arXiv:2304.01191v1 [cs.DM] 3 Apr 2023

Fast 数值多变量多点求值 n

Sumanta Ghosh * Prahladh Harsha† SimaoHerdade‡ Mrinal Kumar†
Ramprasad Saptharishi†

摘要

我们为有理数、实数和复数域上的多变量多点求值问题设计了近线性时间数值算法。我们同时考虑算法的精确版本和近似版本。算法的输入包括：(1) 一个m元多项式f的系数，该多项式关于每个变量的次数为d；(2) 点$\mathbf{a}_1, \ldots, \mathbf{a}_N$，其中每个点的坐标值均不超过 1，且位复杂度为$s$。

近似版本：额外给定精度参数t后，算法计算有理数 $\beta_1, \ldots, \beta_N$，使得 $|f(\mathbf{a}_i) - \beta_i| \leq 1/2^t$ 对所有i均成立，且其运行时间为$((N\text{有限域} + d^m)(s+t))^{1+o(1)}$，适用于所有$m$ 以及所有充分大的d。

精确版本（当定义在有理数上时）：此外，给定所有求值结果的位复杂度上界c，该算法可在时间$((Nm + d^m)(s+c))^{1+o(1)}$ 内计算出有理数$f(\mathbf{a}_1)$、……、$f(\mathbf{a}_N)$，其中m 为任意值，且d足够大。

我们的结果也自然地推广到输入定义在实数域或复数域上的情形，只要采用适用于此类域中域元素表示的标准表示模型。

在本项工作之前，已知在任意无限域上进行多变量多点求值（精确或近似）的近线性时间算法似乎仅适用于单变量多项式；这一算法由 Moroz 在近期的一项工作中发现[Mor21]。本项工作将这一结果从单变量情形推广到多元情形。然而，我们的算法基于一些在概念上似乎与 Moroz[Mor21] 不同的思路，且关键依赖于 Bhargava、Ghosh、Guo、Kumar 和 Umans[BGGKU22] 近期提出的有限域上多变量多点求值算法，以及计算数论中有理数重构和快速中国剩余定理的已知高效算法。

*加州理工学院，帕萨迪纳，美国。 besusumanta@gmail.com

†塔塔基础研究院，孟买，印度。 {prahladh, mrinal, ramprasad}@tifr.res.in。研究由印度政府原子能部通过项目12-R&D-TFR-5.01-0500提供资助。第二作者的研究部分得到了Google印度研究奖的支持。

‡Yahoo研究院，旧金山，美国。 simaoherdade@gmail.com

1 引言

本文研究如何为以下自然的计算问题设计快速算法。

> 给定一个m元多项式f，其在底层域 $\mathbb{F}$ 上的每个变量的次数均小于d，并以系数列表的形式给出；另给定（任意）求值点$\mathbf{a}_1, \mathbf{a}_2, \ldots\ldots \mathbf{a}_N \in \mathbb{F}^m$，输出每个$i$对应的$f(\mathbf{a}_i)$。

文献中将这一计算任务称为多变量多点求值（*MME*）。由于该问题本身显而易见的自然吸引力，以及 MME 作为其他许多代数问题算法的重要子程序所具有的潜在应用，计算代数领域对 MME 的快速算法有着根本性的兴趣（有关这些应用的详细讨论，见[KU11]）。

MME 的输入由 $(\mathrm{d}^m + \mathrm{N})\mathrm{m}$ 个域元素指定；或者说，由 $(\mathrm{d}^m + \mathrm{N})\mathrm{m} \cdot \mathrm{s}$ 位指定，其中 s 是输入中任意域常数的位复杂度上界。对于有限域，s 可取为 $\log |\mathbb{F}|$。显然，该问题存在一种算法，需要大约 $(\mathrm{d}^m \cdot \mathrm{N})^{1+o(1)}$ 次域运算，或约 $(\mathrm{d}^m \cdot \mathrm{N} \cdot \mathrm{s})^{1+o(1)}$ 次位运算：我们逐个输入点迭代求值该多项式。如何获得显著更快、理想情况下近线性的[1]MME 时间算法，是本研究的核心问题。这里，算法的时间复杂度可以按域运算次数衡量（若算法是“代数的[2]”，即只使用底层域上的域运算，例如上述朴素算法），也可以按位运算次数衡量。

1.1 先前工作

在介绍本文研究的具体问题和主要结果之前，我们先简要讨论当前 MME 算法的研究现状。尽管本文的结果针对实数、有理数和复数等无限域，我们还是先回顾有限域上的相关工作，以此开始讨论 MME 的先前研究。

1.1.1 有限域上的多点求值

即使对于单变量多项式，多项式的多点求值也并非易事；即便仅考虑这种情况，在任意（规模足够大）的域上，如何设计有效算法也并不明确。当输入点集具有额外结构时，例如它们都是底层域中某个阶的单位根，快速傅里叶变换（FFT）便能为这一问题提供近线性时间算法。然而，基于 FFT 的思路能否轻易推广到任意求值点的情形，并不显而易见。

[1]如果对于所有足够大的 n，算法在规模为 n 的输入上运行所需时间为 $n^{1+o(1)}$，那么我们称该算法的时间复杂度关于输入规模是近线性的。[2]执行过程中只需在底层有限域上进行算术运算的 MME 算法，换言之，可建模为底层有限域上的算术电路的算法，称为代数算法。

1974年，Borodin 和 Moenck 的一项出色研究 [BM74] 通过结合 FFT 和一种用于单变量多项式的快速带余除法算法，为单变量多点求值设计了一种快得多的算法。Borodin 和 Moenck 的算法适用于所有域，并且是代数算法，正如前文所述；该算法所需的域运算次数为 $(N+d)^{1+o(1)}$，相对于输入中的域元素数量而言是近线性的。

将这些多点求值快速算法从单变量情形推广到多变量情形，即使是双变量情形，也被证明颇具挑战性。Nüsken 和 Ziegler [NZ04]为这一问题给出了一个具有实质性加速的算法，适用于所有域，尽管其算法的精确时间复杂度并未达到输入规模的近线性。Umans[Uma08] 和 Kedlaya & Umans [KU08]（另见[KU11]）的研究显著推进了这一问题的进展：他们针对变量个数m 远小于次数参数d的情形，即$m = d^{o(1)}$，分别在小特征的域和所有有限域上，为 MME 给出了快速算法。

近期，Bhargava、Ghosh、Kumar 和 Mohapatra [BGKM22] 以及 Bhargava、Ghosh、Guo、Kumar 和 Umans [BGGKU22] 针对变量个数较大的情形，分别在小特征的域和所有有限域上，为 MME 给出了快速[3] 算法，适用于所有充分大的m，d。

我们还注意到，Kedlaya & Umans [KU08] 以及 Bhargava、Ghosh、Guo、Kumar 和 Umans [BGGKU22] 针对所有有限域上 MME 的算法并非代数算法，且尤其依赖于对输入域元素的位访问以及对它们进行的位运算。这与 Umans [Uma08] 以及 Bhargava、Ghosh、Kumar 和 Mohapatra [BGKM22] 针对小特征有限域上 MME 的算法形成对比，后者本质上是代数算法。为所有有限域上的 MME 设计代数算法，仍是这一研究方向中一个非常有趣的开放问题。‘

1.1.2 无限域上的多点求值

正如我们将看到的，与有限域上的情形相比，我们对这里这一问题的理解相当有限。不过，在转向结果之前，我们先讨论一下该问题在无限域上的定义本身所涉及的一些细微之处。

域运算与位复杂度： 有限域上的域算术能够保持所生成常数的最坏情况位复杂度，但在无限域上并非如此。中间计算中的位复杂度增加会引发一些问题，我们将在下文讨论。

第一个问题是，即使输出的位复杂度相对于输入位复杂度也可能不是近线性的，从而排除了设计出时间复杂度相对于输入位复杂度为近线性的算法的任何希望。第二个问题是，即使对于输入域元素和输出域元素的位复杂度都被保证较小的输入，

[3]严格来说，这些算法的运行时间并非近线性，因为运行时间包含$(\log|\mathbb{F}|)^c$ 因子，其中c 是一个固定常数，且可能大于 1。不过，运行时间对$(d^m + Nm)$ 这一项的依赖是近线性的。

算法在运行的某个中间阶段也可能生成位复杂度显著较大的域元素。例如，Borodin & Moenck 关于单变量多点求值的经典算法，就域运算次数而言具有近线性的复杂度，但尚不清楚该算法的位复杂度相对于输入和输出的位复杂度是否也为近线性。

输入与输出模型：对于实数域或复数域等域，我们需要指定一种输入提供模型，因为输入可能需要无限精度。数值算法中使用的标准模型是通过黑盒实现的，我们称之为近似预言机（其形式化定义见定义 2.7）。直观地说，实数 $\alpha \in (-1,1)$的近似预言机会对每个$k \in \mathbb{N}$，在时间$\tilde{O}(k)$ 内提供 α 小数点后前k 位以及其符号（对于复数，我们假设其实部和虚部由此类预言机提供）。

对于输出，我们可以要求将求值结果计算到所需精度，也可以在例如有理数的情况下精确计算求值结果。本文讨论这两种问题。

注意，在给定误差 $\varepsilon < 1$ 的情况下计算实数 $\alpha \in (-1,1)$，本质上等同于正确计算输出中最重要的 $\Omega(\log 1/\varepsilon)$ 位。从这个意义上说，$O(\log 1/\varepsilon)$ 为近似多点求值实例的输出位复杂度提供了一个自然的上界。或许有些令人意外的是，直到最近，我们才知道一种多点求值算法，能够在单变量多项式的情形下实现关于输入规模和$(\log 1/\varepsilon)$ 的近线性位复杂度。与之形成对比的是，Borodin & Moenck [BM74] 的结果为所有域上的（精确）单变量多点求值给出了域运算次数的上界（但未给出位运算次数的上界）。

在一项近期的杰出工作中，Moroz [Mor21] 为单变量多点求值的近似版本设计了这样一种算法。具体而言，他证明了以下定理。

定理 1.1 (Moroz [Mor21])。存在一种确定性算法，其输入为一个以复系数列表形式给出的单变量多项式 $f(x)=\sum_{i=0}^{d} f_i x^i \in \mathbb{C}[x]$，其中$(|f|_1 := \sum_{i=0}^{d} |f_i| \le 2^{\tau})$，以及输入 a_1, a_2、……、$a_d \in \mathbb{C}$，这些输入的绝对值小于一；算法输出 β_1，β_2、……、$\beta_d \in \mathbb{C}$，使得对每个 i，

$$|f(a_i) - \beta_i| \le |f|_1 \cdot 2^{-t},$$

且其位复杂度至多为 $\tilde{O}(d(\tau + t))$。

作为本文的主要结果，我们证明了定理 1.1 在多元情形下的推广。

1.2 我们的结果

在陈述结果之前，我们先正式定义所研究的问题。近似 MME 的第一个问题，本质上是 Moroz 所研究问题[Mor21]的一元版本的推广。为方便起见，我们针对有理数域和实数域陈述该问题，但它们可以直接推广到复数域及其其他自然子域。

问题 1.2（近似多元多点求值（近似 MME））。给定输入为一个 m元多项式 $f \in \mathbb{R}[\mathbf{x}]$，其每个变量的次数至多为（$d-1$），并以系数列表的形式给出；还有点$\mathbf{a}_1,\ldots,\mathbf{a}_N \in \mathbb{R}^m$以及精度参数 $t \in \mathbb{N}$。这里假设每个域元素都属于（$-1,1$），并通过近似预言机给出。

计算有理数 $\beta_1,\ldots,\beta_N$，使得对于所有 $i \in [N]$，都有 $|f(\mathbf{a}_i)-\beta_i| < 1/2^t$。

本文还研究了 MME 的以下变体。

问题 1.3（精确多元多点求值（精确 MME））。给定输入为一个 m元多项式 $f \in \mathbb{Q}[\mathbf{x}]$，其每个变量的次数至多为（$d-1$），并以系数列表的形式给出；还有点$\mathbf{a}_1,\ldots,\mathbf{a}_N \in \mathbb{Q}^m$ 以及整数参数 $s > 0$，使得输入和输出中的所有有理数都可表示为 p/q，其中 p,q 为整数，且 $|p|,|q| < 2^s$；并且输入中的每个有理数的绝对值都小于一。

计算 $f(\mathbf{a}_1)$，...，$f(\mathbf{a}_N)$。

所有常数的绝对值均不超过 1 这一限制需要简要讨论。对f 的系数所作的限制不失一般性（通过缩放即可满足），但对点的坐标所作的限制并非不失一般性，不过仍有充分理由采用这一限制。详情请参阅备注 5.1 。

我们的主要结果是针对问题 1.2 和问题 1.3 的快速算法，适用于所有充分大的d。

定理 1.4（近似多点求值——非正式表述）。 存在一种用于近似 *MME*（问题 *1.2*）的确定性算法，其运行时间为

$$((Nm+d^m)t)^{1+o(1)},$$

对所有充分大的 d，t 以及所有 m。

定理 1.5（精确多点求值——非正式表述）。 存在一种用于在有理数上进行精确 *MME*（问题 *1.3*）的确定性算法，其运行时间为

$$((Nm+d^m)s)^{1+o(1)}$$

对于所有充分大的 d，s 以及所有 m。

定理 1.4 通过缩放系数，对 Moroz 的定理 1.1 作了推广，因为它能够处理任意多的变量。有趣的是，我们的证明并非将 定理 1.1 的证明扩展到更多变量。它依赖于一套不同的思想，在概念上似乎也不同于 Moroz 的证明[Mor21]。Moroz 算法依赖几何思想，不涉及任何模算术；而我们的算法则关键依赖于各种归约：将有理数、实数或复数上的 MME 实例（近似或精确）归约为有限域上的 MME 实例。事实上，我们尚不清楚如何将 Moroz 的单变量算法推广到高维，而这一问题值得探究。

如引言中所讨论的，在以位运算次数衡量有理数域上 MME 算法的复杂度时，运行时间依赖于输出的位复杂度（如定理 1.5 所示）是十分自然且基本上无法避免的。然而，定理 1.5 将输出的位复杂度作为输入的一部分，这一点似乎并不十分自然或理想。若能为有理数上的精确 MME 设计一种算法，使其无需将输出复杂度的上界作为输入的一部分，且运行时间关于输入和输出位复杂度近线性，将会非常有意义。

1.3 证明概述

本节概述定理 1.4 和定理 1.5的证明思路。为便于讨论，我们假设输入属于有理数域，且输入中的域常数均已精确给出。借助近似预言机的一些简洁性质，这里的思路可以推广到实数输入的情形（用于近似 MME）。

1.3.1 一种朴素的初步尝试

我们先来设定一些符号。设$f \in \mathbb{Q}[\mathbf{x}]$ 是一个m 元多项式，其关于每个变量的次数至多为$(d-1)$，并设$\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_N \in \mathbb{Q}^m$ 为我们感兴趣的输入点。暂且假设我们的目标是精确输出f 在每个$\mathbf{a}_i$ 处的值。此外，还给定正整数t，使得输入中的每个域常数以及输出的分子和分母至多为 2^t。

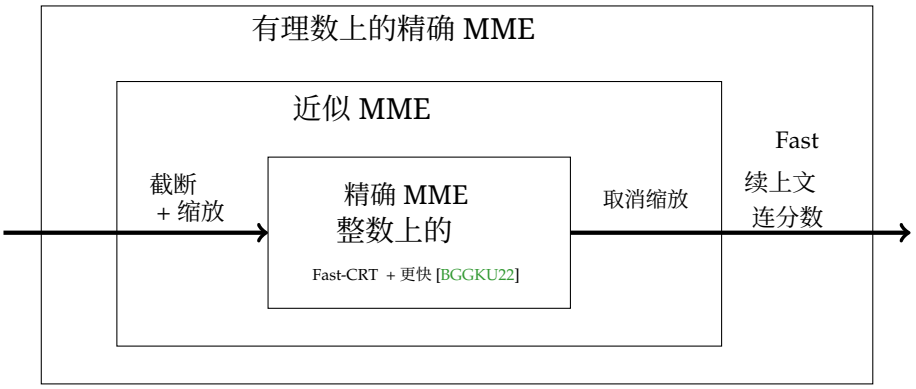
根据 Bhargava、Ghosh、Guo、Kumar 和 Umans 先前的工作[BGGKU22]，我们已经拥有适用于所有有限域上 MME 的快速算法。因此，解决有理数域上 MME 的一种自然策略，是设法将有理数域上的问题归约为有限域上的若干实例，并利用已知的有限域上该问题的算法来解决这些实例。实现这种归约的第一步，是对输入实例中的所有分母取最小公倍数以消去分母，从而得到一个整数环上的 MME 实例；随后对一个足够大的素数（或者出于效率考虑，在需要时对若干个足够小的素数）取模，由此将问题归约为有限域上的若干 MME 实例。然而，即使在每个求值点$\mathbf{a}_i$

的坐标都是整数，且有理数仅出现在多项式f的系数中的简单情形下，这种看似自然的方法也会遇到根本性问题。下面我们简要阐述这一问题。

考虑这样一个输入实例：f 的系数向量中，每个分母都是不同的素数。例如，可以构造这样一个实例，使前d^m 个素数都作为f某个系数的分母。注意，在这种情况下，输入位复杂度参数t 至多需要为 $\mathrm{poly}(\log d, m)$。由于该系数向量的长度为d^m，这些分母的最小公倍数是一个自然数，其值至少等于前d^m个素数的乘积，而该乘积至少为 2^{d^m}，因此其位复杂度至少为d^m。因此，若清除f 系数的ˆ分母，得到一个具有整数系数的多项式f ，则f 的每个ˆ系数的位复杂度都可能高达d^m。在这种情况下，ˆf 的系数向量的总位复杂度至少为d^{2m}，约为原始输入规模的平方。因此，任何通过这种方法得到的算法，其时间复杂度都会大到难以承受。

在我们针对近似 MME 和精确 MME 的两种算法中，确实都关键地依赖于 Bhargava 等人提出的有限域上的 MME 算法[BGGKU22]。然而，这种归约颇为微妙，需要谨慎处理。在此过程中，我们依赖计算代数和数论中的一些知名工具，例如快速中国剩余定理、快速有理重构、克罗内克映射和逆克罗内克映射。或许有些出人意料的是，我们的整数上的精确 MME 算法将近似 MME 算法用作子程序。

图 1：归约概览



现在我们概述这些算法的主要思路。首先介绍一种非常简单的算法，用于解决整数输入这一特殊情况下的精确 MME 问题；它也是近似 MME 算法的关键子程序。

1.3.2 整数上的精确 MME 算法

对于这个问题，输入中的所有域元素都是整数，并且给定参数 s 时，这些输入域元素及输出中的域元素的绝对值均不超过 2^s。

该算法　　对于这种情况，MME 的算法只需通过对一个足够大的素　　e

数取模来完成这项计算（素数的大小根据给定的输入和输出复杂度确定），从而将整数上的问题归约为足够大的有限域上的问题。此时，我们基本上可以调用 Bhargava 等人的有限域上 MME 算法来解决该问题。这里有一点需要注意：正如他们的论文 [BGGKU22], 所述，该算法由于两个因素，运行时间并非严格近线性。首先，运行时间中包含一个 $\mathrm{poly}(\log|\mathbb{F}|)$项，其中 $\mathrm{poly}()$ 项的次数可能严格大于 1。另一个问题是，即使只考虑对 d,m的依赖，他们的算法也只有在 m 不断增大时才具有近线性运行时间。因此，当 m为常数时，我们无法直接将 [BGGKU22] 中的算法用于我们的应用。

我们用一些简单的思路解决这两个问题。为了解决变量个数固定的问题，我们先对多项式应用逆 Kronecker 映射，人为增加变量个数，同时降低个体次数上界。接着，为了处理运行时间对域大小的依赖问题，我们先将问题提升到整数，再使用中国剩余定理，将其化为许多个在较小有限域上求解 MME 的实例。这与 Kedlaya & Umans 在 [KU11] 中使用的归约基本相同。为了保持近线性运行时间，我们使用众所周知的近线性时间算法来进行中国剩余定理计算。详情见 第 3 节 和 第 4 节。

1.3.3 近似 MME 算法

回顾一下，对于近似 MME，我们不需要精确计算多项式在输入点处的值，只要求输出与实际求值结果之间的误差不超过某个范围。为简单起见，假设输入多项式和求值点都是有理数，且其值均已精确给出。正如本节前面提到的，直接清除分母（通过最小公倍数）并归约到整数情形似乎很困难，因为在某些实例中，例如所有分母都是互不相同的素数时，这一过程会使系数的规模过度膨胀。然而，利用近似值进行计算，可以为我们提供必要的灵活空间，使这一思路能够近似实现。

算法的第一步是将所有域常数、给定多项式的系数以及输入点的坐标的小数表示截断到小数点后 k 位（其中 k 需足够大，并经过仔细选择）。将绝对值不超过 1 的实数 α 按此方式舍入，可得到一个形如 $\alpha_a/2^k$ 的有理数，其中整数 a 满足 $|a| \leq 2^k$。此外，我们有 $|\alpha - \hat{\alpha}| < 1/2^k$。现在，我们求解舍入后得到的这个实例上的 MME。此时的关键优势在于，舍入后实例中的所有分母都是 2^k，因此它们的最小公倍数就是 2^k，清除分母不再会导致位复杂度大幅增加。现在，我们调用前一小节中介绍的、用于整数实例的精确 MME 算法。详情见 第 5 节。

1.3.4 有理数上的精确 MME 算法

对于我们的精确 MME 算法，首先要在该实例上调用近似 MME 算法，并选取足够精确的精度参数 t。t 的取值取决于作为输入提供给我们的输出位复杂度。根据近似 MME 算法的输出保证，我们知道，近似 MME 输出的有理数与真实求值的差至多为 $1/2^t$。如果我们能以某种方式从这些近似值中恢复出真实求值，就大功告成了！这里的问题可表述如下：我们的目标是找出一个隐藏的有理数，记为 a/b（真实求值）；我们可以得到另一个有理数，记为 A/B（真实求值的近似值），并且保证 $|A/B - a/b| < 1/2^t$，且 $|A|、|B| < 2^{O(t)}$。关键是，我们还有一个参数 s（作为输入提供给我们），并且保证 $|a|、|b| < 2^s$。

这本质上是有理数重构的一个实例。这是计算代数和数论中一个经过充分研究、具有经典意义的问题。我们依赖这些结果（基本上以黑盒方式），尤其利用连分数的概念和性质，高效地解决这个问题。我们注意到，参数 t（作为 s 的函数）取这样的值，意味着 a/b 是一个渐近分数（通过截断 A/B 的连分数表示得到的有理数）。结合这一观察和渐近分数的一些性质，我们能够在给定 A/B的情况下，以近线性时间求出 a/b。详细内容见 第6节。

2 预备知识

符号

- 我们用粗体字母**a**、**b** 等表示有限维向量。我们也用这种方式表示变量元组$\mathbf{x} = (x1, \ldots, x_n)$等。通常，向量的维数可根据上下文明确得知。

- 对于指数向量 $\mathbf{e} = (e1, \ldots, e_n) \in \mathbb{Z}_{n\geq 0}$ 以及向量$\mathbf{x} = (x1, \ldots, x_n)$，我们用$\mathbf{x_e}$表示单项式$x_{e_1^1} \cdots x_{e_n n}$。

- 对于一个实数 α，我们用 $\lfloor \alpha \rceil$ 表示最接近 α的整数。当 $\alpha = a + \frac{1}{2}$ 对于某个整数a时，$\lfloor \alpha \rceil$ 定义为a。

2.1 有用的不等式

引理2.1（关于二项级数的界）。 对于 $d \in \mathbb{N}$ 和 $\varepsilon > 0$ 满足 $|\varepsilon| < 1/d^2$，有

$$1 + d\varepsilon \leq (1+\varepsilon)^d \leq 1 + d\varepsilon + d^2\varepsilon^2.$$

证明。 当$d = 1,2$ 时，不等式显然成立，因此在接下来的讨论中，我们不失一般性地假设$d \geq 3$。

对于任意$i \geq 3$，都有$\binom{d}{i} \leq \binom{d}{2} \cdot d^{i-2}$。因此，

$$\left|\sum_{i=3}^{d}\binom{d}{i}\varepsilon^i\right| \leq \sum_{i=3}^{d}\binom{d}{i}\left|\varepsilon^i\right| \leq \binom{d}{2}\cdot\varepsilon^2\cdot\sum_{i=1}^{d-2}|d\varepsilon|^i < \binom{d}{2}\cdot\varepsilon^2$$

其中，最后一个不等式用到了 $\varepsilon < 1/d^2$。因此，

$$(1+\varepsilon)^d = 1 + d\varepsilon + \binom{d}{2}\varepsilon^2 + \sum_{i=3}^{d}\binom{d}{i}\varepsilon^i \geq 1 + d\varepsilon$$

and

$$\begin{aligned}(1+\varepsilon)^d &= 1 + d\varepsilon + \binom{d}{2}\varepsilon^2 + \sum_{i=3}^{d}\binom{d}{i}\varepsilon^i \\ &\leq 1 + d\varepsilon + \binom{d}{2}\varepsilon^2 + \binom{d}{2}\varepsilon^2 \\ &\leq 1 + d\varepsilon + d^2\varepsilon^2.\end{aligned}$$

□

2.2 底数 d 的克罗内克映射

克罗内克映射是一种常用工具，可在不改变底层稀疏性的情况下进行变量约简。该映射的形式化定义如下。

定义 2.2 (底数 d 的克罗内克映射**)**。底数 d 的 c元克罗内克映射，记作$\Phi_{d,\ m;c}$ 将 cm 元多项式映射为 c 元多项式，映射方式如下：

$$\Phi_{d,m;c}(f(x_{1,1},\ldots,x_{1,m},\ldots,x_{c,1},\ldots,x_{c,m})) = f\left(1, y_1^d, y_1^{d^2},\ldots,y^{d^{m-1}},\ldots,1,y_c^d,y_c^{d^2},\ldots,y_c^{d^{m-1}}\right).$$

若 f 是一个单变量次数小于 d 的多项式，则单项式 $\mathbf{x}_1^{\mathbf{e}_1}\cdots\mathbf{x}^{\mathbf{e}_c}c$ 会映射为单项式 $y_1^{e_1}\cdots y^{e_c}c$，其中 $\mathbf{e}_i$ 是 e_i 的底数 d 表示。

同理，我们定义逆克罗内克映射，记作 $\Phi^{-1}\ d,\ m;\ c$，它将单变量次数小于 d^m 的 c 元多项式映射为单变量次数小于 d 的 cm 元多项式，其定义为将以下映射通过对单项式线性扩展而得：

$$\Phi_{d,m;c}^{-1}(y_1^{e_1}\cdots y_c^{e_c}) = \mathbf{x}_1^{\mathbf{e}_1}\cdots\mathbf{x}_m^{\mathbf{e}_m}$$

其中 $\mathbf{x}_i = (x_{i,1},\ldots,x_{i,m})$，并且 $\mathbf{e}_i \in \{0,\ldots,d-1\}^m$ 是 $e_i < d^m$ 的底数 d 表示。

与逆克罗内克映射相关联，我们还定义了 $\psi_{d,\ m;c}$：$\mathbb{F}^c \to \mathbb{F}^{cm}$ 它作用于点，定义如下

$$\psi_{d,m;c} : (a_1,\ldots,a_c) \mapsto (1, a_1^d,\ldots,a_1^{d^{m-1}},\ldots,1,a_c^d,\ldots,a_c^{d^{m-1}}).$$

◊

T逆克罗内克映射的定义使我们得到以下观察 .

观察 2.3 (克罗内克映射与求值结果)。 若 $f(x_1, \ldots, x_c)$ 是一个每个变量的次数都小于 d^m 的多项式，那么对于任意 $\mathbf{a} \in \mathbb{F}^c$，都有 $\Phi^{-1} d,\ m;\ c(f)(\psi_{d,\ m;c}(\mathbf{a})) = f(\mathbf{a})$。 □

上述观察有助于根据证明需要，在次数与变量个数之间进行权衡。

2.3 计算小于给定数的所有素数

经典素数定理 [Had96, LVP97] 断言，渐近而言，小于 N 的素数有 $\Theta(N/\log N)$ 个。我们可以在确定性 $\tilde{O}(N)$ 时间内计算出所有小于 N 的素数。

算法 1： `PrimeSieve`

输入： 一个整数 $N > 1$

输出： 所有小于N的素数。

```
1 初始化一个以 2、3、.....、N 为索引的数组S，并将所有值设为真。
2 i ← 2到 √N 循环
3     如果S[i] 为TRUE，则
4         设置 j ← 2i
5         当j ≤ N 时循环
6             将S[j] 设为假。
7             j ← j + i。
8 返回 {i : S[i] 为 真}。
```

引理 2.4 (计算小于给定数的素数)。存在一种确定性算法（算法 *1*），可在确定性时间 $\tilde{O}(N)$内计算出所有小于 N 的素数。 □

2.4 快速中国剩余

我们还将用到以下两个定理，它们涉及与中国剩余定理（CRT）相关问题的快速算法。有关证明，请读者参阅 von zur Gathen 和 Gerhard 所著的书 [GG13]。

引理 2.5 (Fast-CRT：模数计算)。 存在一种算法，当输入互素正整数 p_1,.....，p_r 以及满足 $N < \prod p_i < 2^c$ 的正整数 N 时，可在确定性的 $\tilde{O}(c)$ 时间内，计算出对于 $i = 1$，.....，r 的余数 $a_i = N \bmod p_i$。

上述引理的证明见 [GG13，定理 10。24]。

引理 2.6（Fast-CRT：重构）。 存在一种算法，输入互素正整数 $p_1,\ldots,p_r$ 和 $a_1,\ldots,a_r$，使得 $0 \le a_i < p_i$，输出唯一整数 $0 \le N < \prod p_i$，满足 $N = a_i$ 模 p_i，其中 $i = 1,\ldots,r$，并在确定性 $\tilde{O}(c)$ 时间内运行，其中 $\prod p_i < 2^c$。

上述引理的证明见[GG13，定理 10。25]

2.5 任意精度实数的输入模型

在本节中，我们假设所有作为"输入"的实数（即多项式的系数和求值点的坐标）都在$(-1,1)$范围内，并通过近似预言机提供，且满足以下保证：

定义 2.7 (近似预言机)。 $\alpha \in$ 近似预言机对于 $\alpha \in (-1,1)$，可在 $O(1)$ 时间内给出"符号" α，并在输入 k 时返回一个满足下式的整数 $b_k \in [-2^k, 2^k]$

$$|\alpha - b_k/2^k| < 1/2^k.$$

我们用 $\lfloor \alpha \rceil_k$ 表示由近似预言机得到的分数 $b_k/2^k$ $e.$

近似预言机的运行时间是输出 b_k所需的时间。如果运行时间为 $\tilde{O}(k)$，则称该近似预言机高效。$\diamondsuit$

任何"自然"实数都可以通过任意一个收敛得足够快的级数，构造出这种高效近似预言机。对于形如 $\sqrt{2}$ 等的代数实数，标准泰勒级数就足够了。即使对于"自然"的超越数，我们也可能有这样的近似预言机：

$$\begin{aligned} e &= 1 + \frac{1}{1!} + \frac{1}{2!} + \cdots, \\ \pi &= 4 \cdot \tan^{-1}(1) \\ &= 4 \cdot \left(\tan^{-1}(1/2) + \tan^{-1}(1/3) \right) \\ &= 4 \cdot \left(1/2 - \frac{1/2^3}{3} + \frac{1/2^5}{5} - \cdots \quad + \quad 1/3 - \frac{1/3^3}{3} + \frac{1/3^5}{5} - \cdots \right). \end{aligned}$$

任何显式级数，只要其中 $\tilde{O}(k)$ 项的误差小于 $1/2^k$，就符合要求，可作为实数上的近似 MME 算法所需的高效近似预言机。

引理 2.8 (用于近似预言机的重复乘方)。 给定一个实数 $\alpha \in (-1,1)$ 的近似预言机 A，其运行时间为 $T(k)$，以及任意正整数 D，我们可以构造一个近似预言机 A^D，用于 α^D，其运行时间为 $T(k + O(\log D)) + \tilde{O}(k \log D)$。

证明。在 输入 k 时，我们希望找到一个整数 $r_k \in [-2^k, 2^k]$，使得 $|\alpha^D - r_k/2^k| < 1/2^k$.

我们先考虑D 为偶数的情况。设$t = k + 3$，并假设我们递归地计算出一个整数$a_t \in [-2^t, 2^t]$，使得$|\alpha^{D/2} - a_t/2^t| < 1/2^t$。设 $\delta = a_t/2^t - \alpha^{D/2}$。

$$\left| \alpha^D - a_t^2/2^{2t} \right| = \left| (\alpha^{D/2})^2 - (\alpha^{D/2} + \delta)^2 \right| < 4 \cdot 1/2^t \le 1/2^{k+1}$$

因此，如果$R_k = r_k \cdot 2^{2t-k}$是2^{2t-k}的最接近a_t^2的倍数，那么

$$\begin{aligned}\left|\alpha^D - r_k/2^k\right| &\le \left|\alpha^D - a_t^2/2^{2t}\right| + \left|a_t^2/2^{2t} - r_k 2^{2t-k}/2^{2t}\right| \\ &< 1/2^{k+1} + 2^{2t-k-1}/2^{2t} \le 1/2^k.\end{aligned}$$

若 D 为奇数，则令 $t = k + 4$。我们使用近似预言机 A 得到一个整数 $b_t \in [-2^t, 2^t]$，使得 $|\alpha - b_t/2^t| < 1/2^t$，并递归地计算一个整数 $a_t \in [-2^t, 2^t]$，使得 $\left|\alpha^{(D-1)/2} - a_t/2^t\right| < 1/2^t$。然后，

$$\begin{aligned}\left|\alpha^D - a_t^2 b_t/2^{3t}\right| &\le |\alpha| \left|\left(\alpha^{(D-1)/2}\right)^2 - a_t^2/2^{2t}\right| + \left|a_t^2/2^{2t}\right| |\alpha - b_t/2^t| \\ &< 4 \cdot 1/2^t + 1/2^t \le 1/2^{k+1}.\end{aligned}$$

类似地，若 $R_t = r_t \cdot 2^{3t-k}$ 是 2^{3t-k} 中最接近 $a_t^2 b_t$的倍数，则

$$\left|\alpha^D - r_t/2^k\right| < 1/2^k.$$

若 $\mathcal{T}(k, D)$ 是该算法（即 算法 2）计算 $r_k \in [-2^k, 2^k]$使得 $\left|\alpha^D - r_k/2^k\right| \le 1/2^k$所需的运行时间，则有

$$\begin{aligned}\mathcal{T}(k, D) &\le \mathcal{T}(k+4, D/2) + \tilde{O}(k) \\ &\le \mathcal{T}(k + O(\log D), 1) + \tilde{O}(k \log D) \\ &= T(k + O(\log D)) + \tilde{O}(k \log D).\end{aligned}$$

□

算法 2: ApproximationOracle-Powering

输入：实数 α 的近似预言机 A、整数 D > 0 和整数 k > 0。输出：整数 $r_k \in [-2^k, 2^k]$，使得 $\left|\alpha^D - r_k/2^k\right| < 1/2^k$。1 如果 D $= 1$，则 2 返回 r_k = A(k)。3 如果 D 是偶数，则 4 令 t $=$ k $+ 3$。5 计算 a_t = ApproximationOracle-Powering(A, D/2,t)。6 返回$\lfloor a_t^2/2^{2t-k} \rceil$。7 否则 8 令 t $=$ k $+ 4$。9 计算 b_t = A(t)。计算 a_t = ApproximationOracle-Powering(A, (D $-$ 1)/2, t)。10 返回$\lfloor a_t^2 \cdot b_t/2^{3t-k} \rceil$。

我们还注意到，近似预言机这一概念可自然扩展到以下对象的表示：

复数。这里，每个复数由两个这样的预言机给出，分别对应其实部和虚部。

3 重新审视素域上的 MME

我们回顾 Bhargava、Ghosh、Guo、Kumar 和 Umans 的结果[BGGKU22]。

定理 3.1（有限域上的快速多变量多点求值 [BGGKU22]）。 存在一种确定性算法，其输入为某个有限域 $\mathbb{F}$ 上一个 m 元多项式 f 的系数向量，该多项式在每个变量上的次数都小于 d，以及 N 个点 $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_N \in \mathbb{F}^m$ ；算法在时间内输出 $f(\mathbf{a}_1), f(\mathbf{a}_2), \ldots, f(\mathbf{a}_N)$

$$(d^m + Nm)^{1+o(1)} \cdot \mathrm{poly}(m, d, \log|\mathbb{F}|),$$

对所有 $m \in \mathbb{N}$ 以及充分大的 $d \in \mathbb{N}$,

由于 $\mathrm{poly}(\log|\mathbb{F}|)$ 这一因子，上述运行时间按位数衡量输入时并非完全近线性。此外，在 m 为常数的情况下，我们无法再将 $\mathrm{poly}(d)$ 吸收到 $(Nm + d^m)^{o(1)}$ 中。不过，我们将在下文证明，对于素域的情形，可以克服这些问题，并得到如下近线性时间界。

定理 3.2（素域上的近线性时间多点求值）。 存在一种确定性算法（即算法 3），当输入为素域 $\mathbb{F}_p$ 上一个每个变量次数均小于 d 的 m元多项式 f 的系数向量，以及 N 个点 $\mathbf{a}^{(1)}$、……、$\mathbf{a}^{(N)} \in \mathbb{F}^m$ 时，输出 $f(\mathbf{a}^{(1)})$ 、 …… 、 $f(\mathbf{a}^{(N)})$ ，所需时间为

$$((d^m + Nm) \cdot \log p)^{1+o(1)}$$

对所有 $m \in \mathbb{N}$ 以及充分大的 $d \in \mathbb{N}$均成立。

我们首先在以下两个小节中分别讨论变量个数为常数和随输入增长这两种情况下的处理方法，然后证明定理 3.2。

我们首先在以下两个小节中分别讨论变量个数为常数和随输入增长这两种情况下的处理方法，然后证明定理 3.2。

3.1 处理变量个数过少的情形

如上所述，当变量个数过少时（例如 $m \leq c$ ，其中 c 为常数），我们可能不再有 $\mathrm{poly}(d) = d^{o(m)}$。不过，我们可以利用逆克罗内克映射（定义 2.2），在次数和变量个数之间进行权衡。

为了使参数更具信息量，我们重新命名这些参数，并设f为一个c元多项式，其单变量次数小于D；设$\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(N)} \in \mathbb{F}_p^c$为我们希望对该多项式求值的点。

令 $d = \lfloor \log D \rfloor$，并令 m 为满足 $d^m > D$ 的最小整数。注意，$d^m > D > d^{m-1}$，且 $m = \Theta(\log D / \log\log D)$。若 $f(x_1, \ldots, x_c) = \sum_{\mathbf{e}} f_{\mathbf{e}} \cdot x^{\mathbf{e}}$，则定义多项式 $g(y_{1,1}, \ldots, y_{c,m}) = \Phi^{-1}(d,m;c(f))$，如定义 2.2 所述。

对于所有 $i \in [N]$，按照定义 2.2 中的定义，令 $\mathbf{a}^{(i)} = \psi_{d,m;c}(\mathbf{a}^{(i)})$。于是，根据观察 2.3，对于所有 $i \in [N]$，有 $f(\mathbf{a}^{(i)}) = g(\mathbf{a}^{(i)})$。下面的观察表明，可以从 $\mathbf{a}^{(i)}$ 高效地计算 $\mathbf{a}^{(i)}$。

观察 3.3。 给定 $\mathbf{a} \in \mathbb{F}_p^c$，点 $\tilde{\mathbf{a}} := \psi^{(c)}(d,m)(\mathbf{a}) \in \mathbb{F}_p^{cm}$ 可在多项式 (d, m, c) 时间内计算。

证明。运行时间上界可由重复乘方得到：a^{d^k} 模 $p = (a^{d^{k-1}}$ 模 $p)^d$ 模 p；此外，模 p 加法和乘法可在 $\tilde{O}(\log p)$ 时间内完成。

□

因此，计算 $f(a^{(1)}), \ldots, f(a^{(N)})$ 的任务可归约为计算 $g(\tilde{a}^{(1)}), \ldots, g(\tilde{a}^{(N)})$ 的取值，其中 $\tilde{a}^{(i)} = \psi^{(c)}(d,m)(a)$。此外，由于 $d, m = D^{o(1)}$，该归约的运行时间为 $((D^c + Nc) \cdot \log p)^{1+o(1)}$。

3.2 当单变量次数和变量个数适度增长时

我们回到熟悉的变量记号约定：$f(x_1, \ldots, x_m) \in \mathbb{F}_p[x_1, \ldots, x_m]$，且每个变量的次数都小于 d。根据上一节，可以不失一般性地假设 $d, m = \omega(1)$，因此 $\mathrm{poly}(d, m) = (d^m + Nm)^{o(1)}$。设 f 可写成如下单项式之和。

$$f(x_1, \ldots, x_m) = \sum_{\mathbf{e}} f_{\mathbf{e}} \cdot x_1^{e_1} \cdots x_m^{e_m}.$$

将上述内容理解为一个整数上的多项式，其每个系数均在 $\{0, 1, \ldots, p-1\}$ 中；对于任意 $\mathbf{a} \in \{0, \ldots, p-1\}^m$，整数 $f(\mathbf{a})$ 的界为 $d^m \cdot p \cdot p^{dm}$。其思路是利用中国剩余定理，将问题化简为较小素域上的 MME。

算法 3： NearlyLinearTimeMME-PrimeFields

输入： $f(x_1,\ldots,x_m)\in\mathbb{F}_p[x_1,\ldots,x_m]$ 每个变量的次数小于d，并且 $\mathbf{a}^{(1)},\ldots,\mathbf{a}^{(N)}\in\mathbb{F}_p^m$.

输出： 对于$i\in[N]$，求值结果为$b_i=f(\mathbf{a}^{(i)})$。

1 如果 $m<\log\log d$ **则**

2 令 $d'=\lfloor\log d\rfloor$ 和 m' 为满足 $(d')^m$的最小整数 $'>d$.

3 将f替换为 $\Phi^{-1}{}_{d',m';m}(f)$，并将每个$\mathbf{a}^{(i)}$替换为$\psi_{d',m';m}(\mathbf{a}^{(i)})$。

4 设 $\tilde{L}=(dm+1)\log p+m\log d$。计算前 $\tilde{L}$ 个素数 $\{p_1,\ldots\ldots,p_{\tilde{L}}\}$。

5 设$L\le\tilde{L}$是满足以下条件的最小整数：$p_1\cdots p_L=:M>d^m\cdot p\cdot p^{dm}$。

6 对于 $\mathbf{e}\in\{0,\ldots,d-1\}^m$，**令**

7 计算 $f_{\mathbf{e}}^{(\ell)}=f_{\mathbf{e}}$ 模 p_ℓ，针对 $\ell\in L$ 通过快速 CRT 模数计算（引理 2.5）。

8 对于 $i\in[N],k\in[m]$ **执行**

9 计算 $a_{i,k,\ell}=\mathbf{a}^{(i)}{}_k$ 模 p_ℓ，针对 $\ell\in L$ 通过快速 CRT 模数计算（引理 2.5）。

10 对 $\ell\in L$**执行**

11 令 $f^{(\ell)}(x_1,\ldots\ldots,x_m)=\sum_{\mathbf{e}}f_{(\ell)\mathbf{e}}\,\mathbf{x}^{\mathbf{e}}\in\mathbb{F}_{p_i}[\mathbf{x}]$.

12 令 $\mathbf{a}^{(i,\ell)}=(a_{i,1,\ell},\ldots,a_{i,m,\ell})\in\mathbb{F}_{p_\ell}^m$ 对每个 $i\in[N]$。

13 计算 $b_{i,\ell}=f^{(\ell)}(\mathbf{a}^{(i,\ell)})$ 对所有 $i\in[N]$ 使用 定理 3.1。

14 对$i\in[N]$ **执行**

15 计算唯一的 $b_i\in[0,M)$ 使得 $b_i=b_{i,\ell}$ 模 p_ℓ 对所有 $\ell\in[L]$, 通过快速中国剩余定理重构 (引理 2.6)。

16 返回 $(b_1$ 模 $p,\ldots,b_N$ 模 $p)$。

定理 *3.2* 的证明定理 *3.2*。 算法 3 的正确性算法 3 显而易见。

至于运行时间，根据第 1 行至第 3 行和观察 3.3，其耗时为$(d^m+Nm)^{1+o(1)}$，并可归约到$m\ge\log\log d$的情形。在这种情况下，第 4 行和第 5 行需要 $\tilde{O}(\tilde{L})$ 的时间（引理 2.4），即 $\tilde{O}(\log p)\cdot\mathrm{poly}(d,m)$。

利用引理 2.5，可知第 6 行至第 9 行需要时间$(d^m+Nm)\cdot\tilde{O}(\log M)=((d^m+Nm)\cdot\log p)^{1+o(1)}$。

根据定理 3.1，Line 13 的运行时间为 $(\mathrm{d}^m+\mathrm{Nm})^{1+o(1)}\cdot\mathrm{poly}(\mathrm{d},\mathrm{m},\log\mathrm{p}_i)$。又由于 $\mathrm{p}_i<\tilde{\mathrm{O}}(\tilde{\mathrm{L}})=\tilde{\mathrm{O}}(\mathrm{dm}\log\mathrm{p})$，因此，Lines 10 至 13 中的整个循环耗时为 $(\mathrm{d}^m+\mathrm{Nm})^{1+o(1)}\cdot\tilde{\mathrm{O}}(\log\mathrm{p})=((\mathrm{d}^m+\mathrm{Nm})\log\mathrm{p})^{1+o(1)}$。

最后，根据引理 2.6，第 14 行到第 15 行的整个循环所需时间为$(Nm)\cdot\tilde{O}(\log M)=((d^m+Nm)\cdot\log p)^{1+o(1)}$。因此，算法 3 的运行时间为$((d^m+Nm)\log p)^{1+o(1)}$。

□

4 整数上的精确 MME（已知输出位复杂度）

本节研究整数上的 MME 的以下版本。

输入： 一个整数 $s > 0$，一个多项式 $f(x_1, \ldots, x_m) \in \mathbb{Z}[x_1, \ldots, x_m]$，其单变量次数均小于 d，以 d^m 个整数系数的列表给出；一个点集 $\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(N)} \in \mathbb{Z}^m$，其中每个坐标的绝对值至多为 2^s；并保证 f 的所有系数、$\mathbf{a}^{(i)}$ 的坐标以及求值结果 $f(\mathbf{a}^{(i)})$ 的绝对值均不超过2^s。

输出： 整数 $b_1, \ldots, b_N$，它们是求值结果，即 $b_i = f(\mathbf{a}^{(i)})$，其中 $i \in [N]$。

定理 4.1 (整数上的精确 MME)。存在一种确定性算法（即 算法 4），对于上述输入，会返回上述所需输出，且其确定性运行时间为 $((d^m + Nm) \cdot s)^{1+o(1)}$，适用于所有 $m \in \mathbb{N}$ 以及充分大的 $d \in \mathbb{N}$。

主要思路是利用中国剩余定理，将问题化简为有限域上的 MME。由于我们希望得到近线性时间算法，因此需要再次使用以近线性时间实现的中国剩余定理（引理 2.5 和 2.6），并利用有限域上的 MME 近线性时间算法（定理 3.2）。

算法 4： 整数上的精确 MME

输入： $f(x_1, \ldots, x_m) \in \mathbb{Z}[x_1, \ldots, x_m]$ 和 $\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(N)} \in \mathbb{Z}^n$, 以及一个整数 $s > 0$ 使得 f 的所有系数、$\mathbf{a}^{(i)}$ 的坐标以及 $f(\mathbf{a}^{(i)})$ 的所有求值结果其绝对值均不超过 2^s。

输出： 对 $b_i = f(\mathbf{a}^{(i)})$ 在 $i \in [N]$ 处的求值结果。

计算前 s 个素数 $\{p_1, \ldots, p_s\}$。

令 $L \le s$ 为满足 $p_1 \cdots p_L =: M > 2^{s+1}$ 的最小整数。

对于 $\mathbf{e} \in \{0, \ldots, d-1\}^m$ **执行**

  计算 $f_{\mathbf{e}}^{(\ell)} = f_{\mathbf{e}}$ 模 p_ℓ 对于 $\ell \in L$ 使用 引理 2.5。

对于 $i \in [N], k \in [m]$ **执行**

  计算 $a_{i,k,\ell} = \mathbf{a}^{(i)}{}_k$ 模 p_ℓ 对于 $\ell \in L$ 使用 引理 2.5。

对于 $\ell \in [L]$ **执行**

  令 $f^{(\ell)}(x_1, \ldots, x_m) = \sum_{\mathbf{e}} f^{(\ell)}_{\mathbf{e}} \mathbf{x}^{\mathbf{e}} \in \mathbb{F}_{p_i}[\mathbf{x}]$.

  令 $\mathbf{a}^{(i,\ell)} = (a_{i,1,\ell}, \ldots, a_{i,m,\ell}) \in \mathbb{F}_{p_\ell}^m$ 对每个 $i \in [N]$ 成立。

  使用 $b_{i,\ell} = f^{(\ell)}(\mathbf{a}^{(i,\ell)})$ 对所有 $i \in [N]$ 计算算法 3。

对 $i \in [N]$ **执行**

  计算唯一的 $b_i \in [-M/2, M/2]$，使得 $b_i = b_{i,\ell}$ 模 p_ℓ 对所有 $\ell \in [L]$, 成立，使用 引理 2.6。

返回 $\{b_i : i \in [N]\}$。

定理 *4.1* 的证明。我们保证 $\left|f(\mathbf{a}^{(i)})\right| < 2^s$ 对所有 $i \in [N]$ 都成立。因此，根据中国剩余定理，只需计算 $f(\mathbf{a})$ 模 p_i 对每个 $i \in [L]$ 都成立，因为 $p_1 \cdots p_L > 2^{s+1}$。因此，算法 4 的正确性显而易见。至于运行时间，我们将进行与算法 3 非常相似的分析。

根据引理 2.5，可知第 1 行至第 2 行需要$O(\tilde{s})$的时间。根据素数定理[Had96， LVP97]，我们还可知每个$p_i = \tilde{O}(s)$，因此$p_1 \cdots p_L < 2^{s+1} \cdot \tilde{O}(s)$。

根据 定理 3.1，第 10 行 的运行时间为 $((d^m + Nm) \cdot \log p_\ell)^{1+o(1)}$ 因此，第 7 行 至第 10 行 的整个循环耗时 $((d^m + Nm)(\sum_\ell \log p_\ell))^{1+o(1)} = ((d^m + Nm) \cdot s)^{1+o(1)}$。

最后，根据引理 2.6，第 11 行 至第 12 行 的整个循环耗时 $(Nm) \cdot \tilde{O}(\log M) = ((d^m + Nm) \cdot s)^{1+o(1)}$。因此，算法 4 的运行时间为$((d^m + Nm) \cdot s)^{1+o(1)}$，如前所述。

□

备注 4.2。如果已知f的所有系数以及这些点的所有坐标都是整数，且其绝对值不超过 2^s，但没有关于求值结果位复杂度的先验界，那么对求值结果大小的一个朴素界为

$$|f(\mathbf{a})| \leq d^m \cdot 2^s \cdot 2^{sdm} \leq 2^{sdm+s+m\log d}.$$

因此，我们可以在定理 4.1 中使用 $s' = (sdm + s + m \log d)$，从而使时间复杂度上界为 $((d^m + Nm) \cdot (sdm))^{1+o(1)}$。若 m 是增长函数，则输出复杂度相对于输入复杂度为近线性，因为 $poly(d) = (d^m + Nm)^{o(1)}$。但是，当 m 为常数时，由于额外的 d 因子，复杂度关于输入规模 $(d^m + Nm) \cdot s$ 呈超线性。不过，在这种情况下，运行时间略差是可以预期的，因为最坏情况下输出复杂度为 $\Omega(N \cdot sdm)$。◊

5 实数上的近似 MME

本节中，我们假设输入中的所有实数都位于区间 $(-1,1)$ 内。

备注 5.1 (关于常数绝对值的限制)。给定任意多项式 $f(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$，我们可以用最大系数对该多项式进行缩放，并在缩放后的多项式 $\tilde{f}$ 上运行近似 *MME*。若有 $|\tilde{f}(\mathbf{a}) - \beta_i| \leq \varepsilon$，则立即可得 $|f(\mathbf{a}) - (\text{最大值 } |f_\mathbf{e}|)\ \beta_i| \leq \varepsilon \cdot (\text{最大值 } |f_\mathbf{e}|)$。因此，不失一般性，我们可以假设 f 的所有系数的绝对值至多为 1。

然而，假设所有求值点的坐标的绝对值都不超过 *1*，并非无损一般性，但仍有充分依据。即使对于一元整数多项式，求值 $f(\mathbf{a})$ 也可能大至 $|\mathbf{a}|^d$，其中 $|\mathbf{a}| = \max |\mathbf{a}_i|$。因此，*MME* 的输出比特复杂度可能达到 $O(d \cdot N)$，这比输入的位复杂度呈超线性增长。

要求求值点的坐标绝对值至多为1，这样可以确保求值结果的绝对值不会过大，从而使近似 *MME* 的近线性时间求解更有意义。◊

5.1 问题陈述与算法

现在，我们给出实数域上近似 MME 的精确问题陈述及我们的结果。

输入： 一个多项式 $f(x_1,\dots,x_m)\in\mathbb{R}_{(-1,1)}[x_1,\dots,x_m]$，其单变量次数小于$d$，并以一份列表给出：每个系数都有$d^m$ 个高效近似预言机；一组点$\mathbf{a}^{(1)},\dots,\mathbf{a}^{(N)}\in(-1,1)^m$，其中每个点的坐标也通过高效近似预言机提供；以及一个精度参数 t。

输出： 有理数 $b_1,\dots,b_N$，满足 $\left|f(\mathbf{a}^{(i)})-b_i\right|<1/2^t$，对所有 $i\in[N]$均成立。

定理 5.2 (实数上的近似 MME**)。** 存在一种确定性算法（即 算法 *5*），该算法在输入上述内容后返回上述所需输出，并且当所有 $m\in\mathbb{N}$ 且 $d\in\mathbb{N}$ 足够大时，运行时间为$((d^m+Nm)\cdot t)^{1+o(1)}$。

本节其余部分将用于证明上述定理。

总体思路： 该算法将问题恰当地归约为整数上的精确 MME 问题（定理 4.1）。我们将把每个实数替换为适当选取的近似值，其形式为 $a_i/2^k$（其中 $k=O(t)$ 足够大），使得扰动后多项式在扰动后各点上的求值结果与原始求值结果相差不大。由于现在所有分母的形式均为 2^k，我们可以消去分母，从而将问题归约为计算整数上的 MME。

不出所料，为确保整个算法以近线性时间运行，还需要处理一些细节。

对f的系数进行舍入

令k为稍后选定的参数。将多项式 $\lfloor f\rceil_k$ 定义为

$$\lfloor f\rceil_k(x_1,\dots,x_m):=\sum_{\mathbf{e}}\lfloor f_{\mathbf{e}}\rceil_k\cdot\mathbf{x}^{\mathbf{e}}.$$

观察 5.3（系数舍入导致的误差f）。对于任意 $\mathbf{a}\in(-1,1)^m$，有

$$|f(\mathbf{a})-\lfloor f\rceil_k(\mathbf{a})|\le 1/2^{k-m\log d}.$$

证明。

$$\begin{aligned} f(\mathbf{a})-\lfloor f\rceil_k(\mathbf{a})&=\sum_{\mathbf{e}}(f_{\mathbf{e}}-\lfloor f_{\mathbf{e}}\rceil_k)\cdot\mathbf{a}^{\mathbf{e}}\\ \implies |f(\mathbf{a})-\lfloor f\rceil_k(\mathbf{a})|&\le\sum_{\mathbf{e}}|f_{\mathbf{e}}-\lfloor f_{\mathbf{e}}\rceil_k|\cdot|\mathbf{a}^{\mathbf{e}}|\le d^m\cdot 1/2^k. \end{aligned}$$

□

舍入点

令k为稍后选定的参数。对于任意$\mathbf{a} = (a_1, \ldots, a_m) \in (-1,1)^m$，定义$\lfloor \mathbf{a} \rceil_k$为

$$\lfloor \mathbf{a} \rceil_k := (\lfloor a_1 \rceil_k, \ldots, \lfloor a_m \rceil_k).$$

观察 5.4 (舍入点导致的误差)。设$\mathbf{e} = (e_1, \ldots, e_m) \in \{0, \ldots, d-1\}^m$且$\mathbf{a} \in (-1,1)^m$。假设$k \in \mathbb{N}$满足$2^k > 4d^2m^2$。那么，

$$\left| \mathbf{a}^{\mathbf{e}} - \lfloor \mathbf{a} \rceil_k^{\mathbf{e}} \right| \leq 1/2^{k - \log(4dm)}$$

证明。注意，所有$a_i \in (-1,1)$。设$\delta_i = \lfloor a_i \rceil_k - a_i$，其中$i \in [m]$；我们有$|\delta_i| \leq 1/2^k \leq 1/4d^2m^2$。因此，

$$\begin{aligned}
\lfloor a_1 \rceil_k^{e_1} \cdots \lfloor a_m \rceil_k^{e_m} &= (a_1 + \delta_1)^{e_1} \cdots (a_m + \delta_m)^{e_m} \\
&= a_1^{e_1} \cdots a_m^{e_m} + \sum_{\substack{j_1 \leq e_1, \ldots, j_m \leq e_m \\ \text{not all } j_i = 0}} \binom{e_1}{j_1} \cdots \binom{e_m}{j_m} \cdot \prod_{i=1}^{m} \left(a_i^{e_i - j_i} \cdot \delta_i^{j_i} \right)
\end{aligned}$$

$$\begin{aligned}
\implies \left| \lfloor a_1 \rceil_k^{e_1} \cdots \lfloor a_m \rceil_k^{e_m} - a_1^{e_1} \cdots a_m^{e_m} \right| &\leq \left| \sum_{\substack{j_1 \leq e_1, \ldots, j_m \leq e_m \\ \text{not all } j_i = 0}} \binom{e_1}{j_1} \cdots \binom{e_m}{j_m} \cdot \prod_{i=1}^{m} \delta_i^{j_i} \right| \\
&\leq \left| \prod_{i=1}^{m} \left(1 + \delta_{j_i}\right)^d - 1 \right| \\
&\leq (1 + 2d(1/2^k))^m - 1 \leq 4dm(1/2^k). \quad \text{(Lemma 2.1)} \qquad \square
\end{aligned}$$

处理变量数量过少的情况

为使变量更具提示性，我们将重新命名变量，并令$f(x_1, \ldots, x_c)$为一个关于$\mathbb{R}_{(-1,1)}[x_1, \ldots, x_c]$的$c$元多项式，且每个变量的次数都小于$D$。我们希望在点$\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(N)} \in (-1,1)^c$处对该多项式求值。

再次，令$\mathrm{d} = \lfloor \log \mathrm{D} \rfloor$，并令$\mathrm{m}$为满足$\mathrm{d}^m > \mathrm{D}$的最小整数。注意，$\mathrm{d}^m > \mathrm{D} \geq \mathrm{d}^{m-1}$，且$\mathrm{m} = \Theta(\log \mathrm{D} / \log\log \mathrm{D})$。定义多项式$\mathrm{g}(\mathrm{y}_{1,1}, \ldots, \mathrm{y}_{c,m}) = \Phi^{-1}\mathrm{d,m;c}(\mathrm{f})$，其定义见定义 2.2。定义$\mathbf{a}^{(i)} = \psi_{d,m;c}(\mathbf{a}^{(i)})$。根据观察 2.3，对于所有$\mathrm{i} \in [\mathrm{N}]$，我们有$\mathrm{f}(\mathbf{a}^{(i)}) =\sim \mathrm{g}(\mathbf{a}^{(i)})$。

即使$\mathbf{a}^{(i)}$仅由有理数构成，与 定理 3.2 中的情形不同，在该情形下我们$\sim$可以使用观察 3.3，$\mathbf{a}^{(i)}$中的有理数由于$\sim$幂运算而具有大得多的位复杂度。然而，根据引理 2.8，我们可以为$\mathbf{a}^{(i)}$构造高效近似预言机，这已足以支持我们的算法。

5.2 归约到整数上的精确 MME

根据上一小节，我们现在可以不失一般性地假设，所处理的是一个 m 元多项式 $f(x_1,\ldots,x_n)$，其单变量次数小于 d，其中 m 和 d 都是渐增参数；我们希望在 N 个点 $a^{(1)},\ldots,a^{(N)}\in(-1,1)^m$ 上计算该多项式，所有系数和坐标均通过运行时间为 $\tilde{O}(k+O(m\log d))$ 的近似预言机提供。我们希望计算整数 $b_1,\ldots,b_N$，使得 $\left|f(a^{(i)})-b_i/2^t\right|<1/2^t$。下面介绍该算法（算法 5）。

算法 5： approximate-MME-Reals

输入： 一个m元多项式$f(x_1,\ldots,x_m)\in\mathbb{R}_{(-1,1)}[\mathbf{x}]$，其单变量次数小于 d，以及点$\mathbf{a}^{(1)},\ldots,\mathbf{a}^{(N)}\in\mathbb{R}^m_{(-1,1)}$（所有实数均通过近似预言机提供）和一个整数$t>0$。

输出： 整数$b_1,\ldots,b_N$，使得 $\left|\text{有限域}(\mathbf{a}^{(i)})-b_i/2^t\right|<1/2^t$ 对所有 $i\in[N]$。

1 若 $m<\log\log d$，**则**

2 令 $d'=\lfloor\log d\rfloor$ 和 m' 为满足 $(d')^m$ 的最小整数 $>d$.

3 将 f 替换为 $\Phi^{-1}{}_{d',m'}\circ\psi_{d',m'}(f)$，并将每个 $\mathbf{a}^{(i)}$ 替换为 $\psi_{d',m'}(\mathbf{a}^{(i)})$。

4 令 $k_1=\lceil t+m\log d+2\rceil$ 和 $k_2=\lceil t+m\log d+\log(4md)+2\rceil$；令 $k=\max(k_1,k_2)=k_2$。

5 计算 $\lfloor f\rceil_{k_1}=\sum_{\mathbf{e}}g_{\mathbf{e},k_1}/2^{k_1}\cdot\mathbf{x}^{\mathbf{e}}=1/2^{k_1}\cdot\sum_{\mathbf{e}}g_{\mathbf{e},k_1}\cdot\mathbf{x}^{\mathbf{e}}$。

6 对$i\in[N]$ **执行**

7 计算 $\lfloor\mathbf{a}^{(i)}\rceil_{k_2}=(a_{i,1,k_2}/2^{k_2},\ldots,a_{i,m,k_2}/2^{k_2})=1/2^{k_2}\cdot(a_{i,1},k_2,\ldots,a_{i,m,k_2})$.

8 Let $\widehat{\mathbf{a}^{(i)}}=(a_{i,1},k_2,\ldots,a_{i,m,k_2})$.

9 计算多项式 $G(x_1,\ldots,x_m)$ 定义为

$$G(x_1,\ldots,x_n)=\sum_{\mathbf{e}\in\{0,\ldots,d-1\}^m}g_{\mathbf{e},k_1}\cdot 2^{(k_2dm)-k_2|\mathbf{e}|}\cdot\mathbf{x}^{\mathbf{e}}$$

其中 $|\mathbf{e}|$ 表示坐标之和（即单项式的次数 $\mathbf{x}^{\mathbf{e}}$）。10 运行 算法 4 (整数精确 MME)，输入 $\left(G,\left(\widehat{(1)\mathbf{a}},\ldots,\widehat{\ }\mathbf{a}^{(N)}\right),s=3kdm\right)$，得到 $B_1,\ldots,B_N$，使得对所有 $i\in[N]$, 均有

$$B_i=G(\widehat{\mathbf{a}^{(i)}}).$$

令 $b_i=\lfloor B_i/2^{k_1+k_2dm-t}\rceil$ 对每个$i\in[N]$。

11 **返回** $(b_1,\ldots,b_N)$。

正确性证明： 不失一般性，我们可以假设 d,m 是增长参数（从 第 1 行 到 3）。

注意，对于任意 $\mathbf{a}^{(i)}$，我们有

$$\left|f(\mathbf{a}^{(i)})-\lfloor f\rceil_{k_1}(\lfloor\mathbf{a}^{(i)}\rceil_{k_2})\right|\le\left|f(\mathbf{a}^{(i)})-\lfloor f\rceil_{k_1}(\mathbf{a}^{(i)})\right|+\left|\lfloor f\rceil_{k_1}(\mathbf{a}^{(i)})-\lfloor f\rceil_{k_1}(\lfloor\mathbf{a}^{(i)}\rceil_{k_2})\right|$$

$\leq 1/2^{t+2} + 1/2^{t+2} \leq 1/2^{t+1}$.

其中，最后一个不等式用到了观察 5.3 和观察 5.4，并采用了我们对k_1 和k_2的选择。因此，只需对每个$i \in [N]$计算 $\lfloor f \rceil_{k_1}\left(\lfloor \mathbf{a}^{(i)} \rceil_{k_2}\right)$。多项式 $\lfloor f \rceil_{k_1}$在第 5 行 中计算，而点 $\lfloor \mathbf{a}^{(i)} \rceil_{k_2}$在第 6 行 至8中计算。令$\mathbf{a}^{(i)} = 2^{k_2}\lfloor \mathbf{a}^{(i)} \rceil_{k_2} \in (-2^{k_2}, 2^{k_2})^m$。由于 $2^{k_1} \cdot \lfloor f \rceil_{k_1}$ 的每个系数的绝对值均不超过 2^{k_1}，因此我们有

$$\left|G(\widehat{\mathbf{a}^{(i)}})\right| = \left|\sum_{\mathbf{e}\in\{0,\ldots,d-1\}^m} g_{\mathbf{e},k_1} \cdot 2^{(k_2 dm)-k_2|\mathbf{e}|} \cdot \widehat{\mathbf{a}^{(i)}}^{\mathbf{e}}\right| \leq d^m \cdot 2^{k_1} \cdot 2^{k_2 dm} \cdot 2^{k_2 dm} \leq 2^{3kdm}.$$

根据$G(x_1,\ldots,x_m)$的定义，注意到

$$\begin{aligned} G(\widehat{\mathbf{a}^{(i)}}) &= \sum_{\mathbf{e}\in\{0,\ldots,d-1\}^m} g_{\mathbf{e},k_1} \cdot 2^{(k_2 dm)-k_2|\mathbf{e}|} \cdot \widehat{\mathbf{a}^{(i)}}^{\mathbf{e}} \\ &= \sum_{\mathbf{e}\in\{0,\ldots,d-1\}^m} g_{\mathbf{e},k_1} \cdot 2^{(k_2 dm)} \cdot \left(1/2^{k_2} \cdot \widehat{\mathbf{a}^{(i)}}\right)^{\mathbf{e}} \\ &= 2^{(k_2 \cdot d \cdot m)} \cdot \sum_{\mathbf{e}\in\{0,\ldots,d-1\}^m} g_{\mathbf{e},k_1} \cdot \lfloor \mathbf{a}^{(i)} \rceil_{k_2}^{\mathbf{e}} \\ &= 2^{k_1+k_2 dm} \cdot \lfloor f \rceil_{k_1}(\lfloor \mathbf{a}^{(i)} \rceil_{k_2}). \end{aligned}$$

由于定理 4.1 正确计算了$G(\mathbf{x})$ 在$\mathbf{a}^{(i)}$处的求值结果，因此对于每个$i \in [N]$，都有

$$1/2^{k_1+k_2 dm} \cdot G\left(\widehat{\mathbf{a}^{(i)}}\right) = \lfloor f \rceil_{k_1}(\lfloor \mathbf{a}^{(i)} \rceil_{k_2}) = B_i/2^{k_1+k_2 dm}.$$

最后，如果$b_i = \lfloor B_i 2^{k_1+k_2 dm-t} \rceil$，那么

$$\left|b_i/2^t - B_i/2^{k_1+k_2 dm}\right| = 1/2^t \cdot \left|b_i - B_i/2^{k_1+k_2 dm-t}\right| \leq 1/2^{t+1}.$$

因此，

$$\left|f(\mathbf{a}^{(i)}) - b_i/2^t\right| \leq \left|f(\mathbf{a}^{(i)}) - \lfloor f \rceil_{k_1}(\lfloor \mathbf{a}^{(i)} \rceil_{k_2})\right| + \left|\lfloor f \rceil_{k_1}(\lfloor \mathbf{a}^{(i)} \rceil_{k_2}) - b_i/2^t\right| \leq 1/2^t.$$

运行时间分析： 经过第 1 至3行后，我们可以假设d、$m = \omega(1)$ 以及f 的所有系数和点的坐标均通过近似预言机提供，其运行时间为 $\tilde{O}(r + m\log d)$，用于计算一个r位近似值。

第 5 行至第 8 行整体需要时间

$$(d^m + Nm) \cdot \tilde{O}(k + O(m\log d)) = (d^m + Nm) \cdot \tilde{O}(t + O(m\log d)) = ((d^m + Nm) \cdot t)^{1+o(1)}.$$

计算 $G(\mathbf{x})$ 的系数需要时间 $(d^m)\cdot\tilde{O}(kdm)$。根据定理 4.1，第 10 行需要时间

$$((d^m+Nm)\cdot 3kdm)^{1+o(1)}=((d^m+Nm)\cdot t)^{1+o(1)}.$$

因此，算法 5 的总体耗时为$((d^m+Nm)\cdot t)^{1+o(1)}$。

至此，定理 5.2的证明。 □

6 有理数上的精确 MME（已知输出复杂度）

现在，我们利用针对实数上的近似 MME 的算法，获得一个用于有理数域上的精确 MME 的快速算法。我们首先正式陈述要解决的精确问题，然后介绍算法所需的一些预备知识。

6.1 问题陈述

输入： 一个多项式$f(x_1,\ldots,x_m)\in\mathbb{Q}_{(-1,1)}[x_1,\ldots,x_m]$，其单变量次数小于 d，给定为一个d^m的列表、一组点$\mathbf{a}^{(1)}$、…、$\mathbf{a}^{(N)}\in\mathbb{Q}^m_{(-1,1)}$、一个整数参数$s>0$，使得多项式$f$的系数、各点的坐标以及求值结果$f(\mathbf{a}^{(i)})$ 中的所有有理数都可表示为p/q 的形式，其中 $|p|,|q|<2^s$。

输出： 整数 $b_1,\ldots,b_N,c_1,\ldots,c_N$，使得 $f(\mathbf{a}^{(i)})=b_i/c_i$ 对所有 $i\in[N]$均成立。

定理 6.1 (有理数上的精确 MME**)。** 存在一种确定性算法（即 算法 7），在输入上述内容后，返回上述所需的输出，并且当所有 $m\in\mathbb{N}$ 以及充分大的 $d\in\mathbb{N}$时，其运行时间为 $((d^m+Nm)\cdot s)^{1+o(1)}$。

主要思路： 主要思路是将问题归约为近似 MME（定理 5.2），然后进行有理数重构。若能以合理的精度（取决于输出保证 s）计算出 $f(\mathbf{a}^{(i)})$，就可以从中精确恢复该有理数。在介绍上述定理的算法之前，我们先讨论连分数的概念；它是重构目标有理数的关键。

6.2 连分数、有理数近似和扩展欧几里得算法

定义 6.2 （连分数）。由整数序列$[q_1,\ldots,q_t]$ 表示的有限连分数计算有理数

$$q_1+\cfrac{1}{q_2+\cfrac{1}{\ddots+\cfrac{1}{q_{t-1}+\cfrac{1}{q_t}}}}.$$

由无限连分数由无限整数序列 $[q_1, q_2, \ldots]$ 满足[4]$q_2, \ldots, q_n > 0$ 表示，若其计算实数 α 当且仅当

$$\alpha = q_1 + \cfrac{1}{q_2 + \cfrac{1}{q_3 + \cfrac{1}{\ddots}}}.$$

也就是说，极限 $\lim_{n\to\infty} |\alpha - [q_1, \ldots, q_n]| = 0$。 ♢

我们列出连分数的一些基本性质，这些性质可见于大多数标准教材（参见 Schmidt，[Sch80，第 1]章）。

命题 6.3 （连分数的唯一性（Sch [Sch80]中的引理 4C、4D））。每个实数都有唯一的连分数展开，但有以下例外：

1. 如果 α 是整数，那么 α 恰有两种连分数表示，分别为 $[\alpha]$ 和$[\alpha - 1, 1]$。*2.* 如果 α 是非整数有理数，那么 α恰有两种连分数表示：一种形式为 $[q_1, \ldots, q_n]$，其中 $q_n \geq 2$；另一种为 $[q_1, \ldots, q_n - 1, 1]$。*3.* 如果 α 是无理数，那么 α 恰有一种连分数表示。

定义 6.4 （渐近分数）。对于实数 α 满足 $[q_1, q_2, \ldots]$ 是唯一的[5]，与第 i 个前缀 $[q_1, \ldots, q_i]$ 相对应的有理数 a_i/b_i，称为 α 的第 i 个渐近分数。♢

引理 6.5 （渐近分数的性质）。设 $\{a_i/b_i\}_i$ 是实数 $\alpha = [q_1, q_2, \ldots]$ 的渐近分数。那么

1. 对任意 $n \geq 3$，有

$$a_n = q_n a_{n-1} + a_{n-2},$$
$$b_n = q_n b_{n-1} + b_{n-2}.$$

特别地，分母序列 $\{b_n\}_{n\geq 2}$ 是递增的。

2. 对所有 $n \geq 1$，

$$\frac{a_{n+1}}{b_{n+1}} - \frac{a_n}{b_n} = \frac{(-1)^{n-1}}{b_n(q_{n+1}b_n + b_{n-1})} = \frac{(-1)^{n-1}}{b_n b_{n+1}}.$$

3. 对任意 $n \geq 1$，除非 $\alpha = a_n b_n$，否则我们有

$$\frac{1}{b_n(b_n + b_{n+1})} \leq \left|\alpha - \frac{a_n}{b_n}\right| \leq \frac{1}{b_n b_{n+1}}.$$

[4]传统上，满足此条件的连分数称为“简单”连分数，但我们将省略这一限定词，因为我们只讨论满足这一附加约束的连分数。[5]按照惯例，对于有理数，我们只考虑 命题 6.3 的第 1 和 第 2 种形式中所述的连分数表示。

4. 假设 a/b 是一个满足 $|\alpha - a/b| < 1/2b^2$ 的有理数。那么，a/b 是 α的一个渐近分数。

证明。 第 1 项、2 和4 分别就是 [Sch80、引理 3A、引理 3E、定理 5C] 。

对于 第 3 项，若 $\alpha \neq a_n/b_n$，我们有 q_{n+1} 存在。设 $\alpha_{n+1} = [q_{n+1},\ldots]$。那么，我们可以滥用符号，将 α 表示为“连分数” $\alpha = [q_1,\ldots,q_n,\alpha_{n+1}]$。 此表达式的第 2 项给出

$$\left|\alpha - \frac{a_n}{b_n}\right| = \frac{1}{b_n(\alpha_{n+1}b_n + b_{n-1})}.$$

注意，$q_{n+1} \leq \alpha_{n+1} \leq q_{n+1} + 1$ 因此

$$\begin{aligned}
&\left|\alpha - \frac{a_n}{b_n}\right| = \frac{1}{b_n(\alpha_{n+1}b_n + b_{n-1})} \\
&\qquad \leq \frac{1}{b_n(q_{n+1}b_n + b_{n-1})} = \frac{1}{b_n b_{n+1}}, \quad \text{(by Item 1)} \\
\text{and} \quad &\left|\alpha - \frac{a_n}{b_n}\right| = \frac{1}{b_n(\alpha_{n+1}b_n + b_{n-1})} \\
&\qquad \geq \frac{1}{b_n(q_{n+1}b_n + b_{n-1} + b_n)} = \frac{1}{b_n(b_n + b_{n+1})}.
\end{aligned}$$

□

扩展欧几里得算法

与连分数密切相关的是经典的扩展欧几里得算法，用于计算两个数的最大公约数。

定义 6.6 (余数和商序列)。 对于一对整数 a，$b > 0$，我们定义这对 (a, b) 的余数序列 $\{r^i\}_{i=0,\ldots,t+1}$ 和商序列 $\{q^i\}_{i=1,\ldots,t}$ 如下：

- $r_0 = a$ 和 $r_1 = b$，
- 对于所有 $i \geq 1$，定义 qi，ri_{+1} 分别为 ri_{-1} 除以 ri 所得的商和余数。因此，

$$r_{i+1} = r_{i-1} \bmod r_i = r_{i-1} - q_i r_i.$$

- rt_{+1} 是序列中第一个等于零的元素。 ♢

观察 6.7 （有理数的连分数及商序列）。假设、$b > 0$ 是两个整数，且 $\{q_1,\ldots,q_t\}$ 是相应的商序列。那么，有理数 a/b 的连分数表示为 $[q_1,\ldots,q_t]$:

$$\frac{a}{b} = q_1 + \cfrac{1}{q_2 + \cfrac{1}{\ddots + \frac{1}{q_t}}}.$$

□

计算给定两个整数的最大公约数，更一般地，计算整个商序列，都可以在确定性近线性时间内完成；这归功于 Knuth 和 Schön- hage（参见 Möller [Möl08]，其中有完整的说明和详细的历史介绍）。

定理 6.8 (快速扩展欧几里得算法（参见 *Möller*Möl）。 存在一种确定性算法，对于输入的一对整数 $a > b > 0$ 且 $a,b \leq 2^s$，可在 $\tilde{O}(s)$ 时间内计算这对数的完整商序列 $q_1,\ldots,q_t$，其中这对数为 (a,b)。

推论 6.9 (快速计算渐近分数)。 存在一种确定性算法，对于输入的一对整数 $M,N > 0$ 且 $M,N \leq 2^s$，以及一个整数 $i > 0$，可在 $\tilde{O}(s)$ 运行时间内计算整数 a_i,b_i，使得 a_i/b_i 是有理数 M/N的第 i 个渐近分数。

证明。 令 $q_1,\ldots,q_t$ 为数对 (M,N)的商序列，可使用定理 6.8 在 $\tilde{O}(s)$ 时间内计算得出。根据 观察 6.7，这是 M/N 的连分数表示。因此，容易看出

$$\begin{bmatrix} a_i & a_{i-1} \\ b_i & b_{i-1} \end{bmatrix} = \begin{bmatrix} q_1 & 1 \\ 1 & 0 \end{bmatrix} \cdots \begin{bmatrix} q_i & 1 \\ 1 & 0 \end{bmatrix}$$

其中 a_j/b_j 是第j个渐近分数。注意，$|q_j| < r_{j-1}/r_j$，其中 $\{r_0,.....r_t\}$ 是相应的余数序列，因此我们有 $|q_1 \cdots q_t| \leq M \leq 2^s$。因此，该矩阵乘积可在 $\tilde{O}(s)$时间内计算。

□

6.3 有理数重构

引理 6.10 (快速有理数重构)。 存在一种确定性算法（即算法 6），其输入为整数参数 $s > 0$ 和整数 A,B，并保证$|B| < 2^{2s+1}$ 且存在唯一的有理数（最简形式）a/b，其中 $|b| < 2^s$，并且

$$\left|\frac{A}{B} - \frac{a}{b}\right| < \frac{1}{2^{2s+1}},$$

在 $\tilde{O}<s>$ 时间内找到整数 a,b。

证明。 利用推论 6.9 和引理 6.5，该算法显而易见。

算法 6： 快速有理数重构

输入： 整数 A、B 和整数参数 $s > 0$，使得 $|A|$、$|B| \leq 2^{2s+1}$，并且存在一个有理数 a/b，使得 $|b| < 2^s$ 且 $|A/B - a/b| < 1/2^{2s+1}$。

输出： 整数 a、b。

1 使用 定理 6.8，计算 $q_1,\ldots,q_\ell$ 对应于数对 A、B。

2 使用 推论 6.9 和二分查找，求满足条件的最大下标i，使得第 i 个渐近分数 a_i/b_i 满足 $|b_i| < 2^s$。

3 返回 a_i，b_i。

算法的运行时间显然为 $\tilde{O}(s)$ 正如所声称的那样，为 $\ell = O(\log(A+B)) = O(s)$ 因此，至多需要 $O(\log \ell) = O(\log s)$ 次调用 推论 6.9，在 第 2 行中。

为证明正确性，假设 A/B 处于最简形式。由于已知 $b_1 = 1$，令 i 为满足条件 b_i 的渐近分数 a_i/b_i 的分母 $b_i < 2^s$ 所对应的最大指标。如果这是最后一个渐近分数，那么 $A/B = a_i/b_i$，问题得证。因此，我们可以假设 $A/B \neq a_i/b_i$。

由于已知 $|A/B - a/b| < 1/2b^2 < 1/2b^2$，根据引理 6.5 第 4 项，$a/b$ 是 A/B 的一个渐近分数。对于任意 $\ell > i$，第 ℓ 个渐近分数 a_ℓ/b_ℓ 的分母大于 2^s。对于任意 $j < i$，根据引理 6.5 第 3 项和第 1 项，我们有

$$\left|\frac{A}{B} - \frac{a_j}{b_j}\right| \geq \frac{1}{b_j(b_j + b_{j+1})} > \frac{1}{2 \cdot b_i^2} \geq \frac{1}{2^{2s+1}}.$$

因此，a/b 必定是第 i 个渐近分数 a_i/b_i。 □

6.4 有理数上的精确 MME 算法

现在我们已 具备了描述证明定理 6所需的全部要素。 1.

算法 7： Exact-MME-Rationals

输入： 多项式 $f(x_1, \ldots, x_m) \in \mathbb{Q}_{(-1,1)}[\mathbf{x}]$，个点 $\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(N)} \in \mathbb{Q}^m_{(-1,1)}$，所有有理数均以分子和分母的形式给出，并给定一个整数参数 s，使得多项式 f 的所有系数的分子和分母，点的坐标和求值结果 $f(\mathbf{a}^{(i)})$ 至多为 2^s。

输出： 整数 b_1、b_N 和 c_1、c_N，满足 $f(\mathbf{a}^{(i)}) = b_i/c_i$，对所有 $i \in [N]$ 均成立。

1 使用所需近似预言机的分子和分母，运行 approximate-MME-Reals $\left(f、\left\{\mathbf{a}^{(1)}、\ldots\ldots、\mathbf{a}^{(N)}\right\}、t = 2s+1\right)$（算法 5），得到整数 $(B_1、\ldots\ldots、B_N)$，使得

$$\left|f(\mathbf{a}^{(i)}) - \frac{B_i}{2^t}\right| < \frac{1}{2^t} = \frac{1}{2^{2s+1}}.$$

2 **对** $i \in N$ **执行**

3 运行 快速有理数重构 $(B_i, 2^{2s+1}, s)$ (算法 6) 以得到 b_i, c_i 其中 $|c_i| < 2^s$ 满足

$$\left|\frac{B_i}{2^{2s+1}} - \frac{b_i}{c_i}\right| < \frac{1}{2^{2s+1}}。$$

4 **返回** $(b_1, \ldots, b_N), (c_1, \ldots, c_N)$.

算法正确性算法 7 可由定理 5.2 和引理 6.10。 定理 5.2断言算法 5 正确地提供了求值结果所需的近似值，且

引理 6.10 断言 算法 6 能够重构出正确的有理数。

关于运行时间，给定分子和分母后，我们可以为每个有理数构建近似预言机，所需时间为近线性。因此，第 1 行 需要$((d^m + Nm) \cdot s)^{1+o(1)}$ 时间，而第 2 行 到第 3 行的循环需要 $\tilde{O}(N \cdot s)$ 时间。因此，总运行时间为$((d^m + Nm) \cdot s)^{1+o(1)}$，如所述。

这就完成了定理 6.1的证明。 □

7 复数上的近似 MME

本节简要讨论定理 5.2 向复数域的推广。正如预备知识中所讨论的，此时域常数由两个近似预言机给出：一个用于复数的实部，另一个用于虚部。除定理 3.1 证明中的思路外，这一推广所需的思路相当标准，由 Kedlaya & Umans [KU11] 为非素数阶有限域上的 MME 设计快速算法时提出。这一方法随后也应用于 Bhargava 等人 [BGGKU22], 的工作中，他们同样是在为 MME 设计算法时处理非素数阶有限域。为使讨论简洁并避免重复，我们概述这一推广所需的主要步骤，但省略形式化细节。该算法的结构与算法 5非常相似，只需额外注意一些细节。

与算法 5 的证明一样，我们首先确保底层变量的数量不断增长。接下来，我们用分母为 2^k 的有理数对每个域常数（实部和虚部）进行舍入，其中 k 是稍后选定的一个足够大的整数。此时，我们引入了一些误差（事实证明，当 k 足够大时，误差很小），但将 $\mathbb{C}$ 上的问题实例转化为 $\mathbb{Q}[\omega]$ 上的问题实例，其中 ω 是 -1的平方根。此外，问题中所有域常数的分母都具有 2^k 的形式。现在，我们像算法 5 中那样清除分母，得到一个 MME 实例，其中问题中的常数来自环 $\mathbb{Z}[\omega]$。此时，我们将输入中常数里的 ω 替换为一个新的形式变量 z，并且不再在环 $\mathbb{Z}[\omega]$上运算，而是在环 $\mathbb{Z}[z]/\langle z^2+1\rangle$上运算。注意，这样做足够了，因为给定该环上 MME 的一个解后，只需将 z 替换为 ω，就能得到原问题的解。现在，我们的思路是直接调用针对该问题实例的整数精确 MME 算法（算法 4）。然而，我们不能直接这样做，因为当前实例是在 $\mathbb{Z}[z]/\langle z^2+1\rangle$ 上，而不是所需的 $\mathbb{Z}$ 上。尽管如此，我们仍像算法 4 中那样，选取足够多的素数 $p_1, p_2, \ldots, p_s$，并将 $\mathbb{Z}[z]/\langle z^2+1\rangle$ 上的问题实例分别模这些素数约化，从而对每个 i 得到环 $\mathbb{F}_{p_i}[z]/\langle z^2+1\rangle$ 上的实例。在算法 4 中，我们在此阶段直接调用了 [BGGKU22] 关于素域的结果，然后通过快速中国剩余定理合并输出。然而，在这里，我们得到的是有限环 $\mathbb{F}_{p_i}[z]/\langle z^2+1\rangle$上的实例。不过，这并不会造成问题，因为 Bhargava 等人的算法

在这类环上仍然适用，而且 [BGGKU22] 中的结果和证明确实都是以这种形式表述的。还需注意的一点是，我们在 [BGGKU22] 的 第 3 节 中所做的小幅优化，确保了运行时间对域大小的依赖为近线性；对于这里所用的扩张环，这一结论仍然成立。一旦我们求出了 $\mathbb{F}_{p_i}[z]/\langle z^2+1\rangle$上的所有实例的解，就可以像 算法 4中那样，通过快速中国剩余定理恢复 $\mathbb{Z}[z]/\langle z^2+1\rangle$ 上的解；再对这些求值结果进行适当缩放（同样如 算法 5所示），即可得到原求值结果的近似值，其底层域为 $\mathbb{C}$。误差分析和运行时间界限与 算法 5的分析基本相同。其余细节从略。

8 讨论与开放问题

最后，我们提出一些开放问题。

1. 一个很自然的问题是，能否找到一种在一般域（包括有限域和无限域）上进行多变量多点求值的代数算法。目前，我们只知道如何在小特征有限域上构造这类代数算法 [Uma08, BGKM22]。

2. 前述关于能否为 MME 找到代数算法的问题，在非一致性设定下也很有趣。例如，我们不知道，能否在小特征有限域以外的域上，用近线性（甚至次二次）规模的算术电路计算由多变量 Vandermonde 矩阵给出的线性变换。

3. 除了 Kedlaya 和 Umans 已提到的应用之外，若能进一步找到这些更快算法及其思想的其他应用，也将很有意义 [KU11]。

参考资料

[BGGKU22] Vishwas Bhargava、Sumanta Ghosh、Zeyu Guo、Mrinal Kumar 和 Chris Umans。所有有限域上的快速多变量多点求值。载于 Jelani Nelson 编，《第 63 届 IEEE 计算机科学基础研讨会（FOCS）论文集》，第 221–232 页。2022年。arXiv:2205.00342，eccc:2022/TR22-063。[BGKM22] Vishwas Bhargava、Sumanta Ghosh、Mrinal Kumar 和 Chandra Kanta Mohapatra。小特征下快速代数多变量多点求值及其应用。载于 Stefano Leonardi 和 Anupam Gupta 编，《第 54 届 ACM 计算理论研讨会（STOC）论文集》，第 403–415 页。2022年。arXiv:2111.07572，eccc:2021/TR21-162。[BM74] Allan Borodin 和 Robert Thomas Moenck。快速模变换。《计算机与系统科学杂志》，8(3):366–386，1974年。[GG13] Joachim von zur Gathen 和 Jürgen Gerhard。《现代计算机代数》。剑桥大学出版社，第 3 版，2013年。

[*Had*96] 雅克·*S.* 阿达马。关于函数 $\zeta(s)$的零点分布及其算术后果。《法国数学会通报》，*24:199–220*，*1896*年。[*KU*08] 基兰·*S.* 凯德拉亚和克里斯托弗·乌曼斯。任意特征下的快速模组合成。载于 *R.* 拉维编，《第 *49* 届 *IEEE* 计算机科学基础研讨会（*FOCS*）论文集》，第 *146–155* 页。*2008*年。[*KU*11] ———。快速多项式因式分解与模组合成。《*SIAM* 计算杂志》，*40(6):1767–1802*，*2011*年。（初步版本发表于第 *40* 届 *STOC*，*2008*年，以及第 *49* 届 *FOCS*，*2008*年。）[*LVP*97] 夏尔·让·德·拉·瓦莱-普桑。关于素数理论的解析研究（法语）*[*关于素数理论的解析研究*]*，第 *1–5* 卷。*Hayez*，*1897*年。[*Möl*08] 尼尔斯·默勒。关于舍恩哈格算法与次二次时间整数最大公约数计算。《数学计算》，*77(261):589–607*，*2008*年。[*Mor*21] 纪尧姆·莫罗。用于单变量多项式逼近的新数据结构及其在根隔离、数值多点求值和其他问题中的应用。载于尼希思·维什诺伊编，《第 *62* 届 *IEEE* 计算机科学基础研讨会（*FOCS*）论文集》，第 *1090–1099* 页。*2021*年。*arXiv:2106.02505*。[*NZ*04] 迈克尔·纽斯肯和马丁·齐格勒。二元多项式的快速多点求值。载于苏珊娜·阿尔伯斯和托马什·拉齐克编，《第 *12* 届欧洲算法研讨会（*ESA*）论文集》，《计算机科学讲义》第 *3221* 卷，第 *544–555* 页。施普林格，*2004*年。*arXiv:cs/0403022*。[*Sch*80] 沃尔夫冈·*M.* 施密特。《丢番图逼近》，《数学讲义》第 *785* 卷。施普林格，*1980*年。[*Uma*08] 克里斯托弗·乌曼斯。小特征下的快速多项式因式分解与模组合成。载于辛西娅·德沃克编，《第 *40* 届 *ACM* 计算理论研讨会（*STOC*）论文集》，第 *481–490* 页。*2008*年。