilde{G}(\text{反正切}\, y, y) = \mathrm{d}G(\text{反正切}\, y, y)\,\mathrm{d}y$。因此，对于任意 $y \in \mathbb{R}$，$G(\text{反正切}\, y, y) \in \mathbb{Q}$。由于反正切函数不是代数的，$G \in \mathbb{Q}$，矛盾。 □

引理 4.7。如果 $G \in \mathbb{Q}(y)[x] \setminus \mathbb{Q}$ 且 $\mathtt{LC}(G, x)$ 是一个多项式，那么

$$(\deg(G,x), \deg(\mathtt{LC}(G,x))) \succ (\deg(G^\star,x), \deg(\mathtt{LC}(G^\star,x))), \tag{4.2}$$

其中 $\succ$表示字典序。

证明。 回顾注 4.5。次数 $\deg(G^\star, x)$ 和 $\deg(\mathrm{LC}(G^\star, x))$ 是定义良好的。

若 $\deg(G, x) = 0$，则有 $G \in \mathbb{Q}(y) \setminus \mathbb{Q}$，因此 $\mathrm{LC}(G, x) = G$。所以， G是关于 $\mathbb{Q}[y] \setminus \mathbb{Q}$ 的多项式。容易验证 $G^\star = \frac{\partial G}{\partial y} \in \mathbb{Q}[y]$。因此，$\deg(\mathrm{LC}(G, x)) = \deg(G)$, $\deg(G^\star, x) = 0$，且 $\deg(\mathrm{LC}(G^\star, x)) = \deg(G) - 1$。于是，不等式 (4.2) 成立。

若 $\deg(G, x) \geq 1$，设

$$G = a_n(y)x^n + \sum_{j=0}^{n-1} a_j(y)x^j \tag{4.3}$$

其中 $n = \deg(G, x) \geq 1$, $a_n(y) \in \mathbb{Q}[y]$ 且 $a_j(y) \in \mathbb{Q}(y)$ 对于 $j = 0, \ldots, n-1$ 成立。于是,

$$\tilde{G} = a_n'(y)x^n + \sum_{j=0}^{n-1} \big(a_j'(y) + \frac{(j+1)a_{j+1}(y)}{1+y^2}\big)x^j. \tag{4.4}$$

若 $a_n'(y)$ 是零多项式，则有 $\deg(G, x) < \deg(G, x)$。根据定义4.4，$\deg(G^\star, x) = \deg(G, x)$。因此，$\deg(G^\star, x) < \deg(G, x)$，不等式 (4.2) 成立。若 $a_n'(y)$ 不是零多项式，则 $\deg(G, x) = \deg(G, x)$。由于 $\mathrm{LC}(G, x) = a_n'(y) \in \mathbb{Q}[y]$, $G^\star = G$。因此，$\deg(G^\star, x) = \deg(G, x)$。我们还注意到，$\deg(\mathrm{LC}(G, x)) = \deg(a_n(y))$，且 $\deg(\mathrm{LC}(G^\star, x)) = \deg(a_n(y)) - 1$。证明完毕。 □

引理 4.8。 如果 $G \in \mathbb{Q}(y)[x]$ 具有以下形式

$$G = a_n(y)x^n + \sum_{j=0}^{n-1} \frac{a_j(y)}{(1+y^2)^{m_j}}x^j, \tag{4.5}$$

其中 $n = \deg(G, x)$、 $a_i(y) \in \mathbb{Q}[y]\ (0 \leq i \leq n)$ 和 $m_j \in \mathbb{N}\ (0 \leq j \leq n-1)$，则 $G^\star$ 的公因子为 $(1+y^2)^m$，其中 $m \in \mathbb{N}$。此外， $G^\star$也具有方程 (4.5)中的形式。

证明。 若 $\deg(G, x) = 0$，则 $G = a_n(y) \in \mathbb{Q}[y]$，因此 $G^\star = G'(y)$， $G^\star$ 的公因子为 1。若 $\deg(G, x) \geq 1$，根据引理 4.7 的证明中的式（4.3）和式（4.4），存在 $b_i(y) \in \mathbb{Q}[y]\ (0 \leq i \leq n)$ 和 $d_j \in \mathbb{N}\ (0 \leq j \leq n-1)$，使得

$$\tilde{G} = b_n(y)x^n + \sum_{j=0}^{n-1} \frac{b_j(y)}{(1+y^2)^{d_j}}x^j.$$

若 $b_n(y)$ 不是零多项式，则 $G^\star$ 的公因子为 1。否则，由于 $\deg(G,x) \geq 1$，且 G 在整个 $\mathbb{R}^2$上有定义，根据引理 4.6 的证明， G 不是零多项式。因此，存在一个整数 $0 \leq k \leq n-1$，使得 $b_k(y)$ 不是零多项式。令 k_0 为满足此条件的最大 k。那么， $G^\star$ 的公因子为 $(1+y^2)^{d_{k_0}}$。因此， $G^\star$与式（4.5）具有相同的形式。 □

定理 4.9. 算法 *3* 正确终止。

证明. (终止性) 对任意 $G(x,y) \in \mathbb{Q}[x,y]\backslash\{0\}$，设 $G_1 := G(x,y)$ $G_2 := G_1^\star(x,y)$ $G_3 := G_2^\star(x,y)\ldots$我们只需证明存在 $k(k \geq 1)$，使得 $G_k(x,y)$ 是一个非零有理数。若 $G_1 \in \mathbb{Q} \setminus \{0\}$，则取 $k = 1$。若 $G_1 \in \mathbb{Q}[x,y] \setminus \mathbb{Q}$，则由引理 4.7 可得

$$(\deg(G_1,x), \deg(\mathtt{LC}(G_1,x))) \succ (\deg(G_2,x), \deg(\mathtt{LC}(G_2,x))).$$

回顾注 4.5。对任意 i $(i \geq 2)$， $G_i \in \mathbb{Q}(y)[x]$ 和 $\mathtt{LC}(G_i,x)$ 是一个多项式。因此，假设 $G_2 \notin \mathbb{Q}, G_3 \notin \mathbb{Q}$,...，由引理 4.7 可得

$$\begin{aligned}(\deg(G_1,x), \deg(\mathtt{LC}(G_1,x))) &\succ (\deg(G_2,x), \deg(\mathtt{LC}(G_2,x))) \\ &\succ (\deg(G_3,x), \deg(\mathtt{LC}(G_3,x))) \succ \cdots .\end{aligned}$$

因此，存在 m $(m \geq 2)$ 使得 $G_m \in \mathbb{Q}$。令 k $(k \geq 2)$ 为满足 $G_k \in \mathbb{Q}$ 的最小整数。于是， $G_{k-1} \notin \mathbb{Q}$。注意，根据引理 4.8， G_{k-1} 在整个 $\mathbb{R}^2$ 上都有定义。因此，根据引理 4.6， G_k 是一个非零有理数。

（正确性）对于 $i = 1,\ldots,k-1$，令 $h_i(y)$ 为 $G_i^\star(x,y)$ 的公因子。回顾定义 4.4。我们有 $G_{i+1}(x,y) = h_i(y)G_i(x,y)$。注意，

$$\tilde{G}_i(r\pi + \arctan y, y) = G_i(r\pi + \arctan y, y)'$$

且 $G_{i+1}(x,y) \in \mathbb{Q}(y)[x]$。因此，我们只需证明 $\mathtt{sign}(h_i(y)) > 0$在整个 $\mathbb{R}$上成立，并且 $G_{i+1}(r\pi + 反正切\, y, y)$ 在整个 $\mathbb{R}$上有定义。若 $\deg(G,x) = 0$，则 $G \in \mathbb{Q}[y]$。所以， $h_i(y) = 1$，且 $G_{i+1}(r\pi + 反正切\, y, y)$ 在整个 $\mathbb{R}$上有定义。若 $\deg(G,x) \geq 1$，根据终止性证明，存在 $m(1 \leq m \leq k-1)$，使得 $\deg(G_1,x) \geq 1$，....., $\deg(G_m,x) \geq 1$，但 $\deg(G_{m+1},x) = 0$。将引理 4.8 应用于 $G_i(i = 1,\ldots,m)$，可得$h_i(y) = (1+y^2)^{n_i}(n_i \in \mathbb{N})$，且 $G_{i+1}(r\pi + 反正切\ y, y)$ 在整个$\mathbb{R}$上有定义。由于 $\deg(G_{m+1},x) = 0$， $G_{m+1} \in \mathbb{Q}[y]$。因此，对于任意 $m+1 \leq i \leq k-1$，$h_i(x) = 1$ 且 $G_{i+1} \in \mathbb{Q}[y]$，它在整个 $\mathbb{R}$上也有定义。 □

备注 4.10。令 $G(x,y) \in \mathbb{Q}[x,y]\backslash\{0\}$, $r \in \mathbb{Q}$ 和 $G_1(r\pi+$反正切 $y, y) \ldots, G_k$ $(r\pi+$反正切 $y, y)$ 为由算法 *3* 计算得到的 $G(r\pi+$反正切 $y, y)$ 在 $\mathbb{R}$上的弱傅里叶序列。根据定理 *4.9* 的正确性证明，$G_i(x,y) \in \mathbb{Q}[y]$ 或者

$$G_i(x,y) = a_{in_i}(y)x^{n_i} + \sum_{j=0}^{n_i-1} \frac{a_{ij}(y)}{(1+y^2)^{m_{ij}}}x^j,$$

其中 $n_i = \deg(G_i, x) \geq 1$, $a_{in_i}(y) \in \mathbb{Q}[y]$ $(1 \leq i \leq k)$ 以及 $m_{ij} \in \mathbb{N}$ $(0 \leq j \leq n_i - 1)$。对于 $i = 1, \ldots, k$，令 $m_i := \max(m_{ij}, 0 \leq j \leq n_i - 1)$ 并

$$F_i := \begin{cases} G_i, & G_i \in \mathbb{Q}[y], \\ (1+y^2)^{m_i}G_i, & \text{otherwise.} \end{cases}$$

我们有 $F_i \in \mathbb{Q}[x,y]$，且 $F_i(r\pi +$反正切 $y, y)$ 在 $\mathbb{R}$ 上具有与 $G_i(r\pi +$反正切 $y, y)$ 相同的符号和实根。因此，在下一个小节中，我们考虑 F_i，而不是 G_i。

4.2. 根隔离

回顾一下，第4节的目标是隔离下列方程的根：

$$g(x, \tan x),\ x \in (k\pi - \frac{\pi}{2}, k\pi + \frac{\pi}{2}),\ \text{where } k \in \mathbb{Z}. \tag{4.6}$$

4.2.1. 算法

给定 $g(x,y) \in \mathbb{Q}[x,y]\backslash\{0\}$ 和 $k \in \mathbb{Z}$，我们提出算法 4，以隔离方程 (4.6) 的实根。

算法 4 的流程如下。首先根据引理 2.2 判断 $k\pi$、$-\frac{\pi}{4}+k\pi$ 或 $\frac{\pi}{4}+k\pi$ 是否为 $g(x,$ 正切 $x)$ 的根。然后，计算 $g(x,y)$ 在 $\mathbb{Q}[x,y]$ 中的所有不可约因子 g_1、$\ldots$、g_m。对于每个 $g_i(x,$ 正切 $x)$，我们只需在四个开区间内隔离其所有实根，即 $(k\pi-\frac{\pi}{2}, k\pi-\frac{\pi}{4})$、$(k\pi-\frac{\pi}{4}, k\pi)$、$(k\pi, k\pi+\frac{\pi}{4})$ 和 $(k\pi+\frac{\pi}{4}, k\pi+\frac{\pi}{2})$。如果 g_i 是 $\mathbb{Q}[x]$ 中的多项式，处理起来很容易（一元多项式的实根有许多隔离方法）。否则，我们有 $\deg(g_i, y) \geq 1$。令 $g_i^{(1)}(x,y) := g_i(-x,-y)$。容易验证，在第一个区间（分别在第二个区间）内隔离 $g_i(x,$ 正切 $x)$ 的实根，等价于在 $(-k\pi+\frac{\pi}{4}, -k\pi+\frac{\pi}{2})$（分别在 $(-k\pi, -k\pi+\frac{\pi}{4})$）内隔离 $g_i^{(1)}(x,$ 正切 $x)$ 的实根。注意，这四种情况都可以归结为隔离实根的问题

在 $(r\pi, r\pi + \frac{\pi}{4})$ 上隔离 $G(x,$ 正切 $x)$ 的实根，其中 $G \in \{g_i, g_i^{(1)}\}$ 是一个有理二元不可约多项式，满足 $\deg(G, y) \geq 1$，且 $r \in \{k_0, k_0 + \frac{\pi}{4}\}$（$k_0 \in \mathbb{Z}$）。算法 5 可以解决这一问题，我们将在下一小节中详细说明。最后，在算法 2 中，对每个不可约因子 g_i 得到的隔离区间进行细化，使其彼此不相交。这是利用备注 4.14 以及区间内的根均为互异单根这一事实实现的（命题5.1–5.2）。

算法4：IsolatingBoundedRoots

输入： $g(x, y)$，一个关于 $\mathbb{Q}[x, y] \setminus \{0\}$ 的多项式
k，一个整数
输出： ans，$g(x,$ 正切 $x)$ 的实根的一组两两不相交的隔离区间，位于 $(k\pi - \frac{\pi}{2}, k\pi + \frac{\pi}{2})$

$ans \leftarrow \emptyset$
如果 $g(k\pi, 0) = 0$，那么 $ans \leftarrow ans \cup \{[k\pi, k\pi]\}$
如果 $g(-\frac{\pi}{4} + k\pi, -1) = 0$，那么 $ans \leftarrow ans \cup \{[-\frac{\pi}{4} + k\pi, -\frac{\pi}{4} + k\pi]\}$
如果 $g(\frac{\pi}{4} + k\pi, 1) = 0$，那么 $ans \leftarrow ans \cup \{[\frac{\pi}{4} + k\pi, \frac{\pi}{4} + k\pi]\}$
$g_1, \ldots, g_m \leftarrow g(x, y)$ 在 $\mathbb{Q}[x, y]$ 中的所有不可约因子
对于 i *from* 1 *to* m，**执行**
  如果 g_i *is a polynomial in* $\mathbb{Q}[x]$，**那么**
    $S \leftarrow g_i(x)$ 在以下区间内实根的隔离区间集合 $(k\pi - \frac{\pi}{2}, k\pi + \frac{\pi}{2}) \setminus \{k\pi, -\frac{\pi}{4} + k\pi, \frac{\pi}{4} + k\pi\}$
    $ans \leftarrow ans \cup S$
  else
    令 $g_i^{(1)} \in \mathbb{Q}[x, y]$ 满足 $g_i^{(1)}(x, y) = g_i(-x, -y)$
    $S_1 \leftarrow$ **SubIBR** $(g_i^{(1)}, -k + \frac{1}{4})$
    $S_2 \leftarrow$ **SubIBR**$(g_i^{(1)}, -k)$
    $S_3 \leftarrow$ **SubIBR**(g_i, k)
    $S_4 \leftarrow$ **SubIBR**$(g_i, k + \frac{1}{4})$
    $ans \leftarrow ans \cup \{(-b, -a) \mid (a, b) \in S_1 \cup S_2\} \cup S_3 \cup S_4$
细化上述循环中得到的区间，使其互不相交。**返回** ans

4.2.2. 子算法

设 $g(x, y)$ 是 $\mathbb{Q}[x, y]$ 中的一个不可约多项式，$\deg(g, y) \geq 1$ 且 $r \in \{k, k + \frac{1}{4}\}$，其中 $k \in \mathbb{Z}$。根据 $[4,$ 算法 $47],$ 的框架，提出算法 5，以隔离 $g(x,$ 正切 $x)$ 在 $(r\pi, r\pi + \frac{\pi}{4})$ 中的实根。值得注意的是，$[4,$ 算法 $47]$ 与算法 5 并不相同。这里增加了一些步骤，也省略了一些步骤，或以其他方式实现了一些步骤。

算法 $[4,$ 算法 $47]$ 可用于在开区间（$a,$ {v8）中隔离指数-对数-反正切函数 $w(y)$ 的实根，其中 a, b 为有理数

，且（a, b）包含在 $w(y)$ 的定义域内。下面回顾算法的框架。首先，我们需要计算 $w(y)$ 在（a, b）上的弱傅里叶序列w_1，.....，w_k。设 ans 是已计算出的 $w(y)$ 的隔离区间集合（初始化为 $\emptyset$），设 $stack$ 是由三元组（c, d, m）组成的栈，其中 $c < d$ 为有理数，且 w_m 在（c, d）上具有恒定的非零符号（初始化为（a, b, k））。在每次循环中，我们从 $stack$ 中选取一个三元组（c, d, m），并通过 [4,算法 44], 计算差值 $\mathtt{sgc}_{1,m}(c^+) - \mathtt{sgc}_{1,m}(d^-)$，直到 $stack$ 为空。

1. 若该值等于 0，根据定理 4.3，$w(y)$ 在 (c, d) 上没有根。因此，我们继续下一轮循环。2. 若该值等于 1，根据定理 4.3，$w(y)$ 在 (c, d) 上恰有一个单根。我们将 (c, d) 加入 ans。3. 否则，令 ℓ 为 $\{i \mid 2 \le i \le m$ 和 $\mathtt{sgc}_i$ 中较小的元素，且 $_m(c^+) - \mathtt{sgc}_i$，$_m(d^-) = 1\}$。根据定理 4.3，$w_\ell(y)$ 在 (c, d) 上恰有一个单根sr。(a) 若 sr 是 $w(y)$ 的根，求两个有理数 u、v，使得$c < u < sr < v < d$，且 $w(y)$ 在 $[u, v]$ 中只有一个根 sr（回想一下，根据引理 2.1，我们可以找到这样的 u、v）。将 (u, v) 加入 ans，并将 (c, u, ℓ)、(v, d, ℓ) 加入 $stack$。(b) 若 sr 不是 $w(y)$ 的根，求两个有理数 u、v，使得 $c < u < sr < v < d$，且 $w(y)$ 在 $[u, v]$ 中没有根。将 (c, u, ℓ) 和 (v, d, ℓ) 加入 $stack$。

在说明算法 5 之前，我们先定义一个映射 φ，并介绍引理 4.11 和引理 4.12。映射 φ： $\mathbb{Q}[x, y] \to \mathbb{Q}[x, y_1]$ 定义如下：

$$\varphi(g(x, y)) = g(x, \frac{1 + y_1}{1 - y_1}) \cdot (1 - y_1)^{\deg(g,y)}. \tag{4.7}$$

引理 4.11。若 g是 $\mathbb{Q}[x, y] \setminus \{c(y+1) \mid c \neq 0\}$中的不可约多项式，则 $\varphi(g)$ 是 $\mathbb{Q}[x, y_1]$中的不可约多项式。

证明。由于 $\deg(g(x, 2y-1)\ y) = \deg(g(x, y)\ y)$，$\varphi(g) = \mathcal{R}(g(x, 2y-1))|_{y=1-y_1}$，其中 $\mathcal{R}$ 满足定义 2.4。现在，对于任意非零的 $c \in \mathbb{Q}$，$g(x, y) \neq c(y + 1)$；对于任意非零的 $c \in \mathbb{Q}$，$g(x, 2y - 1) \neq cy$。此外，由于 $g(x, y)$ 不可约，$g(x, 2y - 1)$ 也不可约。根据命题 2.5，$\mathcal{R}(g(x, 2y - 1))$ 不可约。因此，$\varphi(g) = \mathcal{R}(g(x, 2y - 1))|_{y=1-y_1}$ 也不可约。□

下面的引理与 [22, 引理 3.2] 以及 [7, 定理 7] 类似。

引理 4.12. 设 $F_1, F_2 \in \mathbb{Q}[x, y], k \in \mathbb{Z}$和 $r \in \{k, k+\frac{1}{4}\}$。若 $\gcd(F_1, F_2) = 1$，则 $F_1(r\pi + \text{反正切}\, y, y)$ 和 $F_2(r\pi + \text{反正切}\, y, y)$ 在$(0,1)$内没有公共实根。

证明。在 [7, 定理 7] 的证明中，将其中的“p, q”替换为本引理中的 F_1 和 F_2，将其中的“$f^*(x)$ $g^*(x)$”替换为 $F_1(r\pi + \text{反正切}\ y, y)$ 和 $F_2(r\pi + \text{反正切}\ y, y)$，并将其中的函数“trans”替换为函数 $r\pi + \text{反正切}$，可知：若 $\beta \in (0,1)$ 是 $F_1(r\pi + \text{反正切}\ y, y)$ 和$F_2(r\pi + \text{反正切}\, y, y)$ 的公共根，则 β和 $\alpha = r\pi + \text{反正切}\, \beta$均为代数的，这与下文的断言 4.13 矛盾。 □

断言 4.13。对于任意 $y \in (0, 1)$， $r\pi + \text{反正切}\ y$和 y中至少有一个是超越的。

证明。注意到

$$\tan(r\pi + \arctan y) = \begin{cases} y, & \text{if} \quad r = k \in \mathbb{Z}, \\ \frac{1+y}{1-y}, & \text{if} \quad r = k + \frac{1}{4} \in \mathbb{Z} + \frac{1}{4}. \end{cases}$$

假设 y是代数的，那么在两种情况下，$\tan(r\pi + \text{反正切}\ y)$ 也都是代数的。由于 y 和 $(1+y)/(1-y)$ 均不为零，因此 $r\pi + \text{反正切}\, y \neq 0$。此外，根据定理 2.3，它是超越的。 □

备注 *4.14*。假设 $h(x_1, x_2) = \sum_{i=0}^{\deg(h, x_1)} c_i(x_2) x_1^i \in \mathbb{Q}[x_1, x_2]$。对于任意 $y \in (0, 1) \cap \mathbb{Q}$， $r\pi + \text{反正切}\ y$是超越的。因此， $h(r\pi + \text{反正切}\, y, y) = 0$当且仅当 $c_i(y) = 0$ 对所有 i均成立。由此，我们可以判断给定 $y \in (0, 1) \cap \mathbb{Q}$时，$h(r\pi + \text{反正切}\, y, y)$ 是否为零。如果它不为零，我们还可以通过提高近似精度来计算其符号。更一般地，若 ϕ 是一个实函数（例如 $\phi = \text{正切}$），且对其定义域内任意非零有理数 y， $\phi(y)$ 都是超越的，那么我们就能判断 $h(\phi(y)\ y)$ 是否为零，并在其不为零时计算其符号。

回顾一下，我们的目标是在 $(r\pi, r\pi + \frac{\pi}{4})$ 内隔离 $g(x, \text{正切}\ x)$ 的实根，其中 g 是 $\mathbb{Q}[x, y]$ 中的不可约多项式，且 $\deg(g, y) \geq 1$，并且 $r \in \{k, k + \frac{1}{4}\}$ $(k \in \mathbb{Z})$。算法 5 的过程如下。若 $g = y + 1$，则正切 $x + 1$ 在 $(r\pi, r\pi + \frac{\pi}{4})$ 上没有根，因此返回 $\emptyset$。否则，当 $r = k$时，令 $F_1 := g$；当 $r = k + \frac{1}{4}$ 时，令 $F_1 := \varphi(g)$。不难验证，我们的目标等价于求解 $(0, \frac{\pi}{4})$ 上的 $F_1(x + r\pi, \text{正切}\ x)$。

令 $y :=$ 正切 x。我们只需在 (0，1) 上求解 $F_1(r\pi +$ 反正切 $y,\ y)$。回顾备注 4.10。我们可以计算一个弱傅里叶序列

$$w_1 = F_1(r\pi + \arctan y, y), \ldots, w_k = F_k(r\pi + \arctan y, y) \text{ on } (0,1), \quad (4.8)$$

其中 $F_i \in \mathbb{Q}[x,y]$。因此，可以使用 $[4,$ 算法 $47]$。不过，$[4,$ 算法 $47]$与算法 5 之间有两个主要区别。

首先，当 w_ℓ 在 (c,d) 上恰有一个单根 sr 时（见第 17 行），我们提出一种新方法来判断 sr 是否为 w_1 的根。如果 $F_\ell = QF_1$ ($Q \in \mathbb{Q}[y]$) 成立，且 Q 在 (c, d) 上没有根，那么根据 (4.8)，sr 是 w_1 在 (c,d) 上的唯一根。如果 $F_\ell = QF_1$ 成立，且 Q 在 (c,d) 上有根，那么显然 w_1 在 (c,d) 上没有根。如果 $F_\ell = QF_1$ 不成立，那么 sr 不是 w_1 的根。该方法是正确的：根据引理 4.11，F_1 是一个不可约多项式。如果 $F_1 \mid F_\ell$，则有 $\deg(F_1, x) \le \deg(F_\ell, x)$。于是，根据引理 4.7，$\deg(F_1, x) = \deg(F_\ell, x)$。因此，$Q$ 是 $\mathbb{Q}[y]$ 中的一个多项式。否则，$\gcd(F_1, F_\ell) = 1$。于是，根据引理 4.12，sr 不是 w_1 的根。

其次，如果 sr 不是 w_1的根，我们通过算法 6（见算法 5 第 25 行）找到两个有理数 u、v，使得 $c \le u < sr < v \le d$，且 w_1 在 $[u,v]$ 中没有根，而不是通过 $[4,$ 算法 $45]$。算法 6 的过程如下：令 $n := 1$。当 $n \ge 1$时，我们通过两个步骤计算 $W_{\max}$的上界多项式 $_n \in \mathbb{Q}[y]$ 和 $W_{\min}$的下界多项式 $_n \in \mathbb{Q}[y]$，用于 $w_1(y)$。由于 $w_1 = F_1(r\pi +$ 反正切 $y,\ y)$，我们首先通过泰勒级数估计反正切 y，并计算 $T_{\max}(n,\ w_1)$ $T_{\min}(n,\ w_1) \in \mathbb{Q}[\pi][y]$，计算依据为 $[15,$ 定义 3.1$]$。然后，我们使用一个具有有理端点的有界区间，估计 π 在 $T_{\max}(n,\ w_1)$ 和 $T_{\min}(n,\ w_1)$中的取值，并根据 $[25,$ 定义 3.1$]$计算多项式$W_{\max}$、$_n$、$W_{\min}$、$_n \in \mathbb{Q}[y]$。令 $q := \frac{c+d}{2}$。如果 $W_{\max}$，则 $_n$在 $[c,d]$ 上没有根，且 $W_{\max}$、$_n(q) < 0$；或者如果 $W_{\min}$，则 $_n$ 在 $[c,d]$上没有根，且 $W_{\min}$、$_n(q) > 0$；那么 w_1 在 $[c,d]$上没有根。否则，将 n 加 1。由于 w_ℓ 在 $(c,\ d)$上恰有一个单根 sr，我们使用基于备注 4.14 的二分法，找到包含 sr 的半开区间。将 $(c,$ $d)$更新为该半区间，然后进入下一轮。请注意，如果区间 $(c,\ d)$ 足够小，且上界（下界）多项式足够接近，那么算法 6 将根据引理 2.1 终止。

算法 5：SubIBR

输入 ： $g(x,y)$，一个在 $\mathbb{Q}[x,y]$ 中的不可约多项式，且 $\deg(g,y)\geq 1$
$r, r\in\{k,k+\frac{1}{4}\}$ 其中 $k\in\mathbb{Z}$
输出： ans，$g(x,$正切$x)$ 在 $(r\pi, r\pi+\frac{\pi}{4})$ 中实根的隔离区间集合

若 $g=y+1$, **则返回** $\emptyset$
若 $r=k$ **则**
  $F_1\leftarrow g$
否则
  $F_1\leftarrow g(x,\frac{1+y}{1-y})\cdot(1-y)^{\deg(g,y)}$
令 $w_1=F_1(r\pi+$反正切 $y,\ y)\ldots,\ w_k=F_k(r\pi+$反正切 $y,\ y)$，其中 $F_i\in\mathbb{Q}[x,y]$，为备注 4.10 中的序列
$ans\leftarrow\emptyset$
$stack\leftarrow\{(0,1,k)\}$
当 $stack\neq\emptyset$ **时**
  $(c,\ d,\ m)\leftarrow First(stack)$ $stack\leftarrow Rest(stack)$
  $sgc\leftarrow \mathtt{sgc}_{1,m}(c^+)-\mathtt{sgc}_{1,m}(d^-)$
  如果 $sgc=0$，**那么** 继续
  如果 $sgc=1$ **那么**
    $ans\leftarrow ans\cup\{(r\pi+$反正切 $c,\ r\pi+$反正切 $d)\}$
    继续
  $\ell\leftarrow$ 最小值 $\{i\mid 2\leq i\leq m,\ \mathtt{sgc}_{i,m}(c^+)-\mathtt{sgc}_{i,m}(d^-)=1\}$
  $sr\leftarrow \mathtt{Root}(w_\ell,c,d)$
  根据定理 4.3，w_ℓ 在 $(c,\ d)$ 中恰有一个单根 sr。我们使用符号 $\mathtt{Root}(w_\ell,c,d)$ 表示 w_ℓ 在 (c,d) 中唯一的单根。
  如果 $F_\ell=QF_1$ *where* $Q\in\mathbb{Q}[y]$，**则**
    如果 Q *has no real roots on* (c,d)，**则**
      $ans\leftarrow ans\cup\{(r\pi+$反正切 $c, r\pi+$反正切 $d)\}$
    else
      继续
  else
    $u,v\leftarrow$ **有理端点**(w_1,w_ℓ,sr)
    将 $(c,\ u,\ \ell)$ 和 $(v,\ d,\ \ell)$ 添加到 $stack$
返回 ans

5. 整体隔离算法：考虑根的重数

本节中，我们将结合前几节的结果，给出一种用于隔离 MTP 所有实根的算法。此外，该算法还会计算每个根的重数。为此，我们首先介绍一个关于根的重数的命题。

命题 5.1。设 $g\in\mathbb{Q}[x,y]\backslash\mathbb{Q}$ 不可约。则除了 $x=0$ 之外，g（x，正切x）在其定义域中的所有实根都是单根。

证明。若 $g\in\mathbb{Q}[x]\cup\mathbb{Q}[y]$，结论显然成立。当 g是二元时，结论可由 [15, 定理 2.1] 得出。尽管该定理只

算法 6：有理端点

输入： $w_1(y) = F_1(r\pi + \text{反正切 } y, y)$，其中 $F_1(x,y) \in \mathbb{Q}[x,y]$，$r \in \mathbb{Q}$
$w_\ell(y) = F_\ell(r\pi + \text{反正切 } y, y)$，其中 $F_\ell(x,y) \in \mathbb{Q}[x,y]$
$sr = \texttt{Root}(w_\ell, c, d)$，$w_\ell$ 在 (c,d) 中恰好有一个单根 sr，且 $w_1(sr) \neq 0$
输出： 两个有理数 u 和 v ($c \le u < sr < v \le d$)，使得 w_1 在 $[u,v]$ 上没有根

```
1  n ← 1
2  当 n ≥ 1 时执行
3      计算一个上界多项式 W_{max, n} ∈ ℚ[y] 和一个下界多项式
         W_{min, n} ∈ ℚ[y] 对于 w_1(y)
4      q ← (c+d)/2
5      如果 (W_{max, n} has no roots on [c,d] and W_{max, n}(q) < 0) 或
6         (W_{min, n} has no roots on [c,d] and W_{min, n}(q) > 0) 那么
7          返回 c , d
8      else
9          n ← n + 1
10         如果 w_ℓ(q) = 0 那么
11             c ← (63q+c)/64, d ← (63q+d)/64
12         否则如果 sign(w_ℓ(c) w_ℓ(q)) = −1 则
13             d ← q
14         else
15             c ← q
```

声称 $g(x,$ 正切 $x)$ 在 $(0, \frac{\pi}{4}]$ 中的任意根都是单根，但其证明实际上表明，$g(x,$ 正切 $x)$ 在 $(\mathbb{R}\backslash\{0\})\backslash(\mathbb{Z}+\frac{1}{2})\pi$ 中的任意根都是单根，这正是我们所需要的。□

以下观察与引理 4.12 类似。下文给出的证明是按照 [7, 定理 7]的证明写成的。

命题 5.2. 假设 $p, q \in \mathbb{Q}[x,y]$ 互素，则 $p(x,$ 正切 $x)$ 和 $q(x,$ 正切 $x)$ 除 $x = 0$ 外没有公共根。

证明。在 [7, 定理 7]的证明中，将其中的“p, q”替换为本命题中的 p 和 q，将其中的“$f^*(x)$，$g^*(x)$”替换为 $p(x,$ 正切 $x)$和 $q(x,$ 正切 $x)$，并将其中的函数“trans”替换为正切函数，我们便可看出，若$\alpha \in (\mathbb{R}\backslash\{0\})\backslash(\mathbb{Z}+\frac{1}{2})\pi$ 是 $p(x,$ 正切 $x)$ 和 $q(x,$ 正切 $x)$的公共根，则 α 和正切α 都是代数的，这与定理 2.3 矛盾。□

那么，下面的定理显而易见。

定理 5.3。设 $g \in \mathbb{Q}[x,y]\backslash\mathbb{Q}$，且 g 在 $\mathbb{Q}[x,y]$ 中的因式分解为$g = c_0 \prod_{i=1}^{n} g_i^{p_i}$，其中 g_i 是互不相同的不可约因子，且 $c_0 \in \mathbb{Q}$。若$x_0 \in \mathbb{R}\backslash\{0\}$ 是 $g(x,$ 正切 $x)$ 的根，则存在唯一的 g_i，使得$g_i(x_0,$ 正切 $x_0) = 0$，且 x_0 的重数为 p_i。□

根据上述讨论，可以扩展算法2和算法4来计算每个根的重数：将 g 因式分解为 $c_0\prod_{i=1}^{n} g_i^{p_i}$，并对每个不可约因子 g_i调用算法2和算法4，则重数就是 g 中 g_i 的指数， $x=0$处除外。

注5.4。 在命题*5.1–5.2*和定理 *5.3* 中用余切函数替换正切函数，相应的结论也成立。证明过程都类似。

下一个引理用于检测 $f(x,$ 正弦 $x,$ 余弦 $x)$ 在 $k\pi\,(k\in\mathbb{Z})$ 处的可能的根，而函数 $g(\frac{x}{2},$ 正切 $\frac{x}{2})$ 无法处理这些根。

引理 5.5。 可以有效判定 $f(x,$ 正弦 $x,$ 余弦 $x)$ 对于 $f\in\mathbb{Q}[x,y,z]$ 是否恒等于零。此外，可以有效计算 $f(x,$ 正弦 $x,$ 余弦 $x)$ 在 $0,\ 2k\pi\,(k\in\mathbb{Z}\backslash\{0\}),\ (2k+1)\pi\,(k\in\mathbb{Z})$ 处根的重数。

证明。 将 $f(x,$ 正弦 $x,$ 余弦 $x)$ 改写为 $g(\frac{x}{2},$ 正切 $\frac{x}{2})$，其中 $g\in\mathbb{Q}[x,t]$ 满足方程（2.3）；那么， $f(x,$ 正弦 $x,$ 余弦 $x)$ 恒等于零，当且仅当 $g(\frac{x}{2},$ 正切 $\frac{x}{2})$ 恒等于零。由 [15] 中的引理2.3可知， $f(x,$ 正弦 $x,$ 余弦 $x)$ 恒等于零，当且仅当 g 是零多项式。

如果 $f(x,$ 正弦 $x,$ 余弦 $x)$ 不恒等于零，那么 $x=0$ 的根的重数有限。否则，随着 z 在$\mathbb{C}$ 中变化，解析函数 $f(z,$ 正弦 $z,$ 余弦 $z)$ 将恒等于零，因此 $f(x,$ 正弦 $x,$ 余弦 $x)$ 也将恒等于零。通过计算 $\mathrm{d}^n\mathrm{d}x^n f(x,$ 正弦 $x,$ 余弦 $x)|_{x=0}$，其中 $n=0$ 取 1、2、.....，直到对于某个 n_0， $\mathrm{d}^{n_0}\mathrm{d}x^{n_0} f(x,$ 正弦 $x,$ 余弦 $x)|_{x=0}\neq 0$。于是， n_0 就是 $x=0$的重数。

对于 $x=2k\pi$，当 $k\in\mathbb{Z}\backslash\{0\}$ 时， $f(x,$ 正弦 $x,$ 余弦 $x)|_{x=2k\pi}=0$ 当且仅当 $g(x,$ 正切 $x)|_{x=k\pi}=0$，并且 $2k\pi$ 关于 $f(x,$ 正弦 $x,$ 余弦 $x)$ 的根的重数等于 $k\pi$ 关于 $g(x,$ 正切 $x)$ 的根的重数。此外，我们声称 $g(k\pi,$ 正切 $k\pi)=0$ 当且仅当 t 是 g 的因子。“若”部分显然。假设 $g(k\pi,$ 正切 $k\pi)=0$，那么对于 g 的某个不可约因子 $h=\sum_{j=0}^{\deg(h,x)} a_j(t)x^j$，有$0=h(k\pi,$ 正切 $k\pi)=h(k\pi,0)$。由于 $k\pi$ 是超越的，对于任意 $j,\ a_j(0)=0$。因此， t 整除 h 和$h=rt$ （带有一个 $r\in\mathbb{Q}$），从而证明了这一结论。于是，根据定理 5.3， $2k\pi$ 的重数就是 g 的因式分解中 y 的指数。

同样，我们可以将 $f(x,$ 正弦 $x,$ 余弦 $x)$ 改写为 $\hat{}g(\frac{x}{2},$ 余切 $\frac{x}{2})$,–其中

$$\hat{g}(x,t)=f(2x,\frac{2t}{1+t^2},\frac{t^2-1}{1+t^2})(1+t^2)^{\deg(f,[y,z])}\in\mathbb{Q}[x,t].$$

然后，利用注5.4以及与上一段类似的讨论，可以观察到 $(2k+1)\pi$ 的根的重数等于 $\hat{}g$ 的因式分解中 t 的指数。

□

现在给出用于隔离 $f(x,$ 正弦 $x,$ 余弦 $x)$ 所有实根的主算法（算法 7）：

1. 判断 MTP $f(x,$ 正弦 $x,$ 余弦 $x)$ 是否恒等于零（第 2–3 行）。2. 使用引理 5.5 证明中的方法，计算点 $x = k\pi$ 的重数，其中 $k \in \mathbb{Z}$，并根据输入 f 计算多项式 $g \in \mathbb{Q}[x, y]$，使得 $y \nmid g$（第 5–12 行）。3. 计算 $\mathrm{LC}($ $g,\ x)$ 以及 $(a_0,\ b_0)$、$(a_{s+1},\ b_{s+})$ 的隔离区间

$_1),$

，形成潜在周期区间集 I，并对 g 进行因式分解（第 13–17 行）。4. 对每个不可约因子 g_i（其中 $\deg(g_i, y) \geq 1$），连同 I 一起调用算法 2，计算一些 k_1^*、k_2^*，并隔离 $(-k_2^*\pi - \pi/2, k_1^*\pi + \pi/2)$ 之外 $g_i(x,$ 正切 $x)$ 的实根（第 20–24 行）。5. 在 $\mathbb{Q}[x]$ 中隔离因子 g_i 的非零根（第 26–27 行）。注意，这些根对应于 $f(x,$ 正弦 $x,$ 余弦 $x)$ 唯一的非零代数根。我们将 $[0, 0]$ 以及上述步骤得到的区间称为“代数”区间，其他区间称为“超越”区间：前者表示 0 或代数根，后者表示超越根。代数区间可能与超越区间相交，但这不会造成问题，因为我们知道它们表示不同的根。6. 对每个因子 g_i（其中 $\deg(g_i, y) \geq 1$）的 $k_{1\cdot 2}^*$ 进行合并，使用一些 k_+ 和k_-来刻画“有界”根的界（第 22、24 和 29–32 行）。7. 对每个满足 $\deg(g_i, y) \geq 1$ 的因子 g_i，使用算法 4 在 $(k_-\pi - \pi/2,$ $k_+\pi + \pi/2)$ 中隔离 $g_i(x,$ 正切 $x)$ 的实根（第 33–38 行）。8. 返回所有结果。

此外，对于任意实数 $\alpha,\ \beta$，我们将区间 $[[\alpha, \beta]]$ 定义为

$$[[\alpha, \beta]] := \begin{cases} (\alpha, \beta), & \text{if } \alpha < \beta, \\ [\alpha, \beta], & \text{if } \alpha = \beta. \end{cases}$$

该符号说明用于算法 7 的伪代码中。

算法 7：主算法

输入 ： $f(x,\ \tau,\ \upsilon) \in \mathbb{Q}[x,\tau,\upsilon]$;
一个正有理数 ε ，用于指定隔离区间的最大长度。

输出:
由若干二元组组成的有限集 $\{([[\alpha_i,\beta_i]],m_i)\}_i$，其中 $\alpha_i,\beta_i \in$ (2 反正切($\mathbb{Q}$) $+\frac{\mathbb{Z}}{2}\pi$) $\cup\ \mathbb{Q}$, 使得对于对于每个 i, $f(x,$ 正弦 $x,$ 余弦 $x)$ 在 $[[\alpha_i,\beta_i]]$ 中有 1 个实根，且该根的重数为 m_i;
某些三元组 $\{([[\alpha_i,\beta_i]],(c_i,\ m_i)\ k_i)\}_i$ 构成的有限集，其中 $\alpha_i,\ \beta_i \in 2$ 反正切($\mathbb{Q}$) $+\frac{\mathbb{Z}}{2}\pi$，使得对于每个 i: 若 $k_i \geq 0$，则 $f(x,$ 正弦 $x,$ 余弦 $x)$ 在 $[[\alpha_i,\beta_i]]+2k\pi\,(k \geq k_i)$ 中恰有 c_i 个根，且每个根的重数为 m_i; 若 $k_i < 0$，则 $f(x,$ 正弦 $x,$ 余弦 $x)$ 在 $[[\alpha_i,\beta_i]]+2k\pi\,(k \leq k_i)$ 中恰有 c_i 个根，且每个根的重数为 m_i。

$ans_1 \leftarrow \emptyset$; $ans_2 \leftarrow \emptyset$; $g \leftarrow f(2x,\ \frac{2y}{1+y^2},\ \frac{1-y^2}{1+y^2})\cdot(y^2+1)^{\mathtt{tdeg}(f,\ [\tau,\upsilon])}$
若 g *is the zero polynomial* **则**
    错误 "$f(x,$ 正弦 $x,$ 余弦 $x)$ 恒等于零"
$gcot \leftarrow f(2x,\ \frac{2y}{1+y^2};\ \frac{y^2-1}{1+y^2})\cdot(1+y^2)^{\mathtt{tdeg}(f,[\tau,\upsilon])}$
令 m_1 为满足 $y^{m_1} \mid gcot$的最大正整数
如果 $m_1 \neq 0$ **那么**
    ans_2.add(($[\pi,\pi]$, (1, m_1), 0)); ans_2.add(($[\pi,\pi]$, (1, m_1) -1))
如果 $f(0,$ 正弦 $0,$ 余弦 $0) = 0$ **那么**
    $m_2 \leftarrow f(x,$ 正弦 $x,$ 余弦 $x)$ 在 0 处的重数; ans_1.add(($[0,0]$, m_2))
设 m_3 为满足 $y^{m_3} \mid g$ 的最大正整数
如果 $m_3 \neq 0$ **那么**
    ans_2.add(($[0,0]$, (1, m_3), 1)); ans_2.add(($[0,0]$, (1, m_3) -1)); $g \leftarrow g/y^{m_3}$
$(a_1,b_1)\ldots,(a_s,b_s) \leftarrow$ LC(g,x) 的隔离区间，使得反正切 b_i − 反正切 $a_i < \varepsilon/2\,(1 \leq i \leq s)$
$a_0 \leftarrow -\infty$; 令 $b_0 \in \mathbb{Q}$，使得反正切 $b_0 - (-\frac{\pi}{2}) < \frac{\varepsilon}{2}$且 $b_0 < a_1$。
$b_{s+1} \leftarrow +\infty$; 令 $a_{s+1} \in \mathbb{Q}$，使得 $\frac{\pi}{2}$ −反正切$a_{s+1} < \frac{\varepsilon}{2}$ 且 $a_{s+1} > b_s$。
$I \leftarrow [(a_0,b_0)\ldots,(a_{s+1},b_{s+1})]$
将 g 因式分解为 $c_0\prod_{i=1}^{n} g_i^{p_i}$
$ans_{21} \leftarrow \emptyset$; $ans_{22} \leftarrow \emptyset$; $k_- \leftarrow 0$; $k_+ \leftarrow 0$
对 i *from* 1 *to* n **执行**
    如果 $\deg(g_i,y) \geq 1$ **，则**
        k_1^*, $\{((\alpha_j,\ \beta_j)\ c_j)\}_j \leftarrow$ 使用 $g_i(x,y)$ 和 I 调用算法 2
        $k_+ \leftarrow$最大值$\{k_1^*,k_+\}$; ans_{21}.添加($\{((2\alpha_j,2\beta_j),(c_j,p_i))\}_j$)
        k_2^*, $\{((\alpha_j,\ \beta_j)\ c_j)\}_j \leftarrow$ 调用算法 2，传入 $g_i(-x,\ -y)$ 和 I
        $k_- \leftarrow \min\{-k_2^*,k_-\}$; ans_{22}.add($\{((-2\beta_j,\ -2\alpha_j),\ (c_j,\ p_i))\}_j$)
    else
        $\{[[\alpha_j,\beta_j]]\}_j \leftarrow g_i$ 的非零根的隔离区间，使其与 $[0,0]$ 不相交以及 $\beta_j - \alpha_j < \varepsilon/2$。
        ans_1.添加($\{([[2\alpha_j,2\beta_j]],\ p_i)\}_j$)
细化上述循环中**"else"**部分得到的区间，使它们彼此不相交。
对于 *each* $((\alpha^*,\beta^*),(c^*,p^*))$ *in* ans_{21} **执行**
    ans_2.add(((α^*, β^*), (c^*, p^*) $k_+ + 1$))
对于 *each* $((\alpha^*,\beta^*),(c^*,p^*))$ *in* ans_{22} **执行**
    ans_2.添加(((α^*, β^*), (c^*, p^*) $k_- - 1$))
对 k *from* k_- *to* k_+ **执行**
    对 i *from* 1 *to* n**执行**
        如果 $\deg(g_i,y) \geq 1$ **则**
            $\{[[\alpha_j,\beta_j]]\}_j \leftarrow$ 使用 $g_i(x,y)$ 和 k调用算法 4
            移除 $[0,0]$ 在 $\{[[\alpha_j,\beta_j]]\}_j$中; ans_1.add($\{([[2\alpha_j,2\beta_j]],p_i)\}_j$)
    细化上方循环中得到的区间 $[[\alpha_j,\beta_j]]$ ，使其中任意两个都不相交，并且 $\beta_j - \alpha_j < \varepsilon/2$。
返回 ans_1, ans_2

6. 实验

主算法（算法 7）通过 Maple2021 实现为名为 RootOfMTP 的工具。为了说明该工具的有效性和效率，我们准备了两类测试示例，其中一类收集自

一个来自文献，另一个则是随机生成的。一方面，我们通过一些数值方法测试该工具在这些示例上的正确性；另一方面，这些示例的实验结果表明，该工具功能强大，能够处理一些复杂示例。

6.1. 命令

该工具 RootOfMTP 可在以下网址下载：

https://github.com/lihaokun/RootOfMTP。

所有测试示例也都可以从该处下载。所有测试均在配备 128GB 内存、运行 Windows 11 的 16 核英特尔酷睿 i7-12900KF@3.20GHz 上进行。

下面通过一个单根示例说明如何使用 RootOfMTP 。假设我们想用 RootOfMTP隔离以下 MTP 的所有实根：

$$x\sin x+\cos x-1.$$

我们只需在 Maple2021中运行以下命令：

```
read ".../RootOfMTP.mpl";
RootOfMTP(x*sin(x)+cos(x)-1,x,1);
```

其中，第一个命令用于读取文件，第二个命令的输入包括一个 MTP、该 MTP 的变量，以及一个有理数 $\epsilon>0$，用于指定隔离区间的最大长度。输出为

```
For every k <= -1 (k in Z), 2kPi+(0) is a real root with multiplicity 1.
For every k >= 1 (k in Z), 2kPi+(0) is a real root with multiplicity 1.
There is 1 real root with multiplicity 2 at 0.For every k >= 2 (k in Z),
 there is 1 real root with multiplicity 1in (2kPi+(2*arctan(63/16)), 2kPi+(Pi)).
For every k <= -2 (k in Z), there is 1 real root with multiplicity 1
in (2kPi+(-Pi), 2kPi+(-2*arctan(63/16))).
There is 1 real root with multiplicity 1 in each open interval of the list
[[-5/2*Pi-2*arctan(29666650363354128505/36893488147419103232),
-5/2*Pi-2*arctan(7242537696610193/9007199254740992)],
[-1/2*Pi-2*arctan(7030038563941/17592186044416),
-1/2*Pi-2*arctan(1474193477312957037 7/36893488147419103232)],
[1/2*Pi+2*arctan(1474193477312957037 7/36893488147419103232),
1/2*Pi+2*arctan(7030038563941/17592186044416)],
[5/2*Pi+2*arctan(7242537696610193/9007199254740992),
5/2*Pi+2*arctan(29666650363354128505/36893488147419103232)]].
```

6.2. 文献中的示例

我们从文献中收集了 21 个测试示例 [21, 22]。由于其中一些示例是有理函数不等式，因此需要将它们转换为 MTP。以 [21,例4] 为例：

$$\frac{\sin^2 x}{x^2}+\frac{\tan x}{x}>\frac{x^2}{\sin^2 x}+\frac{x}{\tan x}.$$

首先，将右侧的项移至左侧。其次，将 tan x 替换为正弦 x 余弦 x。第三，消去分母，得到一个 MTP： $-x^4$ 余弦~~x~~ $-x^3$ 正弦 x 余弦^2 $x+x$ 正弦^3 $x+$ 正弦^4 x 余弦 x。

实验结果见表 2。请注意，这些示例都比较简单。界 k_- 和 k_+ 接近 0， `RootOfMTP`可以在 2 秒内处理每一个示例。

k_-	一个整数，使得 $2k_-\pi-\pi$ 是所有有界实根的下界
k_+	一个整数，使得 $2k_+\pi+\pi$ 是所有有界实根的上界
PRoots	周期实根
BRoots	有界实根
wSeq	四个已计算的弱傅里叶序列的长度
wTime	计算四个弱傅里叶序列所用的时间
Time	总计算时间
$n(m)$	周期实根簇的数量 / 有界实根的数量（每个根的重数）
-	没有周期或有界实根 / 无需计算弱傅里叶序列

表1：符号说明。

示例	k_-	k_+	PRoots	BRoots	wSeq	wTime	Time	示例	k_-	k_+	PRoots	BRoots	wSeq	wTime	Time
[22, 例1]	-1	1	4(1)	7(1)	6;5;5;6	0.005秒	0.251秒	[22, 例2]	0	0	-	1(5)	6;5;5;6	0.006秒	0.054秒
[22, 例4]	0	0	1(2)	-	-	0s	0.003秒	[21, 例1, $k=1$]	0	0	4(1)	1(3)	8;7;7;8	0.009秒	0.075秒
[21, 例1, $k=2$]	0	0	4(1)	1(5)	8;7;7;8	0.009秒	0.058秒	[21, 例1, $k=3$]	0	0	4(1)	1(3);2(1)	8;7;7;8	0.01秒	0.085秒
[21, 例2, $c=9/45$]	-1	1	4(1)	1(6);6(1)	21;43;43;21	0.124秒	1.089秒	[21, 例2, $c=8/45$]	-1	1	4(1)	1(8);6(1)	21;43;43;21	0.084秒	1.065秒
[21, 例2, $c=7/45$]	-1	1	4(1)	1(6);4(1)	21;43;43;21	0.123秒	1.013秒	[21, 例2, $c=6/45$]	-1	1	4(1)	1(6);4(1)	21;43;43;21	0.074秒	0.964秒
[21, 例3]	-3	3	4(1)	1(5);12(1)	28;51;51;28	0.238秒	1.719秒	[21, 例4]	0	0	4(1)	1(8)	6;6;6;6	0.005秒	0.246秒
[21, 例5]	0	0	-	2(1)	7;7;7;7	0.007秒	0.098秒	[21, 例6]	0	0	4(1)	1(7)	17;15;15;17	0.053秒	0.18秒
[21, 例7]	0	0	-	1(5)	6;5;5;6	0.005秒	0.048秒	[21, 例8]	-1	1	4(1)	1(2);4(1)	4;3;3;4	0.001秒	0.129秒
[21, 例9]	0	1	-	1(2)	5;5;5;5	0.005秒	0.138秒	[21, 例10]	0	0	4(1)	1(3)	8;7;7;8	0.056秒	0.105秒
[21, 例11]	0	0	-	1(1)	4;4;4;4	0.003秒	0.042秒	[21, 例12]	-1	1	4(1)	8(1)	7;8;8;7	0.006秒	0.187秒
[21, 例13]	0	0	4(1)	1(3)	8;7;7;8	0.007秒	0.053秒								

表 2：文献中的示例的结果（秒：秒）。

6.3. 随机示例

现在，我们展示随机生成的多项式的实验结果。一个随机生成的多项式

$$\mathbf{rp}(d,p,c)\in\mathbb{Q}[x,y,z]$$

构造方法如下：首先，从集合 $\{x^{a_1}y^{a_2}z^{a_3}\mid a_i\in\mathbb{N},\ a_1+a_2+a_3\le d\}$ 中独立选取若干单项式，使得

每个单项式被选取的概率为 p，被舍弃的概率为$1-p$。然后，为每个选取的单项式随机赋予介于 $-c$ 和 c 之间的系数，并将它们相加得到一个多项式。我们以这种方式从 16 个不同类别中生成 160 个随机多项式 F_1, ……, F_{160}，其中

$F_1, \ldots, F_{10} \in$ *rp(5, 0.1, 10)* $F_{11}, \ldots, F_{20} \in$ *rp(10, 0.015, 10),* $F_{21}, \ldots, F_{30} \in$ *rp(10, 0.05, 10)* $F_{31}, \ldots, F_{40} \in$ *rp(10, 0.1, 10),*$F_{41}, \ldots,$ $F_{50} \in$ *rp(15, 0.01, 10)* $F_{51}, \ldots, F_{60} \in$ *rp(15, 0.05, 10),*$F_{61}, \ldots, F_{70} \in$ *rp(15, 0.1, 10)* $F_{71}, \ldots, F_{80} \in$ *rp(20, 0.0025, 10),*$F_{81}, \ldots, F_{90} \in$ *rp(5, 0.1, 1000)* $F_{91}, \ldots, F_{100} \in$ *rp(10, 0.015, 1000),*$F_{101}, \ldots, F_{110} \in$ *rp(10, 0.05, 1000)* $F_{111}, \ldots, F_{120} \in$ *rp(10, 0.1, 1000),*$F_{121}, \ldots, F_{130} \in$ *rp(15, 0.01, 1000)* $F_{131}, \ldots, F_{140} \in$ *rp(15, 0.05, 1000),*$F_{141}, \ldots, F_{150} \in$ *rp(15, 0.1, 1000)* $F_{151}, \ldots, F_{160} \in$ *rp(20, 0.0025, 1000).*

然后，我们在 $F_1, \ldots, F_{160}$ 中分别用正弦 x 和余弦 x 替换 y 和 z，得到 MTP $M_1, \ldots, M_{160}$。

实验结果见表3。该工具 `RootOfMTP`在这些随机示例中表现出色：每个示例都能在 3000 秒内求解，其中 93% 的示例在 1 分钟内求解完成。我们注意到，计算弱傅里叶序列的效率不高。在 59% 的示例中，计算弱傅里叶序列所用的时间超过了总计算时间的三分之一。另一个令人惊讶的发现是，即使面对这些更难的示例，界 k_- 和 k_+ 仍然很小。

最后，我们给出图3，以展示 `RootOfMTP`在所有示例（无论是来自文献还是随机生成）上的表现。横轴表示时间，纵轴表示在相应时间内已求解的示例数量。

7. 结论

本文首次提出了一种完整算法，用于“隔离”给定 MTP 的所有实根。更准确地说，该算法计算

- 任意小的区间，其并集覆盖给定 MTP 的所有实根（根和区间都可能有可数多个），以及
- 每个区间内互异根的确切数量，以及其中每个根的重数。

示例	k_-	k_+	PRoots	BRoots	wSeq	wTime	Time	示例	k_-	k_+	PRoots	BRoots	wSeq	wTime	Time
M_1	0	0	-	2(1)	27;27;27;27	0.186秒	0.532秒	M_2	0	0	4(1)	4(1)	26;26;26;26	0.148秒	0.634秒
M_3	0	0	4(1)	2(1)	22;24;24;22	0.088秒	0.286秒	M_4	-3	3	4(1)	15(1)	18;14;14;18	0.032秒	0.392秒
M_5	0	2	8(1)	11(1)	16;20;20;16	0.054秒	0.542秒	M_6	0	0	4(1)	4(1)	29;53;53;29	0.212秒	0.62秒
M_7	0	0	7(1)	2(1)	20;20;20;20	0.078秒	0.361秒	M_8	-2	2	4(1)	11(1)	49;26;26;49	0.215秒	1.402秒
M_9	-1	1	6(1)	7(1)	39;21;21;39	0.146秒	0.878秒	M_{10}	0	0	-	-	13;12;12;13	0.013秒	0.101秒
M_{11}	0	0	10(1)	5(1)	99;88;88;99	1.325秒	2.167秒	M_{12}	0	0	5(1);2(2)	1(1)	-	0s	0.018秒
M_{13}	0	0	3(1)	4(1)	53;54;54;53	0.421秒	1.367秒	M_{14}	0	0	2(1)	1(4)	-	0s	0.009秒
M_{15}	-1	1	4(1)	9(1)	87;192;192;87	2.765秒	8.305秒	M_{16}	-1	0	12(1)	15(1)	175;174;174;175	4.416秒	13.173秒
M_{17}	0	0	2(2)	1(2)	-	0s	0.001秒	M_{18}	0	0	3(2)	1(3);1(1)	20;23;23;20	0.06秒	0.238秒
M_{19}	-5	1	10(1)	29(1)	70;212;212;70	3.87秒	15.224秒	M_{20}	0	0	3(8)	1(10)	-	0s	0.001秒
M_{21}	0	1	-	2(1)	107;107;107;107	1.566秒	7.036秒	M_{22}	0	0	-	1(2);2(1)	79;79;79;79	1.001秒	3.233秒
M_{23}	-1	1	12(1)	16(1)	216;195;195;216	5.432秒	23.089秒	M_{24}	-4	4	4(1)	20(1)	104;213;213;104	4.035秒	22.34秒
M_{25}	0	0	8(1)	4(1)	107;243;243;107	5.278秒	9.108秒	M_{26}	-1	2	8(1)	16(1)	193;154;154;193	5.656秒	19.36秒
M_{27}	-2	2	8(1)	26(1)	143;161;161;143	4.508秒	19.111秒	M_{28}	0	0	-	1(2);1(1)	92;92;92;92	1.49秒	2.893秒
M_{29}	-1	1	8(1)	12(1)	108;177;177;108	3.516秒	10.417秒	M_{30}	0	1	4(1)	5(1)	99;233;233;99	4.859秒	10.586秒
M_{31}	0	0	4(1)	2(1)	103;247;247;103	5.227秒	7.974秒	M_{32}	0	1	4(1)	6(1)	105;249;249;105	5.022秒	14.39秒
M_{33}	-1	1	8(1)	12(1)	250;97;97;250	7.234秒	19.997秒	M_{34}	0	0	4(1)	4(1)	270;111;111;270	6.47秒	11.022秒
M_{35}	-1	1	8(1)	12(1)	535;111;111;535	55.14秒	83.45秒	M_{36}	0	0	4(1)	1(1)	259;106;106;259	8.499秒	10.169秒
M_{37}	0	1	4(1)	3(1)	103;260;260;103	7.776秒	11.775秒	M_{38}	0	0	4(1)	5(1)	100;100;100;100	1.575秒	4.608秒
M_{39}	0	1	4(1)	4(1)	103;103;103;103	2.115秒	6.234秒	M_{40}	-1	1	4(1)	7(1)	232;104;104;232	4.875秒	12.368秒
M_{41}	0	0	-	1(1)	74;90;90;74	1.325秒	1.669秒	M_{42}	-1	1	4(1)	1(5);10(1)	521;204;204;521	43.424秒	84.569秒
M_{43}	0	0	4(1)	1(6);3(1)	106;51;51;106	0.787秒	1.46秒	M_{44}	-6	6	8(1)	1(6);53(1)	463;382;382;463	44.613秒	374.288秒
M_{45}	0	0	4(1)	1(3);2(1)	183;611;611;183	97.672秒	116.082秒	M_{46}	-1	1	8(1)	1(2);13(1)	283;95;95;283	10.331秒	29.126秒
M_{47}	0	0	8(1)	2(1)	167;218;218;167	13.509秒	17.824秒	M_{48}	-1	1	10(1)	13(1)	336;538;538;336	70.131秒	190.27秒
M_{49}	0	0	2(1)	1(3)	84;91;91;84	1.13秒	2.598秒	M_{50}	-1	2	11(1)	17(1)	141;151;151;141	5.713秒	19.468秒
M_{51}	-2	1	8(1)	14(1)	200;905;905;200	230.777秒	610.059秒	M_{52}	0	0	-	1(2);2(1)	223;223;223;223	7.84秒	19.67秒
M_{53}	0	0	4(1)	1(2);5(1)	709;222;222;709	140.087秒	173.988秒	M_{54}	-1	1	8(1)	10(1)	605;787;787;605	233.809秒	518.236秒
M_{55}	0	1	-	2(1)	197;197;197;197	7.109秒	25.48秒	M_{56}	0	0	4(1)	1(4)	219;219;219;219	6.28秒	16.779秒
M_{57}	0	0	-	4(1)	223;223;223;223	7.016秒	25.085秒	M_{58}	-1	1	8(1)	1(2);14(1)	701;593;593;701	126.757秒	368.925秒
M_{59}	0	0	-	1(2);3(1)	230;230;230;230	10.187秒	24.844秒	M_{60}	-1	1	8(1)	12(1)	1133;219;219;1133	382.221秒	806.921秒
M_{61}	-1	0	4(1)	7(1)	245;650;650;245	60.368秒	153.185秒	M_{62}	0	0	-	4(1)	233;233;233;233	11.5秒	32.585秒
M_{63}	-1	1	4(1)	8(1)	243;617;617;243	50.81秒	146.751秒	M_{64}	-1	1	4(1)	7(1)	565;236;236;565	44.672秒	97.59秒
M_{65}	-7	7	4(1)	31(1)	234;537;537;234	43.514秒	284.004秒	M_{66}	0	1	4(1)	1(2);8(1)	237;597;597;237	52.063秒	121.855秒
M_{67}	-1	1	8(1)	13(1)	236;616;616;236	53.379秒	171.749秒	M_{68}	-1	1	4(1)	10(1)	242;631;631;242	50.009秒	162.565秒
M_{69}	-1	0	4(1)	5(1)	612;234;234;612	38.078秒	91.951秒	M_{70}	-1	0	4(1)	5(1)	609;238;238;609	32.149秒	102.304秒
M_{71}	0	0	3(1)	1(8);2(1)	164;164;164;164	3.571秒	6.428秒	M_{72}	-1	0	3(6);2(2);4(1)	1(13);6(1)	30;32;32;30	0.13秒	5.488秒
M_{73}	0	0	3(4);2(1)	1(6);3(1)	122;122;122;122	2.128秒	4.072秒	M_{74}	0	0	3(5);4(1)	1(7);5(1)	120;309;309;120	11.067秒	23.141秒
M_{75}	0	0	3(4)	1(4)	86;96;96;86	1.869秒	2.906秒	M_{76}	-1	0	15(1)	9(1)	1005;385;385;1005	338.574秒	940.537秒
M_{77}	-1	0	3(2);2(3);4(1)	1(4);2(1)	34;42;42;34	0.274秒	0.819秒	M_{78}	0	0	3(3);4(1)	1(7);3(1)	168;437;437;168	36.463秒	45.945秒
M_{79}	0	0	7(1)	1(2);1(1)	278;803;803;278	161.677秒	170.291秒	M_{80}	0	0	13(1)	1(4)	767;1009;1009;767	515.406秒	518.845秒
M_{81}	-1	0	-	4(1)	23;20;20;23	0.07秒	0.378秒	M_{82}	0	0	4(1)	2(1)	31;22;22;31	0.095秒	0.298秒
M_{83}	0	0	8(1)	3(1)	50;53;53;50	0.35秒	0.787秒	M_{84}	0	0	-	1(2);3(1)	17;18;18;17	0.104秒	0.532秒
M_{85}	0	0	4(1)	1(2);2(1)	16;14;14;16	0.04秒	0.15秒	M_{86}	0	0	-	3(1)	28;28;28;28	0.185秒	0.739秒
M_{87}	-1	1	11(1)	14(1)	17;18;18;17	0.049秒	0.61秒	M_{88}	0	0	-	1(2);1(1)	30;30;30;30	0.188秒	0.588秒
M_{89}	0	0	-	1(1)	19;19;19;19	0.109秒	0.355秒	M_{90}	-1	1	4(1)	6(1)	28;51;51;28	0.241秒	1.035秒
M_{91}	-1	0	-	1(2);1(1)	58;49;49;58	0.69秒	2.015秒	M_{92}	0	0	3(1)	1(7)	55;55;55;55	0.333秒	1.269秒
M_{93}	0	0	5(2);4(1)	1(3);2(1)	20;17;17;20	0.062秒	0.243秒	M_{94}	0	0	5(2)	1(6)	-	0s	0.002秒
M_{95}	0	0	3(5)	1(9)	-	0s	0.0s	M_{96}	0	0	6(1)	1(6);1(1)	155;67;67;155	1.853秒	2.914秒
M_{97}	0	0	3(2);2(3)	1(5)	-	0s	0.006秒	M_{98}	0	0	13(1)	5(1)	46;38;38;46	0.24秒	0.667秒
M_{99}	-10	10	6(1)	1(3);24(1)	5;5;5;5	0.003秒	0.818秒	M_{100}	0	0	4(1)	1(4);4(1)	28;48;48;28	0.205秒	0.627秒
M_{101}	0	0	-	1(2)	88;88;88;88	1.345秒	3.151秒	M_{102}	0	13	4(1)	1(3);78(1)	320;88;88;320	11.792秒	95.276秒
M_{103}	-7	7	8(1)	1(2);58(1)	327;87;87;327	15.814秒	98.576秒	M_{104}	-3	2	12(1)	32(1)	202;307;307;202	11.731秒	66.217秒
M_{105}	-1	0	7(1)	1(3);3(1)	94;222;222;94	4.1s	8.278秒	M_{106}	-1	1	8(1)	9(1)	223;98;98;223	8.054秒	16.822秒
M_{107}	-1	0	8(1)	6(1)	335;242;242;335	14.961秒	29.263秒	M_{108}	0	0	8(1)	5(1)	179;175;175;179	8.592秒	11.608秒
M_{109}	-2	2	8(1)	21(1)	237;100;100;237	4.992秒	28.19秒	M_{110}	-1	0	-	1(1)	68;68;68;68	0.871秒	3.634秒
M_{111}	-1	1	4(1)	7(1)	110;245;245;110	5.441秒	20.069秒	M_{112}	-1	0	-	3(1)	82;82;82;82	1.066秒	2.765秒
M_{113}	-1	1	8(1)	13(1)	266; 286; 286; 266	12.064秒	42.789秒	M_{114}	-1	0	4(1)	5(1)	107; 245; 245; 107	6.711秒	12.651秒
M_{115}	-1	1	4(1)	7(1)	104;253;253;104	5.52秒	15.223秒	M_{116}	-1	4	8(1)	24(1)	271;271;271;271	14.857秒	66.69秒
M_{117}	-1	0	-	1(1)	104;104;104;104	1.941秒	5.852秒	M_{118}	-1	1	8(1)	8(1)	431;111;111;431	32.513秒	49.383秒
M_{119}	-1	0	4(1)	6(1)	111;279;279;111	11.232秒	19.781秒	M_{120}	0	0	-	-	107;107;107;107	1.763秒	3.846秒
M_{121}	-1	1	4(1)	8(1)	397;151;151;397	13.506秒	39.123秒	M_{122}	0	0	11(1)	1(10);4(1)	1443;182;182;1443	1273.551秒	1766.713秒
M_{123}	-6	0	17(1)	1(3);30(1)	833;543;543;833	267.967秒	1832.919秒	M_{124}	-68	1	6(1)	1(3);141(1)	151;623;623;151	88.491秒	1008.551秒
M_{125}	0	0	10(1)	3(1)	429;389;389;429	43.514秒	65.364秒	M_{126}	0	1	8(1)	6(1)	499;981;981;499	292.23秒	870.745秒
M_{127}	-1	1	2(2);4(1)	1(2);8(1)	311;129;129;311	25.591秒	35.585秒	M_{128}	0	0	-	3(1)	32;32;32;32	0.396秒	0.928秒
M_{129}	-1	1	8(1)	11(1)	132;132;132;132	2.595秒	12.191秒	M_{130}	-1	0	10(1)	1(3);10(1)	329;448;448;329	48.12秒	93.598秒
M_{131}	0	0	8(1)	5(1)	687;661;661;687	153.159秒	249.382秒	M_{132}	-1	0	4(1)	5(1)	225;560;560;225	40.575秒	79.235秒
M_{133}	0	0	-	4(1)	217;217;217;217	12.072秒	23.128秒	M_{134}	-1	1	8(1)	10(1)	513;1395;1395;513	1017.194秒	2496.384秒
M_{135}	-1	1	4(1)	6(1)	547;225;225;547	40.101秒	99.09秒	M_{136}	-1	1	8(1)	12(1)	649;533;533;649	106.612秒	259.645秒
M_{137}	0	1	4(1)	6(1)	623;241;241;623	58.741秒	98.687秒	M_{138}	-3	3	8(1)	1(2);28(1)	503;459;459;503	72.135秒	325.427秒
M_{139}	0	0	4(1)	4(1)	595;607;607;595	105.192秒	138.137秒	M_{140}	0	0	-	2(1)	215;215;215;215	8.132秒	26.755秒
M_{141}	0	0	4(1)	4(1)	242;621;621;242	60.554秒	102.052秒	M_{142}	-23	1	8(1)	18(1)	240;1361;1361;240	574.013秒	1821.513秒
M_{143}	-1	0	4(1)	6(1)	245;245;245;245	10.063秒	55.821秒	M_{144}	-1	1	4(1)	9(1)	227;586;586;227	51.978秒	140.508秒
M_{145}	0	0	-	1(1)	246;246;246;246	10.292秒	32.055秒	M_{146}	0	1	8(1)	8(1)	226;607;607;226	67.49秒	115.328秒
M_{147}	-4	4	4(1)	18(1)	247;647;647;247	55.513秒	278.322秒	M_{148}	-1	0	4(1)	4(1)	245;645;645;245	69.837秒	131.675秒
M_{149}	-1	0	8(1)	7(1)	234;630;630;234	73.299秒	138.112秒	M_{150}	0	1	12(1)	9(1)	1117;610;610;1117	542.28秒	1267.414秒
M_{151}	0	0	3(6);6(1)	1(13);2(1)	46;46;46;46	0.604秒	1.933秒	M_{152}	0	0	11(1)	1(4);2(1)	154;85;85;154	455.72秒	663.41秒
M_{153}	0	0	4(1)	1(3);1(1)	170;430;430;170	27.564秒	43.251秒	M_{154}	0	0	3(10);2(5)	1(12)	-	0s	0.023秒
M_{155}	0	0	3(5);4(1)	1(5);2(1)	25;35;35;25	0.163秒	0.568秒	M_{156}	0	0	2(2);8(1)	4(1)	200;635;635;200	235.405秒	275.278秒
M_{157}	-1	1	4(1)	6(1)	433;213;213;433	15.891秒	26.104秒	M_{158}	0	0	2(3);4(1)	5(1)	94;138;138;94	3.508秒	9.013秒
M_{159}	-1	15	6(1)	64(1)	345;145;145;345	12.958秒	168.032秒	M_{160}	-1	1	11(1);2(2)	1(3);10(1)	203;279;279;203	13.479秒	30.66秒

表3：随机示例的结果（秒：秒）。

因此，据我们所知，这得出了首个能够判定给定 MTP 是否存在实根，或是否存在无穷多个实根的完整算法。算法的正确性揭示了一个具有数学意义的现象：在距离零点足够远的每个周期内，实根都具有相同的数量、相同的根的重数多重集，甚至在各周期内还具有相似的相对位置。

该算法已实现，并通过大量示例测试来说明其效率。

与文献中的某些其他方法只关注某个有界区间内的实根不同，我们的（子）算法能够检测

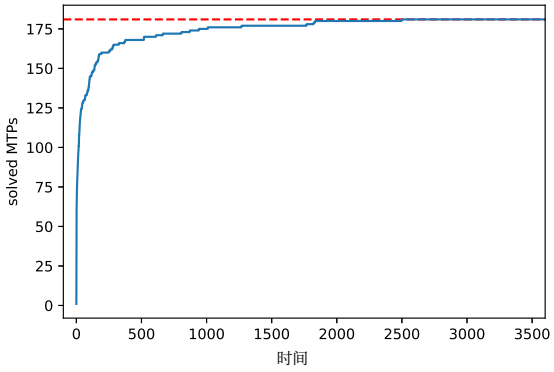

图3：给定时间内已求解的MTP数量（s：秒）。

并“隔离”任意形如 $[2k_-\pi-\pi, 2k_+\pi+\pi]$ 的有界区间，以及形如 $(2k_+\pi+\pi, \infty)$ 和 $(-\infty, -2k_-\pi-\pi)$ 的无界区间内的根，其中 $k_\pm$ 足够大。基于此，我们的方法可以轻松修改，以便在任意形如 (a, b)、$[a, b)$、$(a, b]$ 或 $[a, b]$ 的区间 I 内“隔离”并计算 MTP 的所有实根的重数，其中 $a, b \in \mathbb{Q} \cup \{\pm\infty\}$：

- 选择一个区间 $I_\mathrm{e} \supset I$，使子算法能够自然地“隔离” I_e 中的实根。

- 如果某些“超越”区间覆盖了 I 的某些端点，则将它们替换为一些细化子区间，每个子区间都不覆盖任何端点，但恰好包含一个超越根。

- 此后，每个覆盖 I 某些端点的“隔离”区间都是代数的，且恰好包含一个根（该根为 0 或代数数）。如果存在此类区间，则判断被覆盖的端点是否为根。

- 如果被覆盖的端点不是根，则通过二分法细化这些覆盖端点的区间，使细化后的区间不再覆盖端点。否则，如果它们分别覆盖 a 或 b，则将其细化至 $[a, a]$ 或 $[b, b]$。

这样，我们就能计算每个“隔离”区间内实根的精确个数，以及每个实根的重数。基于此，我们可以

进一步证明区间 I 上的 MTP不等式，前提是使用以下简单结论：

命题7.1。 对于任意不恒为零的 *MTP* f，以及任意正长度区间 I（有界或无界），

- $f > 0$ 在 I 上当且仅当 f 在 I 中没有根，且对于某个 $x_0 \in I$， $f(x_0) > 0$；
- $f \geq 0$ 在 I 上当且仅当 f 在 I 内部的每个根（若有）的重数均为偶数，且对于某个 $x_0 \in I$， $f(x_0) > 0$。

由此可以导出一种算法，用于证明端点属于 $\mathbb{Q} \cup \{\pm\infty\}$ 的任意区间（有界或无界）上的 MTP不等式。据我们所知，这是首个能够证明无界区间上 MTP 不等式的算法。

致谢

本工作得到国家自然科学基金委员会资助（项目编号：61732001和12071467），以及国家重点研发计划资助（项目编号：2018YFA0306702）。

参考文献

[1] G. E. Collins，利用柱形代数分解对实闭域进行量词消去，载于：《自动机理论与形式语言》，Springer，1975年，第134–183页。[2] D. P. Mitchell，利用区间算术进行稳健射线求交，载于：Graphics Interface论文集，第90卷，1990年，第68–74页。[3] A. Strzebo´nski，指数-对数函数的实根隔离，载于：国际符号与代数计算研讨会论文集，2008年，第303–314页。[4] A. Strzebo´nski，指数-对数-反正切函数的实根隔离，《符号计算杂志》47 (3) (2012) 282–314。[5] A. Strzebonski，温和初等函数的实根隔离，载于：国际符号与代数计算研讨会论文集，2009年，第341–350页。

[6]M. Achatz，S. McCallum，V. Weispfenning，判定多项式-指数问题，载于：国际符号与代数计算研讨会论文集，2008年，第215–222页。[7]S. McCallum，V. Weispfenning，判定多项式-超越问题，《符号计算杂志》47 (1) (2012) 16–31。[8] G. E. Collins，R. Loos，通过微分进行多项式实根隔离，载于：第三届ACM符号与代数计算研讨会论文集，1976年，第15–25页。[9] A. B. Shidlovskii，《超越数》，德古意特出版社，2011年。[10] A. Maignan，求解一维和二维指数多项式系统，载于：国际符号与代数计算研讨会论文集，1998年，第215–221页。[11] A. Strzebo´nski，第一变量中超越系统的柱形分解，《符号计算杂志》46 (11) (2011) 1284–1290。[12] M. Xu，Z.-B. Li，L. Yang，一类指数多项式公式的量词消去，《符号计算杂志》68 (2015) 146–168。[13] C.-C. Huang，J.-C. Li，M. Xu，Z.-B. Li，通过排除法和微分进行多重幂的正根隔离，《符号计算杂志》85 (2018) 148–169。[14] D. Wang，J. Xu，一种用于隔离某些根式实根的符号-数值算法，《计算与应用数学杂志》391 (2021) 113424。[15] S. Chen，Z. Liu，混合三角-多项式不等式的自动证明，《符号计算杂志》101 (2020) 318–329。[16] C. Mortici，Wilker-Cusa-Huygens不等式的自然方法，《数学不等式及其应用》14 (3) (2011) 535–541。[17]C.-P. Chen，反三角函数和反双曲函数的尖锐Wilker型与Huygens型不等式，《积分变换与特殊函数》23 (12) (2012) 865–873。

[18] G. Bercu, 《与三角函数相关的重要不等式的 Pad´e 逼近》, 《不等式与应用杂志》2016 年第 1 期（2016）, 1–11。[19] G. Bercu, 《三角不等式的自然方法——Pad´e 逼近》, 《数学不等式期刊》11（1）（2017）, 181–191。[20]M. Makragic, 《证明一些混合双曲-三角多项式函数不等式的方法》, 《数学不等式期刊》11（3）（2017）, 817–829。[21] S. Chen、Z. Liu, 《利用泰勒展开自动证明三角函数不等式》, 《系统科学与数学科学杂志》36（8）（2016）, 1339–1348。[22] S. Chen、X. Ge, 《无界情形下混合三角-多项式不等式的自动证明》, arXiv 预印本 arXiv:2204.01481（2022）。[23] I. R. S. Shafarevich、N. Fel'dman 等, 《数论 IV：超越数》, 第 4 卷, 施普林格科学与商业媒体, 1998年。[24] S. McCallum, 《柱面代数分解的一种改进投影运算》, 载于：《量词消去与柱面代数分解》, Springer, 1998年, 第 242–268 页。[25] B. Xia、T. Zhang, 《利用区间算术进行实数解隔离》, 《计算数学与应用》52（6–7）（2006）, 853–860。