arXiv:2302.03766v1 [math.AC] 7 Feb 2023

二元变量中字典序 Gröbner 基的牛顿迭代

´Eric Schost 和 Catherine St-Pierre

滑铁卢大学计算机科学学院，滑铁卢，安大略省，加拿大

摘要

我们提出了一种 $\mathfrak{m}$-进牛顿迭代，用于求二元变量零维理想的字典序 Gröbner 基，并具有二次收敛速度。我们依赖 Conca 和 Valla 关于此类基中 Syzygy 的一项结构性结果；该结果使他们能够通过仿射参数显式描述这些 Gröbner 基。我们的牛顿迭代直接处理这些参数。

关键词 主分量， $\mathfrak{m}$-进算法，Gröbner 基

1 引言

求解二元多项式方程是计算拓扑或计算机图形学算法中的重要环节。因此，已有大量研究致力于这一问题，并采用符号、数值或混合符号数值技术 [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]。

在许多情况下，这些算法会为给定系统 $f_1, \ldots, f_t$ 在 $\mathbb{K}[x, y]$ 中的解找到一种集合论描述（此处，$\mathbb{K}$ 是一个域）。一种值得注意的实现方式是借助形状引理：在一般坐标下，输出是一对多项式 u, v，它们属于 $\mathbb{K}[x]$，其中 u 无平方因子，并且 $V(\langle f_1, \ldots, f_t\rangle)$ 可由 $u(x) = 0$ 和 $y = v(x)/u'(x)$ 描述（这种关于 y 的有理形式，若 $\mathbb{K} = \mathbb{Q}$，便可精确控制 v 的位大小）。我们可以稍微扩充这种集合论描述：不再要求 u 无平方因子，而是将 $J = \langle f_1, \ldots, f_t\rangle$ 在点 (ξ, ν) 处的重数赋给 u 的一个根 ξ，该根对应于点 (ξ, ν)（并相应调整 v 的定义）。Rouillier 的有理单变量参数化 [17]，采用了这种做法，但这仍只能提供部分信息：例如，它不足以检测局部同构。

为了描述 J 的解，以及 J 在这些零点处的局部结构，自然会转向格雷布纳基。本文正是采用这一方法，重点关注一种 $\mathfrak{m}$-进近似过程。

我们的问题与主要结果 设基域 $\mathbb{K}$ 是整环 $\mathbb{A}$ 的分式域，并取 $f_1, \ldots, f_t \in \mathbb{A}[x, y]$。

进一步考虑理想 $J = \langle f_1$、……、$f_t \rangle$ 在 $\mathbb{K}[x, y]$中的情形。我们感兴趣的是求出 J 本身的 Gr¨obner 基，或者求出它的某些特定主分量的 Gr¨obner 基。因此，我们令 I 为 $\mathbb{K}[x, y]$ 中的一个理想，并假设它是 J 的某些零维主分量的交：典型的研究对象包括：当 $I = J$的维数为零时的 $I = J$，或者当原点在 $V(J)$中是孤立点时， J的 $\langle x$、$y\rangle$-主分量 I。

令 $\mathcal{G} = (g_0$、……、$g_s)$为关于由 $y \succ x$诱导的字典序的 I的最小既约 Gr¨obner 基；这就是我们所研究的对象。

例 1.1。令 $\mathbb{A} = \mathbb{Z}$，因而 $\mathbb{K} = \mathbb{Q}$、 $t = 2$ 以及输入多项式

$$f_1 = -12xy^5 - 20x^2y^4 - 14y^4 - 7x^3y^3 - 3x^2y^2 + 13x^3y - 17xy + 34x^2$$
$$f_2 = -x^2y^4 - 19x^3y^3 + 18xy^3 + 22x^3y^2 + 2x^2y^2 - 10x^2y.$$

令 I为 $\langle f_1$的 $\langle x$、$y\rangle$-主分量；其 Gr¨obner 基 $\mathcal{G}$i s

$$\left| \begin{array}{l} y^4 + \frac{17}{14}xy - \frac{17}{7}x^2, \\ xy^3 - \frac{10}{9}x^3, \\ x^2y - 2x^3, \\ x^4. \end{array} \right. \tag{1}$$

现在令 $\mathfrak{m}$ 为 $\mathbb{A}$ 中的一个极大理想，其剩余域为 $\mathbb{K} = \mathbb{A}/\mathfrak{m}$。从 $\mathcal{G}$ 对 $\mathfrak{m}$ 的约化出发（假设其定义良好），本文旨在说明如何恢复 $\mathcal{G}$ 模 $\mathbb{K}$ $\mathfrak{m}$ 的幂。上文所见的情形 $\mathbb{A} = \mathbb{Z}$ 是最基本的例子；另一种重要情形是“参数”情形，其中 $\mathbb{A} = \mathbb{K}[t_1, \ldots, t_m]$，且 $\mathfrak{m}$ 是形如 $\langle t_1 - \tau_1, \ldots, t_m - \tau_m \rangle$ 的极大理想。

令 $\mathbb{A}_{\mathfrak{m}}$ ($\mathbb{A}_{\mathfrak{m}} \subseteq \mathbb{K}$) 为 $\mathbb{A}$ 在 $\mathfrak{m}$ 处的局部化。对于 $K \geq 0$，存在定义良好的约化算子 $\mathbb{A}_{\mathfrak{m}} \to \mathbb{A}/\mathfrak{m}^K$，记作 $c \mapsto c$ 余数 $\mathfrak{m}^K$；将其逐系数扩展为约化映射 $\mathbb{A}_{\mathfrak{m}}[x, y] \to \mathbb{A}/\mathfrak{m}^K[x, y]$，并进一步扩展到多项式向量。

定义 1.2. 如果满足以下条件，我们称 $\mathfrak{m}$ 关于 $f_1, \ldots, f_t$ 和 $\mathcal{G}$ 是良好的：

- $\mathcal{G}$ 中的所有元素都属于 $\mathbb{A}_{\mathfrak{m}}[x, y]$，
- 由 $\mathcal{G}$余数 $\mathfrak{m}$ 在 $\mathbb{K}[x, y]$ 中生成的理想，是理想 $\langle f_1$ 余数 $\mathfrak{m}$，……，f_t 余数 $\mathfrak{m}\rangle$的某些主分量的交集。

特别地，如果 $\mathfrak{m}$ 是良好的，我们将 $\mathcal{G}_{\mathfrak{m}}$ 写作约化 $\mathcal{G}$ 余数 $\mathfrak{m}$。这些是 $\mathbb{K}[x, y]$中的多项式，并且它们仍然构成字典序 $y \succ x$下的最小、约化 Gr¨obner 基。

例 1.3。 在 例 1.1中， $\mathfrak{m} = \langle 11 \rangle$ 相对于 f_1、……、 f_t 和 $\mathcal{G}_{\mathfrak{m}}$ 是

$$\left| \begin{array}{l} y^4 + 2xy + 7x^2, \\ xy^3 + 5x^3, \\ x^2y + 9x^3, \\ x^4. \end{array} \right.$$

若 $\mathbb{A} = \mathbb{Z}$，则只有有限多个素数 p 不满足此条件。在 $\mathbb{A} = \mathbb{K}[t_1, \ldots, t_m]$ 的情形下，形如 $\langle t_1 - \tau_1$、……、 $t_m - \tau_m \rangle$ 的所有极大理想都是良好的，但位于 $\mathbb{K}^m$ 中某个超曲面上的极大理想 $(\tau_1$、……、 $\tau_m)$ 除外（对坏极大理想数量的定量分析将留待未来工作）。

我们的主要结果是一种基于牛顿迭代的高效提升过程，用于在给定 f_1、……、 f_t、 $\mathcal{G}_{\mathfrak{m}}$ 和 K 的情况下计算 $\mathcal{G}$ rem $\mathfrak{m}^K$。提升方法广泛应用于计算机代数，例如用于求解线性方程组或计算多项式最大公因式，并有两个用途。首先，虽然解决当前问题（此处为计算 I 的 Gr¨obner 基）的算术成本可能很高，但我们的结果将表明，对模 $\mathfrak{m}$ 的幂的近似解进行提升相对简单。其次，这些技术通常用于这样的情形： $\mathbb{A}$ 和 $\mathbb{K}$ 中的元素具有自然的“大小”概念（例如，当 $\mathbb{A} = \mathbb{Z}$ 时的高度，或当 $\mathbb{A} = \mathbb{K}[t_1, \ldots, t_m]$ 时的次数）。此时，直接在 $\mathbb{K}$ 中计算往往会造成显著的“中间表达式膨胀”，即算法执行过程中计算出的多项式可能具有比最终输出更大的系数； $\mathfrak{m}$-进近似格式可以避免这一问题。

我们的算法具有牛顿迭代所特有的二次收敛性质，即它计算 $\mathcal{G}$ 余数 $\mathfrak{m}^2$、 $\mathcal{G}$ 余数 $\mathfrak{m}^4$，……；因此，不失一般性，我们假设 $K = 2^\kappa$ 是 2 的幂。算法的成本用以下两类量表示：

- 环 $\mathbb{A}/\mathfrak{m}^{2^i}$ 中的运算次数（我们将在引言末尾更详细地讨论我们的计算模型）

- 将多项式 f_i 的系数模 $\mathfrak{m}^{2^i}$ 约简的成本：我们假设，对于 $i \geq 0$，每个此类系数都可以在时间 T_{2^i} 内模 $\mathfrak{m}^{2^i}$ 约简（对于 $\mathbb{A} = \mathbb{Z}$，该时间将取决于这些系数的位大小；在 $\mathbb{A} = \mathbb{K}[t_1, \ldots, t_m]$ 上，该时间将取决于它们的次数以及参数的数量 m）。

全文中， $O\tilde{}$ 记号表示我们省略了多对数因子，而 ω 是线性代数的一个可行指数。

定理 1.4. 设 $f_1, \ldots, f_t$ 在 $\mathbb{A}[x, y]$ 中的次数至多为 d，其中 $\mathbb{A}$ 是一个整环，并生成 $\mathbb{K}[x, y]$ 中的理想 J， $\mathbb{K}$ 是 $\mathbb{A}$ 的分式域。设 I 是 J 的若干零维主分量的交，其最小约化 Gr ¨obner 基为 $\mathcal{G}$，对应于由 $y \succ x$ 诱导的字典序。

进一步，令 $\boldsymbol{E} = (y^{n_0}, x^{m_1}y^{n_1}, \ldots, x^{m_{s-1}}y^{n_{s-1}}, x^{m_s})$ 为 $\mathcal{G}$ 的首项，并令 $\delta = \dim_{\mathbb{K}} \mathbb{K}[x, y]/I$。

令 $\mathfrak{m} \subseteq \mathbb{A}$ 为 $\mathcal{G}$ 的一个良好极大理想。对于形如 $K = 2^k$ 的 K，给定 $\mathcal{G}$ rem $\mathfrak{m}$，可以找到 $\mathcal{G}$ rem $\mathfrak{m}^K$，其代价如下：

在 $\mathbb{A}/\mathfrak{m}^{2^i}$ 中进行 • $O\tilde{}(s^2 n_0 m_s + t\delta(d^2 + dm_s + s\delta + \delta^{\omega-1}))$ 次运算，其中 $i = 1, \ldots, k$；

进行 • $td^2 T_{2^i}$ 步系数约化，其中 $i = 1, \ldots, k$。

注 1.5。当 I 是 J 的 $\langle x, y\rangle$-主分量时，可以进一步改进运行时间，得到

在 $\mathbb{A}/\mathfrak{m}^{2^i}$ 中进行 • $O\tilde{}(s^2 n_0 m_s + t\delta^2(m_s + \delta^{\omega-2}))$ 次运算，其中 $i = 1, \ldots, k$；

进行 • $t\delta m_s T_{2^i}$ 步系数约化，其中 $i = 1, \ldots, k$。

由于 $m_s \leq \delta$，这些特别是 $O\tilde{}(s^2 n_0 m_s + t\delta^3) \subset O\tilde{}((s+t)\delta^3)$，分别为 $t\delta^2 T_{2^i}$。对于后者，我们也有定理中给出的界 $td^2 T_{2^i}$，但这里我们更倾向于仅用重数 δ 来表示代价。

本文关注理想 I 不是根理想的情形（也就是说，某些点 $p \in V(I)$ 是奇异点），旨在计算这些点处的局部结构。如果唯一的目标是求出 $V(I$，那么我们的方法就显得过于复杂：文献 [9, 12] 中的算法使用牛顿迭代，以高效的方式计算解的集合论描述。

例 1.6。一个极端情形是 $t = 2$ 和 f_1, f_2 “一般”，也就是说，它们在 $\mathbb{K}[x, y]$ 中定义了一个根理想，其 d^2 个解处于一般位置。在这种情况下，若取 $I = J$，则有 $s = 1, m_s = \delta = d^2$ 和 $n_0 = 1$。于是，定理第一项中的复杂度变为模每个 $\mathfrak{m}^{2^i}$进行 $O\tilde{}(d^5)$ 次运算。相比之下，文献 *[9]* 报告了类似任务的次立方成本$O\tilde{}(d^{(\omega+3)/2})$。

显然，对于这些一般情形，我们的算法与最先进水平相比并不占优。对于 例 1.6 中的情形，[9] 中的一些技巧也可用于我们的情形，但最多只能将运行时间降至 $O\tilde{}(d^{2+(\omega+3)/2})$ 次运算，$\mathbb{A}/\mathfrak{m}^{2^i}$，仍然存在二次开销。这是由于这些论文应用牛顿迭代的方式不同：在我们的情形中，我们将问题线性化到维数 d^2(或者一般而言，δ)，因而使用相应大小的矩阵；而 [9] 使用大小为 2 的矩阵（尽管其项为多项式）。

在存在重数交的情况下，定理 1.4 的结果值得关注；在这种情况下，[9] 等方法并不适用。此类情况下，[12] 中的算法无法解决我们的问题，因为它计算的不是 ¨ 的格罗布纳基，而是其根基。 I

请注意，要从我们的结果推导出一个完整的算法，还需要其他要素：对坏理想数量的定量界 $\mathfrak{m}$（例如，当 $\mathbb{A} = \mathbb{Z}$ 或 $\mathbb{A} = \mathbb{K}[t_1, \ldots, t_m]$时），计算起始点 $\mathcal{G}_{\mathfrak{m}}$ 的成本分析，以及关于充分精度的界 K，以便我们能够从其近似 $\mathcal{G}$ 余数 $\mathfrak{m}^K$中恢复 $\mathcal{G}$。为避免本文篇幅过长，我们将在另一篇文章中讨论这些问题

稿件。

现在我们回顾关于双变量系统以及用于 Gr¨obner 基的牛顿迭代的先前研究。正如我们将看到的，在一般位置下针对“简单”解（输入方程的雅可比矩阵满秩）的牛顿迭代算法，与能够处理任意情形的算法之间存在显著差异。

用于非退化解的牛顿迭代。继早期讨论之后，[18],p-进技术由 Trinks 于 20 世纪 80 年代引入，用于计算 Gr¨obner 基 [19]。该文聚焦于生成元属于 $\mathbb{Z}[x_1, \ldots, x_n]$、处于形状引理位置的零维根理想；也就是说，其字典序 Gr¨obner 基的形式为 $x_1 - G_1(x_n) \ldots, x_{n-1} - G_{n-1}(x_n)$ $G_n(x_n)$，其中采用字典序 $x_1 \succ \cdots \succ x_n$。在这一假设下，给定一个“幸运”素数 p，便可应用牛顿迭代的符号形式，将 $(G_1, \ldots, G_n)$ rem p 提升为 $(G_1, \ldots, G_n)$ rem p^K，其中 $K \geq 0$ 为任意整数。几何分辨算法 [20, 21, 22, 23]中也使用了类似技术；随后，[24] 将这种符号形式的牛顿迭代扩展到三角集。此处的三角集是指特定的字典序 Gr¨obner 基 $(G_1, \ldots, G_n)$，其首项分别具有 $x_1^{e_1}, \ldots, x_n^{e_n}$ 的形式，其中 $e_1, \ldots, e_n$ 为正整数。[9], 对本文所关注的情形 $n = 2$ 中的这些技术进行了详细研究，重点考察了提升过程的复杂度。

在计算上，这些算法相当直接：它们主要执行大小为 n 的矩阵乘法，其中的元素是系数属于 $\mathbb{Z}/p^K\mathbb{Z}$（或更一般地属于 $\mathbb{A}/\mathfrak{m}^K$）的多项式。这些方法也具有与其数值对应方法相同的二次收敛性（一次迭代中，精度从 p^K 提升至 p^{2K}，即翻倍），但它们都无法直接处理具有重数的解。

一般输入的提升算法 [25] 提出了一种能够处理任意输入的算法：给定关于 f_1 的 Gröbner 基 $\mathcal{G}$，其中 ¨obner 被模“幸运”素数 p 化简，它可以恢复同一系统模 p^K 下的 Gröbner 基，其中 ¨obner 保持不变，适用于任意 $K \geq 0$。该算法不对 $V(\langle f_1, \ldots, f_t \rangle)$ 的维数或方程的雅可比矩阵的秩作任何假设。与上述算法相比，其计算更复杂，因为它不仅要提升 Gröbner 基 $\mathcal{G}$ 本身，还要提升 $f_1, \ldots, f_t$ 除以 $\mathcal{G}$ 时所得的所有商，以及 $\mathcal{G}$ 的 S-多项式。

在后续工作中，[26] 讨论了幸运素数的选择；对于齐次输入或分次序，[27] 给出了停止提升的有效判据，并借鉴 Pauer 的思想简化了提升算法本身（不再需要 S-多项式）。

据我们所知，这里提到的算法仅执行线性提升，即从模 p^K 下的近似提升到精度 p^{K+1}；我们尚不清楚是否可能实现二次收敛。尚未进行成本分析。

去膨胀 Ojika、Watanabe 和 Mitsui 在数值计算背景下提出了去膨胀的概念 [28], ，以便即使对于重根，也能恢复牛顿迭代的二次收敛。其核心思想是用另一组方程替换给定的方程组，使其在我们感兴趣的根处具有

重数为一，并且可能引入新的变量。如今已有许多文献讨论这种方法，例如参见 [29, 30, 31, 32, 33, 34, 35, 36]。

我们尤其会用到文献 [37]中的一个想法。在该文献中，Hauenstein、Mourrain 和 Szanto 为一个 n元方程组 $f_1, \ldots, f_t$设计了一个缺陷化算子，该算子二次收敛到一个增广根（ξ, ν）；其中，ν 是一个向量，它通过点 $\xi \in V$处局部代数中乘法矩阵的系数，刻画点 $\xi \in V(\langle f_1, \ldots, f_t\rangle)$处的局部结构。若已知 ξ，这便给出了一个二次收敛的算子，特别适用于计算结构常数。

我们的贡献 我们提出的提升算法比文献[25, 27] 中的更简单（无需考虑 $f_1, \ldots, f_t$ 被 $\mathcal{G}$除法时产生的多项式商），但目前仅适用于两个变量的字典序。

第一步是找出一族自由参数，用以描述具有给定首项的 Gr¨obner 基（这些 Gr¨obner 基构成一个 Gr ¨obner 胞腔）。Gr¨obner 基中出现的系数并不构成这样一族参数，因为它们之间存在非平凡关系。然而，对于两个变量的字典序，Conca 和 Valla 明确构造了一个一一参数化：通过描述自关系模的典范生成元，将给定的 Gr¨obner 胞腔参数化为仿射空间 [38], 。我们的牛顿迭代计算对应于 $\mathcal{G}_{\mathfrak{m}}$ 的参数，并将其提升模 $\mathfrak{m}^K$。

这是通过调整[37]的方法实现的：关于未知格罗布纳基 $\mathcal{G}$取模后，f_1、……、f_t 的标准形的系数是格罗布纳胞腔参数的多项式；我们证明，这些多项式有一个（不一定唯一的）解，即对应于 $\mathcal{G}$的参数，并且它们的雅可比矩阵在此解处满秩。随后，我们可以对这些多项式应用牛顿迭代，并利用 $\mathcal{G}_{\mathfrak{m}}$ 得到它们模$\mathfrak{m}$ 意义下的解，作为起始点。

从计算角度看，核心操作仅仅是关于字典序格罗布纳基取模的约化。尽管对于关于单个多项式取模，或关于两个初始项分别为 y^n 和 x^m的多项式取模，我们已有准线性代价的算法，但我们并不了解适用于任意字典序基的具体结果。本文的另一项贡献是一种约化算法，其中我们将 van der Hoeven 和 Larrieu 为某些加权序开发的技术 [39]调整后用于我们的目的。

指南 在第 2 节中，我们讨论 初始段 在 $\mathbb{N}^2$中的应用；它们使我们能够描述模 Gr¨öbner 基约化后的多项式。我们尤其给出了两个此类多项式相乘的算法。

在第 3 节中，我们回顾了关于双变量字典序 Gr¨öbner 基结构的已知结果：Lazard 定理 [40], 以及 Conca 和 Valla 对 Gr¨öbner 胞腔的描述；第 4 节随后介绍我们用于模字典序 Gr¨öbner 基约化的算法。

在第 5 节和第 6节中，我们给出计算与 Gr¨öbner 胞中一组参数对应的 Gr¨öbner 基的算法，反之亦然。最后，我们在第 7节描述 Gr¨öbner 胞参数的牛顿迭代，证明定理 1.4。

计算模型 在本文中，算法的代价均以基环或基域中的运算次数来衡量。

我们首先主要统计 $\mathbb{Z}$-代数运算。对于一个输入和输出均在（含幺）环 $\mathbb{A}$ 中的算法，这些运算是对输入、先前计算出的量以及取自典范映射 $\mathbb{Z} \to \mathbb{A}$ 的像的常数进行的加法和乘法（例如，当 $\mathbb{A}$ 的特征为零时，常数可以是整数）；我们将其简称为“$(+, \times)$ 运算”。如果算法只执行这类运算，那么它的输出就在由其输入生成的 $\mathbb{A}$ 子环中。

重要的例子包括在 $\mathbb{A}[x]$ 中进行加法、乘法和欧几里得除法（除以首一除数）；这些运算都可以在任意基环 $\mathbb{A}$ 上用软线性数量的（$+$, $\times$）运算在 $\mathbb{A}$ 中完成。背景知识请参阅 [41] 中的第 8 章和第 9 章。

我们偶尔还会用到其他运算，例如可逆性测试和求逆（用于求解线性方程组）。最后，若 $\mathfrak{m}$ 是环 $\mathbb{A}$ 中的一个理想，给定 $a \in \mathbb{A}/\mathfrak{m}$，我们假设可以用一次 $\mathbb{A}$ 运算找到 $A \in \mathbb{A}$，使得 A 的余数为 $\mathfrak{m} = a$。

记号 本文中将始终使用以下记号。在下列各项中，$\mathbb{A}$是任意环。

- 对于 $d \geq 1$，我们令 $\mathbb{A}[x]_{<d}$ 为由 $\mathbb{A}[x]$ 中次数小于 d的所有多项式构成的自由 $\mathbb{A}$-模。
- 对于 f、$g \in \mathbb{A}[x]$，若 f 为首一多项式，则我们分别将 f rem g 和 f div g 定义为用 g除 f 所得欧几里得除法中的余式和商。
- 对于 $f \in \mathbb{A}[x, y]$，$\deg(f, x)$ 和 $\deg(f, y)$ 分别表示其关于 x 和 y的偏次数。
- 对于 $f \in \mathbb{A}[x, y]$ 和 $i \geq 0$，f中 y^i 的多项式系数指表达式 $f = \sum_{i=0}^{d} f_i y^i$中的系数 f_i，其中 f_0、……、$f_d \in \mathbb{A}[x]$。在伪代码中，我们用 $\textsc{PolynomialCoefficient}(f, y^i) \in \mathbb{A}[x]$ 表示这个多项式系数。
- 若 $f \in \mathbb{A}[x, y]$ 关于 y的次数为 d，则当 y^d 的多项式系数为 1 时，我们称 f 关于 y 是首一的（这个定义以及前一个定义也适用于关于 x 的系数，但本文不会用到）。
- 如果 T 是 $\mathbb{N}^2$ 的子集，我们将 $\mathbb{A}[x, y]_{\mathsf{T}}$ 记作由 T 支撑的 $\mathbb{A}$-模多项式，即所有形如 $\sum_{(u,v)\in \mathsf{T}} a_{u,v} x^u y^v$ 的多项式，其中只有有限多个系数 $a_{u,v}$ 非零。

我们不需要定义环上的格罗布纳基。具体来说，对于二元多项式的约化，我们只考虑域上的情形：若 $\mathcal{G}$ 是 $\mathbb{K}[x, y]$ 中的格罗布纳基，其中 $\mathbb{K}$ 是一个域，且 $\mathbb{K}[x, y]$ 配备了一个单项式序，则 f rem $\mathcal{G}$ 表示通过 $\mathcal{G}$ 约化 f 所得的余式。

2 $\mathbb{N}^2$中的初始段

本节首先介绍有关称为初始段的 $\mathbb{N}^2$ 子集的术语和基本构造。第二部分将给出用于计算支撑集包含在此类初始段中的多项式乘积的算法。

2.1 基本定义

初始段 若集合 T $\subset \mathbb{N}^2$ 满足：对于所有属于 T 的 (m, n)，任何满足 $m' \leq m$ 且 $n' \leq n$ 的对 (m', n') 也属于 T，则称 T 为一个初始段。

设 T 是 $\mathbb{N}^2$中的一个初始段，令 $\mathbb{K}$ 为一个域，x、y 为 $\mathbb{K}$上的变量。支撑集为 $\mathbb{N}^2 -$ T 的 $\mathbb{K}[x, y]$ 中的元素构成一个单项式理想 $I \subset \mathbb{K}[x, y]$。反过来，$\mathbb{N}^2$ 中的任意初始段 T 都可以由单项式理想 I以这种方式得到，即作为不属于 I的单项式的指数集合。若 T 是有限的，我们将 I 的极小单项式生成元写作

$$\boldsymbol{E} = (y^{n_0}, x^{m_1}y^{n_1}, \ldots, x^{m_{s-1}}y^{n_{s-1}}, x^{m_s})$$

其中 m_i递增，n_i递减，并令 $m_0 = n_s = 0$。我们称 n_0 为 T 的高度，称 m_s 为其宽度。若 T 由 I 决定，或等价地由 $\boldsymbol{E}$决定，则称 T 是由它们确定的。

对于 $i = 1$，...，s，我们设 $d_i = m_i - m_{i-1}$，使得 $m_i = d_1 + \cdots + d_i$。那么，δT 的基数可写为$\sum_{i=1}^{s} d_i n_{i-1}$；$\delta$ 也称为次数：$\boldsymbol{E}$。类似地，对于 $i = 1$，...，s，我们写作 $e_i = n_{i-1} - n_i$。图1对这些定义进行了说明，其中 $\boldsymbol{E}$ 中的单项式是式（¨Gröbner 基1）的首项。

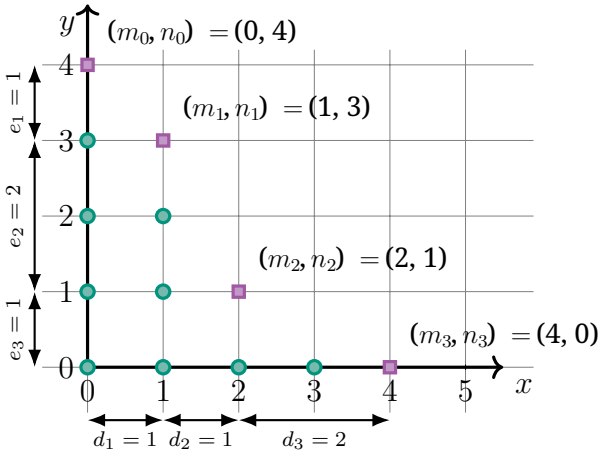

图1：初始段 T（绿色）和单项式 $\boldsymbol{E} =$ （y^4、xy^3、x^2y、x^4）（紫色），其中 $s = 3$ 和 $\delta = 9$。

本文的成本分析将特别使用参数 s 和 δ。如有需要，可以使用以下针对 s 的显式上界来简化此类表达式。

引理 2.1. 整数 s 属于 $O(\sqrt{\delta})$，并且在某些情形下，此下界是紧的。

证明. 从等式 $\delta = \sum_{i=1}^{s} d_i n_{i-1}$出发，可推出 $\delta \geq \sum_{i=1}^{s} n_{i-1}$。由于 $n_s = 0$和 $n_{i-1} > n_i$，由归纳法可得，对于所有 i，都有 $n_i \geq s - i$。这意味着 $\delta \geq s(s-1)/2$，因此 s属于 $O(\sqrt{\delta})$。对于下界，任取整数 d，我们可以取 $\boldsymbol{E} = (x^i y^{d-i},\ i = 0, \ldots, d)$，此时 $s = d$ 且 $\delta = d(d+1)/2$。 □

初始段的平移 我们偶尔会用到以下构造。设 T 是 $\mathbb{N}^2$ 中的一个有限初始段，并假设 T 由单项式理想 I 确定，其极小单项式生成元如上所示为 $\boldsymbol{E}$。对于 $i = 0, \ldots, s$，我们令 $\mathsf{T}_{\leftarrow i}$ 表示由冒号理想 $I : x^{m_i}$ 确定的初始段，其单项式生成元为

$$\boldsymbol{E}_{\leftarrow i} = (y^{n_i}, x^{m_{i+1}-m_i} y^{n_{i+1}}, \ldots, x^{m_{s-1}-m_i} y^{n_{s-1}}, x^{m_s - m_i}).$$

集合 $\mathsf{T}_{\leftarrow i}$ 的高度为 n_i，宽度为 $m_s - m_i$；其基数记为 δ_i，且等于 $\sum_{j=i+1}^{s} d_j n_{j-1}$。我们称 $\mathsf{T}_{\leftarrow i}$ 为 T 的第 i 个平移。

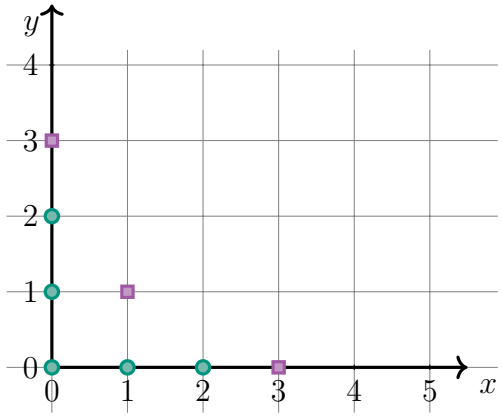

图 2：第一次平移 $\mathsf{T}_{\leftarrow 1}$ 的 T，取自图1。

初始段的外壳 设 T是 $\mathbb{N}^2$ 中的有限初始段。本段中，我们定义它的壳T′，它是另一个初始段，构成T 的外近似，且生成元较少。以下定义和引理出自 [42, **A.2**]。

和之前一样，我们令

$$\boldsymbol{E} = (y^{n_0}, x^{m_1} y^{n_1}, \ldots, x^{m_{s-1}} y^{n_{s-1}}, x^{m_s})$$

为与 T 关联的极小单项式生成集。通过引入指标 $i_\sigma < i_{\sigma-1} < \cdots < i_0$ 来定义 T′，其定义如下。令 $i_0 = s$。令 $i_1 \geq 0$ 为小于 i_0 且满足 $m_{i_1} < m_{i_0}/2$ 的最大指标，并重复这一过程来定义序列 $i_\sigma = 0 < i_{\sigma-1} < \cdots < i_0 = s$。然后我们可以考虑这些单项式

$$\boldsymbol{E}' = (y^{n_{i_\sigma}}, x^{m_{i_{\sigma-1}}} y^{n_{i_{\sigma-1}}}, \ldots, x^{m_{i_0}}) = (y^{n_0}, x^{m_{i_{\sigma-1}}} y^{n_{i_{\sigma-1}}}, \ldots, x^{m_s}),$$

并令 T′ 为由 $\boldsymbol{E}'$ 确定的初始段。

引理 2.2。初始段 T′ 包含 T，其基数至多为 2δ，且 σ 属于$O(\log(\delta))$。

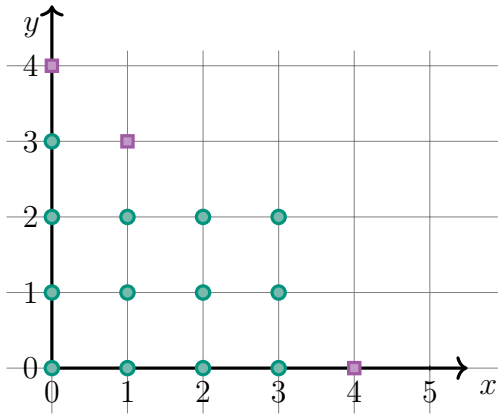

图 3： T 的壳，见图1。

在我们的伪代码中，我们用 T′ ← Shell(T) 表示 T′ 是 T 的壳。该算法 Shell 不使用任何基域或基环运算，只进行指标操作（特别是，它不会出现在我们的成本分析中）。

2.2 结构化多项式乘法

现在我们证明两个关于 $\mathbb{A}[x,y]$ 中多项式乘法的命题，其中的环 $\mathbb{A}$ 任意。这些命题将作为若干算法运行时间分析的基石。我们在下文的所有命题中都指出，本节的算法只使用 $\mathbb{A}$ 中的加法和乘法，后文将需要这一性质。下文中，给定 $\mathbb{N}^2$ 中的两个集合 S、T，S + T 表示它们的闵可夫斯基和。

主要的前提是以下事实：若 $S \subset \mathbb{N}^2$ 是一个矩形，给定 A 和 B 中的元素，我们可以使用 $O\tilde{}(|S|)$ 次运算（+， ×）在 $\mathbb{A}$ 中计算 $AB \in \mathbb{A}[x,y]_{S+S}$：若 S 包含原点，则使用克罗内克代换将问题化为在 $\mathbb{A}[x]$ 中的乘法，见 [41, 推论 8.28]；一般情况下，我们通过从 A 和 B 中分别提出 $x^u y^v$，将问题化为 S 包含原点的情形，其中（u， v）是 S 的唯一最小元素。

话虽如此，我们在此强调的第一个结果给出了计算乘积AB的代价，其中 A 和 B 支撑在同一个初始段 T 上。注意， AB 支撑在 T +T 上；若 T 的高度为 n、宽度为 m，则 T +T 的基数为 $\Theta(nm)$。事实上，这个集合包含基数为 nm的矩形 $\{0,\dots,m-1\}\times\{0,\dots,n-1\}$，并包含于基数小于 $4nm$的矩形 $\{0,\dots,2m-2\}\times\{0,\dots,2n-2\}$ 中，因此 $|T+T| \in \Theta(nm)$。这与 T 本身的基数形成对比，后者的取值范围可以是$n+m$ 到 nm之间的任意值。

命题 2.3。设 $\mathsf{T} \subset \mathbb{N}^2$ 是一个有限初始段，其高度为 n 、宽度为 m。给定A和 B，且它们属于 $\mathbb{A}[x,y]_{\mathsf{T}}$，可以使用 $O^{\sim}(|\mathsf{T}+\mathsf{T}|) = O^{\sim}(nm)$ 次运算（$+$， $\times$）在 $\mathbb{A}$中计算 AB。

证明。令 S 为矩形 $\{0,\ldots,m-1\}\times\{0,\ldots,n-1\}$，使得 S 包含 T。那么，$A$ 和 B 属于 $\mathbb{A}[x,y]_{\mathsf{S}}$，因此可以使用 $O^{\sim}(|\mathsf{S}+\mathsf{S}|) = O^{\sim}(nm)$ 次运算（$+$， $\times$），通过上文所述的克罗内克代换在 $\mathbb{A}$ 中将它们相乘，运行时间同样为$O^{\sim}(|\mathsf{T}+\mathsf{T}|)$。 □

我们的第二个命题给出了一个计算 $AB \in \mathbb{A}[x,y]$ 的算法，其中 A 的支撑集包含在一个含有原点的矩形内，而 B 的支撑集包含在一个初始段内。

命题 2.4。考虑一个矩形 $\mathsf{S} \subset \mathbb{N}^2$ 和一个有限初始段 $\mathsf{T} \subset \mathbb{N}^2$。给定 A（属于 $\mathbb{A}[x,y]_{\mathsf{S}}$）和 B（属于 $\mathbb{A}[x,y]_{\mathsf{T}}$），可以在 $\mathbb{A}$ 中使用 $O^{\sim}(|\mathsf{S}+\mathsf{T}|)$ 次运算（$+$， $\times$）计算 AB。

不失一般性，我们假设 S 包含原点 (0, 0)；否则，如上所述，将单项式 $x^u y^v$ 从 A中因式分解，其中 (u,v) 是 S 中的极小元素。因此，我们可以假设 S 是矩形 $\{0,\ldots,\ell-1\}\times\{0,\ldots,h-1\}$，其中 ℓ、$h \geq 1$ 为整数，特别是 $|\mathsf{S}| = \ell h$；并且 T 是基数为 $|\mathsf{T}| = \delta$ 的初始段，其高度为 n，宽度为 m。

若 $\mathbb{A}$ 是特征零域，则该结果可直接由 [43]的稀疏求值与插值算法得出。更一般地，若 $\mathbb{A}$ 是基数至少为 $\max(\ell+m, h+n)-1$ 的域，利用 [44] 中的算法也可得到这一结果。下述算法在不对 $\mathbb{A}$作任何假设的情况下，达到相同的渐近运行时间。该证明比前一个命题的证明稍复杂，将占据本节余下部分。

T 为矩形时的算法 假设首先 $\mathsf{T} = \{0,\ldots,m-1\}\times\{0,\ldots,n-1\}$，因此 $\delta = nm$；则 $\mathsf{S}+\mathsf{T}$ 的基数为 $(\ell+m-1)(h+n-1)$。

取 A属于 $\mathbb{A}[x,y]_{\mathsf{S}}$，且 B属于 $\mathbb{A}[x,y]_{\mathsf{T}}$。那么， A和 B都属于 $\mathbb{A}[x,y]_{\mathsf{S}+\mathsf{T}}$。由于 $\mathsf{S}+\mathsf{T}$ 是一个矩形，我们在本节开头看到，利用克罗内克代换，可以通过 $O^{\sim}(|\mathsf{S}+\mathsf{T}|) = O^{\sim}((\ell+m-1)(h+n-1))$ 次运算（$+$，$\times$）在 $\mathbb{A}$ 中计算它们的乘积。在下面的主算法中，这记作 KroneckerMultiply(A,B)。

第一个通用算法 现在假设 T 是一个任意的初始段，并且它由以下单项式确定

$$\boldsymbol{E} = (y^{n_0}, x^{m_1}y^{n_1}, \ldots, x^{m_{s-1}}y^{n_{s-1}}, x^{m_s}),$$

随着 m_i 递增、 n_i 递减，以及 $m_0 = n_s = 0$；注意，我们还有 $n_0 = n$ 和 $m_s = m$。与之前一样，对于 $i = 1$、……、 s，我们令 $d_i = m_i - m_{i-1}$，从而得到 $m_i = d_1 + \cdots + d_i$。

输入 $B \in \mathbb{A}[x,y]_{\mathsf{T}}$ 随后可写为 $B = \sum_{0\leq i<s} B_i x^{m_i}$，其中 B_i的支撑集位于 T $_i = \{0,\ldots,d_{i+1}-1\}\times\{0,\ldots,n_i-1\}$。因此，要计算 AB，其中 A属于 $\mathbb{A}[x,y]_{\mathsf{S}}$，我们只需计算所有 AB_i 并将结果相加。

算法 2.1 MULTIPLYNAIVE(*A*,S *B*, T)

INPUT: *A* 属于 $\mathbb{A}[x,y]_{\mathsf{S}}$， *B* 属于 $\mathbb{A}[x,y]_{\mathsf{T}}$OUTPUT: *AB* 属于 $\mathbb{A}[x,y]_{\mathsf{S}+\mathsf{T}}$1: 令 $B = B_0 + B_1 x^{m_1} + \cdots + B_{s-1} x^{m_{s-1}}$，其中对所有 *i*均有 $B_i \in \mathbb{A}[x,y]_{\{0,\ldots,d_{i+1}-1\}\times\{0,\ldots,n_i-1\}}$ 2: 对于 $i = 0, \ldots, s-1$，执行 $C_i \leftarrow$ KRONECKERMULTIPLY(A, B_i)3: 返回 $C_0 + C_1 x^{m_1} + \cdots + C_{s-1} x^{m_{s-1}}$

根据 前一段的结果，每个乘积 AB_i 都可以在

$$O\tilde{}((\ell + d_{i+1} - 1)(h + n_i - 1)) = O\tilde{}((\ell-1)(h-1) + (\ell-1)n_i + d_{i+1}(h-1) + d_{i+1}n_i)$$

次运算内计算出来，而将此乘积加到最终结果中的代价也符合相同的上界。利用对所有 i 都成立的不等式 $n_i \le n_0 = n$，以及 $d_1 + \cdots + d_s = m_s = m$ 和 $d_1 n_0 + \cdots + d_s n_{s-1} = \delta$（T 的基数），可得总成本为

$$O\tilde{}(s(\ell-1)(h-1) + s(\ell-1)n + m(h-1) + \delta).$$

另一方面，我们可以如下确定和 U $=$S $+$T 的基数。集合 U 是以下集合的不交并：

- $\mathsf{U}_1 = \{0, \ldots, \ell-2\} \times \{0, \ldots, h-2\}$,
- $\mathsf{U}_2 = (0, h-1) + \{0, \ldots, \ell-2\} \times \{0, \ldots, n-1\}$
- $\mathsf{U}_3 = (\ell-1, 0) + \{0, \ldots, m-1\} \times \{0, \ldots, h-2\}$
- $\mathsf{U}_4 = (\ell-1, h-1) + \mathsf{T}$.

这是通过取 S 中的 (i,j)、T 中的 (v,w)，并根据 $v - (\ell-1-i)$ 和 $w - (h-1-j)$ 的符号进行讨论而得到的。因此，我们得到

$$|\mathsf{S}+\mathsf{T}| = (\ell-1)(h-1) + (\ell-1)n + m(h-1) + \delta.$$

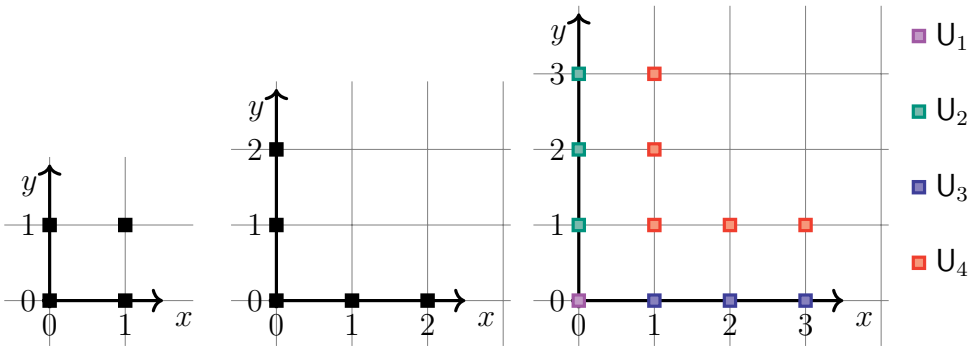

图 4: 集合 S、T 和 U $=$S $+$T，以及 $\ell = h = 2$ 和 $n = m = 3$。

主算法。 上面报告的运行时间不符合目标代价 $O^{\sim}(|\mathsf{S}+\mathsf{T}|)$，因为 s 可能很大。为规避这一问题，我们应用前一段中的算法，但将 T 替换为其壳 T′。我们知道（引理 2.2），T′的基数至多为 2δ，其宽度和高度与 T的相同，并且它由 $\sigma \in O(\log(s)) \subset O(\log(\delta))$ 个项生成。

算法 2.2 MULTIPLY(A,S B, T)

INPUT: A 令 $\mathbb{A}[x,y]_{\mathsf{S}}$, B 属于 $\mathbb{A}[x,y]_{\mathsf{T}}$
OUTPUT: AB 属于 $\mathbb{A}[x,y]_{\mathsf{S}+\mathsf{T}}$1: T′ ← SHELL
(T)2: 返回 MULTIPLYNAIVE(A, S B, T′)

前一段的算法仍然适用（因为 T 包含于 T′），其运行时间为 $O^{\sim}((\ell-1)(h-1)\log(\delta)+(\ell-1)n\log(\delta)+m(h-1)+\delta)$ 次运算（$+,\times$）在$\mathbb{A}$中。由于我们看到 $|\mathsf{S}+\mathsf{T}|=(\ell-1)(h-1)+(\ell-1)n+m(h-1)+\delta$，上面的表达式确实属于 $O^{\sim}(|\mathsf{S}+\mathsf{T}|)$。由此完成命题 2.4 的证明。

3 字典序 Gr¨obner 基

本节首先回顾 Lazard 结构定理 [40] 关于字典序 Gr¨obner 基的结果，其中 $\mathbb{K}[x,y]$是一个域 $\mathbb{K}$，随后介绍由 [38]给出的这类基的参数化。虽然讨论的核心不对所考虑的理想作任何假设，但我们也特别关注在原点处为主初等的理想，即 $\langle x,\ y\rangle$-主初等理想。

在下文中，我们均使用由$y \succ x$诱导的 $\mathbb{K}[x,y]$ 上的字典序单项式序 $\succ$ 。

3.1 结构定理

考虑一个零维理想 $I \subseteq \mathbb{K}[x,y]$，并设 $\mathcal{G}=(g_0,\ldots,g_s)$ 是其按降序排列的约化极小 Gr¨obner 基。进一步设

$$\boldsymbol{E}=(y^{n_0},x^{m_1}y^{n_1},\ldots,x^{m_{s-1}}y^{n_{s-1}},x^{m_s})$$

是初始理想 $in(I)$ 的按降序排列的极小约化基，因此 n_i 递减，而 m_i 递增；与前面一样，令 $m_0=n_s=0$。

由此可知，对于所有 i， g_i 的首项为 $x^{m_i}y^{n_i}$；特别地， g_0 在 y 中是首一的，首项为 y^{n_0}。

与第 2.1节一样，对于 $i=1$、.....、 s，我们令 $d_i=m_i-m_{i-1}$，因此 $m_i=d_1+\cdots+d_i$，并且 $e_i=n_{i-1}-n_i$。

Lazard 在[40, 定理 1] 证明了存在多项式 $D_1,\ldots,D_s$ 属于 $\mathbb{K}[x]$，它们在 x 中均为首一多项式，且相应次数为 $d_1,\ldots,d_s$，使得对于 $i=0,\ldots,s$， g_i 可以写成 M_iG_i，其中 $M_i=D_1\cdots D_i\in\mathbb{K}[x]$ 并且 $G_i\in\mathbb{K}[x,y]$ 是 n_i 中次数为 y（对于 $i=0$,

记为 $D_0 = 1$）。特别地，对于 $i = s$，这给出 $g_s = M_s = D_1 \cdots D_s$ 以及 $G_s = 1$。此外，对于 $i = 0, \ldots, s-1$，我们有隶属关系

$$G_i \in \langle G_{i+1},\ D_{i+2}G_{i+2},\ \ldots,\ D_{i+2}\cdots D_s \rangle = \left\langle \frac{g_{i+1}}{M_{i+1}}, \frac{g_{i+2}}{M_{i+1}}, \ldots, \frac{g_s}{M_{i+1}} \right\rangle, \tag{2}$$

其中，多项式 $G_{i+1}, D_{i+2}G_{i+2}, \ldots, D_{i+2}\cdots D_s$ 也构成一个零维 Gröbner 基。

如果 $\mathcal{G}$ 生成一个 $\langle x,\ y\rangle$-主理想，则有 $D_i = x^{m_i}$ 对所有 i，从而 $g_s = x^{m_s}$。此外，对所有 i，$G_i(0, y)$ 仅在 $y = 0$ 处为零，即 $G_i(0, y) = y^{n_i}$，参见 [40, 定理 2]。

就数据结构而言，表示 $\mathcal{G} = (g_0, \ldots, g_s)$ 需要 $O(s\delta)$ 个域元素，其中 δ 是 I 的次数。顺便指出，只存储多项式 $\boldsymbol{D} = (D_1, \ldots, D_s)$ 和 $\boldsymbol{G} = (G_0, \ldots, G_s)$ 就足够了。若 $\mathsf{T} \subset \mathbb{N}^2$ 是由 $\boldsymbol{E}$ 确定的初始段，则结构定理表明，对于 $i = 0, \ldots, s$，$G_i - y^{n_i}$ 的支撑集位于 T 的第 i 次平移 $\mathsf{T}_{\leftarrow i}$ 上。特别地，存储它需要 δ_i 个域元素，其中 $\delta_i = |\mathsf{T}_{\leftarrow i}|$，因此存储 $\boldsymbol{D}$ 和 $\boldsymbol{G}$ 总共只需稍少一些，即 $O(\sum_{i=0}^{s} \delta_i)$ 个域元素。

3.2 Conca 和 Valla 的参数化

在本小节中，我们假设元组 $\boldsymbol{E} = (y^{n_0}, x^{m_1}y^{n_1}, \ldots, x^{m_{s-1}}y^{n_{s-1}}, x^{m_s})$ 是固定的。沿用 [38], ，我们关注于描述 I 中初始理想由 $\boldsymbol{E}$ 生成的理想集合。我们将此集合称为 $\boldsymbol{E}$ 的 Gröbner 胞，并记作 $\mathcal{C}(\boldsymbol{E}) := \{I \mid in(I) = \langle \boldsymbol{E} \rangle\}$。我们还将提及它的一个子集，即 I 中初始理想由 $\boldsymbol{E}$ 生成且为 $\langle x, y\rangle$-primary 的理想集合；这称为 $\boldsymbol{E}$ 的局部 Gröbner 胞，记作 $\mathcal{C}_0(\boldsymbol{E})$。

用给定初始理想描述理想的思想，最早可追溯到 [45, 46, 47] 关于 $\mathbb{K}[\![x, y]\!]$ 中的理想以及 [48] 关于 $\mathbb{K}[x_1, \ldots, x_n]$ 的工作；随后又在 [49, 50, 51] 及其他几篇文献中得到发展。众所周知，这些 Gröbner 胞（也称为分层）具有相应的模空间，而这些模空间是仿射空间；但据我们所知，至今尚未给出一般性的显式描述。然而，在我们的情形中，Conca 和 Valla 在 [38] 中完整描述了字典序下二元理想的 Gröbner 胞和局部 Gröbner 胞（沿袭了 [52], 先前的工作，其中这些胞的维数已得到明确）。

例3.1。关于一个零维 Gröbner 单元的例子，取 $\boldsymbol{E} = (y^4,\ xy^3,\ x^2y,\ x^4)$如图 *1*所示，利用以下事实：$g_i = x^{m_i}G_i$ 以及 $G_i(0,\ y) = y^{n_i}$，我们可推得，整数环 $\mathbb{K}$上的一个理想的字典序 Gröbner 基 $\mathcal{C}_0(\boldsymbol{E})$ 必然具有以下形式，其中某些系数为 c_1，……，c_8 属于 $\mathbb{K}$：

$$\begin{aligned} g_1 &= y^4 + c_1xy^2 + c_2xy + c_3x^3 + c_4x^2 + c_5x \\ g_2 &= xy^3 + c_6x^3 + c_7x^2 \\ g_3 &= x^2y + c_8x^3 \\ g_4 &= x^4 \end{aligned}$$

不过，到目前为止，我们还没有考虑 (2)中的隶属等式，它对系数 c_i施加了关系。下面给出的 $\mathcal{C}(\boldsymbol{E})$ 和 $\mathcal{C}_0(\boldsymbol{E})$ 的参数化解决了这个问题。

回顾一下，我们将 $d_i = m_i - m_{i-1}$和 $e_i = n_{i-1} - n_i$记作，对于 $i = 1, \ldots, s$。给定 I属于$\mathcal{C}(\boldsymbol{E})$，Conca 和 Valla 证明了存在唯一的多项式 $(\sigma_{j,i})_{0\le i\le s-1, i\le j\le s}$属于 $\mathbb{K}[x, y]$，满足以下次数约束：

- 对所有 $i = 0, \ldots, s-1$ 以及 $j = i, \ldots, s$，$\deg(\sigma_{j,i}, x) < d_{i+1}$
- 对所有 $i = 0, \ldots, s-1$，$\sigma_{i,i}$属于 $\mathbb{K}[x]$，且对于 $j = i+1, \ldots, s$，$\deg(\sigma_{j,i}, y) < e_j$成立，

并且满足以下性质。定义多项式 $\mathcal{H} = (h_0, \ldots, h_s)$ 属于 $\mathbb{K}[x, y]$，其定义为

- $h_s = (x^{d_1} - \sigma_{0,0}) \cdots (x^{d_s} - \sigma_{s-1,s-1})$
- 对于 $i = 0, \ldots, s-1$,

$$x^{d_{i+1}} h_i - y^{e_{i+1}} h_{i+1} = \sigma_{i,i} h_i + \sigma_{i+1,i} h_{i+1} + \cdots + \sigma_{s,i} h_s; \tag{3}$$

那么，所有多项式 h_i都属于 I。由于上述关系表明，对于 $i = 0$，……，s，h_i的首项为 $x^{m_i} y^{n_i}$，$\mathcal{H} = (h_0$，……，$h_s)$ 是 ¨的一个极小 Gr Iobner 基。（注意，等式 (3) 给出了 h_i 与 h_{i+1}之间的协 syzygy 的标准形。）

反过来，对于任意满足上述次数约束的多项式 $\sigma_{j,\,i}$，由此得到的多项式 $\mathcal{H}$ 构成了环 $\mathcal{C}(\boldsymbol{E})$ 中理想 I 的一个极小 Gr¨obner 基。

我们简要介绍多项式 h_0，……，h_s的一些性质。首先，我们声称它们的 x-次数要么恰好为 m_s（对于 h_s），要么小于 m_s，对于 h_0，……，h_{s-1}。根据构造，这对 h_s 成立。对于其他指标，由递降归纳法将式 (3) 改写为

$$(x^{d_{i+1}} - \sigma_{i,i}) h_i = y^{e_{i+1}} h_{i+1} + \sigma_{i+1,i} h_{i+1} + \cdots + \sigma_{s,i} h_s, \tag{4}$$

其中右侧的所有项 $\sigma_{j,\,i} h_j$ 的 x 次数均小于 $d_{i+1} + m_s$。

接下来注意，对于 $i = 0$，…，s，$(x^{d_1} - \sigma_{0,\,0}) \cdots (x^{d_i} - \sigma_{i-1,\,i-1})$ 整除 h_i，因此也整除所有多项式 h_i，…，h_s；这由（4）通过递降归纳法得到（对于 $i = 0$，规定空乘积为 1）。由于 h_i 的首项为 $x^{m_i} y^{n_i} = x^{d_1 + \cdots + d_i} y^{n_i}$，我们推得 $(x^{d_1} - \sigma_{0,\,0}) \cdots (x^{d_i} - \sigma_{i-1,\,i-1})$ 正是 y^{n_i} 在 h_i 中的多项式系数。

于是，$\mathcal{G} = (g_0, \ldots, g_s)$ 是通过对 $\mathcal{H}$ 进行相互约化得到的约化 Gr¨obner 基。由于 $(x^{d_1} - \sigma_{0,\,0}) \cdots (x^{d_i} - \sigma_{i-1,\,i-1}) y^{n_i}$ 中没有任何项可以被 $h_0, \ldots, h_{i-1}$ 或 $h_{i+1}, \ldots, h_s$ 约化，因此可知 $(x^{d_1} - \sigma_{0,\,0}) \cdots (x^{d_i} - \sigma_{i-1,\,i-1})$ 也是 y^{n_i} 在 g_i 中的多项式系数。因此，Lazard 结构定理中出现的多项式 D_i 和 M_i 分别为 $D_i = x^{d_i} - \sigma_{i-1,\,i-1}$ 和 $M_i = (x^{d_1} - \sigma_{0,\,0}) \cdots (x^{d_i} - \sigma_{i-1,\,i-1})$。

总而言之，对于 Gr¨obner 单元 $\mathcal{C}(\boldsymbol{E})$，多项式 $(\sigma_{j,\ i})_{0\le i\le s-1,\ i\le j\le s}$ 中出现的系数总数 N 为

$$\begin{aligned} N &= \sum_{i=0}^{s-1}\left(\sum_{j=i+1}^{s} d_{i+1}e_j + d_{i+1}\right) \\ &= \sum_{i=0}^{s-1} d_{i+1}n_i + \sum_{i=0}^{s-1} d_{i+1} \\ &= \delta + m_s, \end{aligned}$$

其中 δ $\boldsymbol{E}$的次数。这些系数记作 $\lambda_1, \ldots, \lambda_N$，并称为 Gr ¨obner 参数；由此得到 $\Phi_{\boldsymbol{E}}$ 从 $\mathbb{K}^N$ 到 $\mathcal{C}(\boldsymbol{E})$ 的双射。

在定点 Gr¨obner 胞腔 $\mathcal{C}_0(\boldsymbol{E})$ 中的元素，可通过将某些 Gr¨obner 参数设为零得到，这对应于以下附加条件：

- 多项式 $\sigma_{0,0}, \ldots, \sigma_{s-1,s-1}$ 均为零（回忆一下，对于定点 Gr¨obner 胞腔，我们有 $D_i = x^{d_i}$ 和 $M_i = x^{m_i}$，对所有 i 均成立）
- $\sigma_{i+1,i}$ 可被 x 整除，其中 $i = 0, \ldots, s-1$。

$\sigma_{1,0}, \ldots, \sigma_{s,s-1}$ 中剩余系数的个数为

$$\begin{aligned} N_0 &= \sum_{i=0}^{s-1}\left(\sum_{j=i+1}^{s} d_{i+1}e_j - e_{i+1}\right) \\ &= \sum_{i=0}^{s-1} d_{i+1}n_i - \sum_{i=0}^{s-1} e_{i+1} \\ &= \delta - n_0, \end{aligned}$$

从而建立了 $\mathbb{K}^{N_0}$ 与 $\mathcal{C}_0(\boldsymbol{E})$ 之间的双射。回忆一下，在初级情形中，$\boldsymbol{E}$ 的次数δ 等于 $\mathcal{C}_0(\boldsymbol{E})$ 中各理想在原点处的重数。

例3.2. 我们来描述运行示例（ ¨中的零维 Gr $\boldsymbol{E}$ obner 单元（例 1.1）。其维数为 $N_0 = 9 - 4 = 5$，因此我们可以使用参数 $\lambda_1, \ldots, \lambda_5$，以及形如 $(\sigma_{i,j})$ 的多项式

$$\sigma_{0,0} = \sigma_{1,0} = 0, \quad \sigma_{2,0} = \lambda_1 y + \lambda_2, \quad \sigma_{3,0} = \lambda_3, \quad \sigma_{1,1} = n_{2,1} = 0, \quad \sigma_{3,1} = \lambda_4, \quad \sigma_{2,2} = 0, \quad \sigma_{3,2} = \lambda_5 x.$$

那么，$\mathcal{C}_0(\boldsymbol{E})$ 中的理想恰好是具有如下 Gr ¨obner 基的那些理想：

$$\begin{aligned} h_0 &= y^4 + \lambda_5 x y^3 + \lambda_1 x y^2 + (\lambda_1\lambda_5 + \lambda_4)x^2 y + \lambda_2 x y + \lambda_3 x^3 + \lambda_2\lambda_5 x^2 \\ h_1 &= x y^3 + \lambda_5 x^2 y^2 + \lambda_4 x^3 \\ h_2 &= x^2 y + \lambda_5 x^3 \\ h_3 &= x^4. \end{aligned}$$

正如预期的那样，这些并不是约化的 Gr ¨obner 基。约化后，我们得到以下多项式 $\mathcal{G}$:

$$\begin{aligned} g_0 &= y^4 + \lambda_1 xy^2 + \lambda_2 xy + (-\lambda_1\lambda_5^2 + \lambda_3 - 2\lambda_4\lambda_5)x^3 + \lambda_2\lambda_5 x^2 \\ g_1 &= xy^3 + \lambda_4 x^3 \\ g_2 &= x^2 y + \lambda_5 x^3 \\ g_3 &= x^4. \end{aligned} \tag{5}$$

4 模关于字典序 Gr¨obner 基的约化

与前文相同，设 $\mathcal{G} = (g_0, \ldots, g_s)$ 是 $\mathbb{K}[x, y]$ 中的字典序 Gr¨obner 基，其初始段为 T $\subset \mathbb{N}^2$。给定 f in $\mathbb{K}[x, y]$，我们关注如何计算余式$r = f$ rem $\mathcal{G} \in \mathbb{K}[x, y]_{\mathsf{T}}$；本文会多次用到这一计算，而它本身也是一个有趣的问题。值得注意的是，我们不了解此前是否有人研究过这一特定问题的复杂度。

我们首先将必要的背景作为一个平面几何问题来展开。这受到 [39], 的工作的启发，该工作针对的是某些特定的加权序（下文将进一步讨论）。接下来，我们介绍将多项式转换为所谓的混合基数表示并转换回来所用的算法；最后一个小节将给出约化算法本身。

4.1 一个铺砌问题

对于如上所述的 $\mathcal{G}$ 以及 f 属于 $\mathbb{K}[x, y]$ 的情形，余式 $r = f$ rem $\mathcal{G}$ 是唯一确定的，但关系式 $f = Q_0 g_0 + \cdots + Q_s g_s + r$ 中的商 Q_i 并非如此。约化算法将通过依次计算各个 Q_i 来得到 r 。因此，为了完整地明确该算法，我们需要使这些商的选取毫无歧义：每当一个单项式 $x^u y^v$ 可以被多个 Gr¨obner 基元素约化时，我们必须规定使用哪一个 g_i。所得算法的代价将实质性地取决于这些选择。

[39] 针对加权次数序，在模某些“良好”的 Gr¨obner 基约化的情形下，引入了一种二分方案。在本小节中，我们将其构造调整到我们的情形。

与之前一样，假设 $\mathcal{G}$ 的首项是这些单项式

$$\boldsymbol{E} = (y^{n_0}, x^{m_1} y^{n_1}, \ldots, x^{m_{s-1}} y^{n_{s-1}}, x^{m_s});$$

我们仍将 $d_i = m_i - m_{i-1}$ 和 $e_i = n_{i-1} - n_i$ 记作 $i = 1$、.....、 s。我们将对其应用主要约化算法的单项式集合是 $\{x^u y^v,\ 0 \le u < m_s,\ 0 \le v < n_0\}$，因此其基数为 $n_0 m_s$ （一般情形将化归为这一情形）。特别地，g_0 和 g_s 都不能约化这些单项式中的任何一个。

于是，我们可以将问题转化为平面上的铺砌问题。我们希望用矩形覆盖 S $= \{0, \ldots, m_s - 1\} \times \{0, \ldots, n_0 - 1\} -$ T，并满足以下约束：

- 我们使用 $s-1$ 两两不相交的矩形 R_1、……、R_{s-1}，使得 R_i 将标记使用 g_i约化的单项式集合
- 对所有 i，R_i 具有形式 $\{m_i, \ldots, m_{i+\ell_i} - 1\} \times \{n_i, \ldots, n_{i-h_i} - 1\}$，其中 ℓ_i、h_i 为正整数，且满足 $i + \ell_i \le s$ 和 $i - h_i \ge 0$
- 所有 R_i 的并集覆盖 S。

序列 $((\ell_1, h_1) \ldots, (\ell_{s-1}, h_{s-1}))$ 足以指定这样的铺砌。因此，我们的目标是使下列量最小：

$$c := n_0 \sum_{i=1}^{s-1} (m_{i+\ell_i} - m_i) + m_s \sum_{i=1}^{s-1} (n_{i-h_i} - n_i),$$

其中 $(m_{i+\ell_i} - m_i)$ 和 $(n_{i-h_i} - n_i)$ 分别是 R_i 的宽度和高度。事实证明，这个量决定了约化算法的代价；我们的目标是使 c 保持在 $\tilde{O}(n_0 m_s)$ 中，因为如前所述，$n_0 m_s$ 是我们要约化的多项式中单项式数量的上界。

下图展示了两种可能的铺砌，对应于引理 2.1 证明中已经出现的那族对象的 $d = 4$ 情形，其中 $\boldsymbol{E} = (y^d, xy^{d-1}, \ldots, x^d)$。对于这族对象，$n_0 = m_s = d$ 和 $n_0 m_s = d^2$；下方示例中的策略要么在 $\Theta(d^2)$ 中有 $\sum_{i=1}^{s-1}(m_{i+\ell_i} - m_i)$，要么有 $\sum_{i=1}^{s-1}(n_{i-h_i} - n_i)$，因此无论哪种情况，$c$ 都在 $\Theta(d^3) = \Theta((n_0 m_s)^{1.5})$ 中。

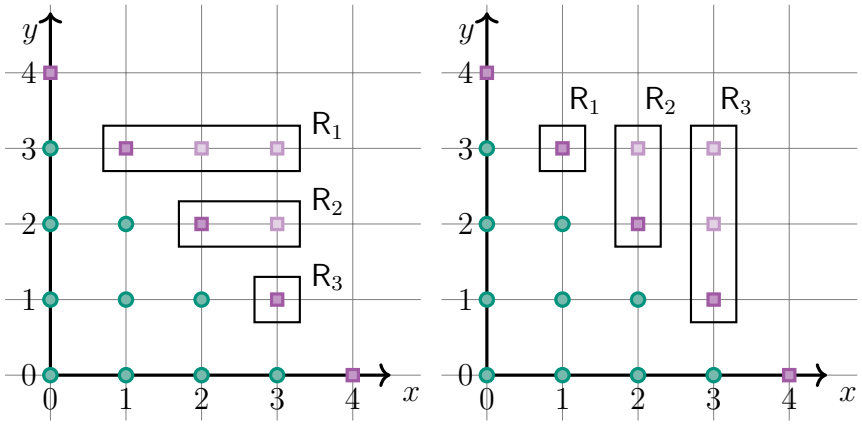

图 5：两种可能的铺砌，带有 $d = 4$。

对于这一族，下面给出一种更好的解法。

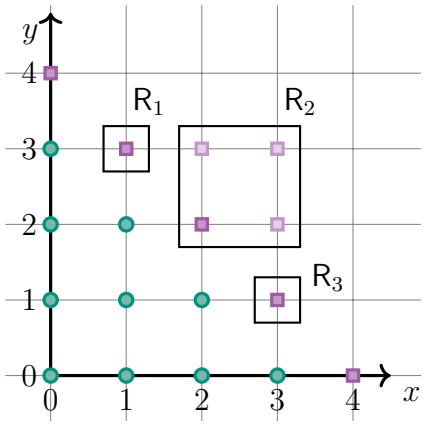

图6：一种改进的铺砌。

这一设计最初是为 [39], 类似于示例中那一类的族 $\boldsymbol{E}$ 提出的，其中阶梯宽度 d_i （几乎）为常数，且所有阶梯高度 e_i 都等于 1。下面给出的适用于任意输入的构造可直接由此推导而来。下文中，$\mathrm{val}_2(i)$ 表示正整数 i的 2 进赋值。

定义 4.1。对于 $i = 1, \ldots, s-1$，定义：

- $h_i = 2_{\mathrm{val}_2(i)}$
- $\ell_i = \min(h_i, s-i)$

As a 结果，矩形 R_i 是 $\{m_i, \ldots, m_{\min(i+h_i,s)} - 1\} \times \{n_i, \ldots, n_{i-h_i} - 1\}$.

命题4.2。对于任意 s 以及任意选择的 $m_1, \ldots, m_s$ 和 $n_0, \ldots, n_{s-1}$，矩形 $\mathsf{R}_1, \ldots, \mathsf{R}_{s-1}$两两不相交，覆盖 $\mathsf{S} = \{0, \ldots, m_s - 1\} \times \{0, \ldots, n_0 - 1\} - \mathsf{T}$，并且满足$i + \ell_i \le s$ 和 $i - h_i \ge 0$，对所有 i均成立。

证明。最后一个断言是定义的直接推论。我们通过约化到所有 d_i 和 e_i 都等于一的情形，来证明命题的其余部分。这个证明较为技术性，但没有特别的难点。

对于任意正整数 s，我们定义单项式 $\mathscr{E}_s = (x^i y^{s-i},\ 0 \le i \le s)$，以及由 $\mathscr{E}_s$ 和 $\mathscr{S}_s = \{0, \ldots, s-1\} \times \{0, \ldots, s-1\} - \mathscr{T}_s$ 确定的初始段 $\mathscr{T}_s$；注意， $\mathscr{T}_s$ 是由所有满足 $b < s - a$ 的非负整数对 (a, b) 构成的集合。最后，对于 $i = 1, \ldots, s-1$，我们定义矩形 $\mathscr{R}_{i,\ s} = \{i, \ldots, \min(i + h_i, s) - 1\} \times \{s - i, \ldots, s - i + h_i - 1\} \subset \mathscr{S}_s$。

我们从命题陈述中的 $m_1, \ldots, m_s$ 和 $n_0, \ldots, n_{s-1}$ 出发，并考虑 $\mathbb{N}^2$ 中相应的集合 T 和 S。取 S 中一点 (u, v)。由于 $u < m_s$，存在唯一的一对 (α, u') 使得 $u = m_\alpha + u'$，其中 $0 \le \alpha \le s-1$ 和 $0 \le u' < d_{\alpha+1}$。同理，由于 $v < n_0$，存在唯一的一对 (β, v') 使得 $v = n_\beta + v'$，其中 $1 \le \beta \le s$ 和 $0 \le v' < e_\beta$。我们断言，$(\alpha, s - \beta)$ 属于前一段中定义的集合 $\mathscr{S}_s$，并且对于 $i = 1, \ldots, s-1$，(u, v) 属于矩形 R_i，当且仅当 $(\alpha, s - \beta)$ 属于矩形 $\mathscr{R}_{i,\ s}$。

- 对于第一个断言，我们已经指出不等式 $0 \le \alpha \le s-1$ 和 $1 \le \beta \le s$，由此得到 $0 \le s-\beta \le s-1$，因此 $(\alpha, s-\beta)$ 属于正方形 $\{0,\dots,s-1\} \times \{0,\dots,s-1\}$。另一方面，我们有 $v \ge n_\alpha$（否则 (α,β) 就会属于 T），因此 $\beta \le \alpha$ 和 $s-\beta \ge s-\alpha$。这证明点 $(\alpha, s-\beta)$ 不属于 $\mathscr{T}_s$，所以总而言之，它属于 $\mathscr{S}_s$。

- 对于第二个断言，注意到由于 $u = m_\alpha + u'$，且 $0 \le u' < d_{\alpha+1}$，$m_i \le u < m_{\min(i+h_i,s)}$ 等价于 $i \le \alpha < \min(i+h_i, s)$。同理，不等式 $n_i \le v < n_{i-h_i}$ 等价于 $s-i \le s-\beta < s-i+h_i$。由此得证。

最后，现在只需证明对于所有 s，以下性质 $P(s)$ 成立：矩形 $\mathscr{R}_{1,s}, \dots, \mathscr{R}_{s-1,s}$ 两两不相交，且覆盖 $\mathscr{S}_s$。首先，我们用归纳法证明当 s 是 2 的幂且形式为 $s = 2^k$ 时，该性质成立，其中归纳变量为 $k \ge 1$。对于 $k = 1$（因此 $s = 2$），无需证明，因为 $\mathscr{S}_2 = \{1\} \times \{1\} = \mathscr{R}_{1,2}$。

假设对于 $s = 2^k$，性质 $P(s)$ 成立，现在我们证明它对于 $s' = 2s$ 也成立。对于 $\mathscr{S}$ 的一个子集 $\mathbb{N}^2$，我们用 $\mathscr{S} \cap \{x \le t\}$ 表示所有满足 $x \le t$ 的 (x,y)（属于 $\mathscr{S}$）组成的集合。集合 $\mathscr{S} \cap \{x \ge t\}$、$\mathscr{S} \cap \{x \le t, y \le t'\}$ 等也类似定义。

首先，我们注意到，对于任意的 2 的幂 $\sigma = 2^t$ 和 $i = 1$，……，$\sigma - 1$，都有 $i + h_i \le \sigma$，因此矩形 $\mathscr{R}_{i,\sigma}$ 就是 $\mathscr{R}_{i,\sigma} = \{i, \dots, i+h_i-1\} \times \{\sigma - i, \dots, \sigma - i + h_i - 1\}$。因此，矩形 $\mathscr{R}_{1,s'}$，……，$\mathscr{R}_{s-1,s'}$ 分别是 $\mathscr{R}_{1,s}$，……，$\mathscr{R}_{s-1,s}$ 沿 $(0,s)$ 平移得到的，所以根据归纳假设，它们两两不相交，覆盖 $\mathscr{S}_{s'} \cap \{x \le s-1\}$，并且不与 $\mathscr{S}_{s'} \cap \{x \ge s\}$ 相交（在图6中，我们有 $s = 2$、$s' = 4$，且只有一个这样的矩形，记作 R_1）。由于对于 $i = 1$，……，$s-1$都有 $h_i = \varepsilon_{i+s}$，我们还可以推出，矩形 $\mathscr{R}_{s+1,s'}$，……，$\mathscr{R}_{2s-1,s'}$ 分别是 $\mathscr{R}_{1,s}$，……，$\mathscr{R}_{s-1,s}$ 沿 $(s,0)$ 平移得到的。因此，它们两两不相交，覆盖 $\mathscr{S}_{s'} \cap \{x \ge s, y \le s-1\}$，并且不与 $\mathscr{S}_{s'} \cap \{x \ge s, y \ge s\}$ 相交（在图6中，这就是 R_3）。最后，$\mathscr{R}_{s,s'}$ 是矩形 $\{s, 2s-1\} \times \{s, 2s-1\}$（在图6中，这就是 R_2）。综上，$P(s')$ 成立，归纳完成。

最后一步是证明对于所有 s，$P(s)$ 都成立，已知它对所有 2 的幂都成立。任取 s，令 s' 为大于或等于 s 的最小 2 的幂，因此我们知道 $P(s')$ 成立。令 $s'' = s'/2$。由于 $s' < 2s$，$s'' \le s$。对于 $i < s''$，$\mathscr{R}_{i,s} = \mathscr{R}_{i,s'} - (s'-s, 0)$，而对于 $s'' \le i \le s-1$，$\mathscr{R}_{i,s} = \mathscr{R}_{i,s'} \cap \{x \le s-1\} - (s'-s, 0)$。已知 $P(s')$，这意味着所有这些集合两两不相交。此外，它们覆盖了 $\mathscr{S}_{s'} \cap \{x \le s-1\} - (s'-s, 0)$，而这恰好就是 $\mathscr{S}_s$。因此，$P(s)$ 得证。□

这个构造的关键性质是，$c = n_0 \sum_{i=1}^{s-1} (m_{i+\ell_i} - m_i) + m_s \sum_{i=1}^{s-1} n_{i-h_i} - n_i) = n_0 m_s$ ( ) 的对应值关于 是软线性的。这接近最优，因为不等式 $\sum_{i=1}^{s-1}(m_{i+\ell_i} - m_i) \ge m_s - 1$ 和 $\sum_{i=1}^{s-1}(n_{i-h_i} - n_i) \ge n_0 - 1$ 意味着 c 属于 $\Omega(n_0 m_s)$。

命题 4.3. 对于上述 $R_1, \dots, R_{s-1}$，$c = n_0 \sum_{i=1}^{s-1}(m_{i+\ell_i} - m_i) + m_s \sum_{i=1}^{s-1}(n_{i-h_i} - n_i)$ 属于 $O^{\sim}(n_0 m_s)$。

证明。我们证明，采用定义 4.1中的选择后，$\sum_{i=1}^{s-1}(m_{i+\ell_i} - m_i)$ 属于 $O^{\sim}(m_s)$；我们省略以类似方式证明 $\sum_{i=1}^{s-1}(n_{i-h_i} - n_i)$ 属于 $O^{\sim}(n_0)$ 的其余部分。

首先，我们将问题归约到 s 是 2 的幂的情形。对于 $i \geq s$，令 $\ell_i = 0$ 和 $m_i = m_s$；那么和 $\sum_{i=1}^{s-1}(m_{i+\ell_i} - m_i)$ 等于 $\sum_{i=1}^{s'-1}(m_{i+\ell_i} - m_i)$，其中 $s' = 2^k$ 是大于或等于 s的最小 2 的幂。此外，这一约定意味着对所有 i都有 $m_{i+\ell_i} = m_{i+h_i}$。

对于给定的 κin $\{0, \ldots, k-1\}$，2 进赋值为 κ 的指标 $i \in \{1, \ldots, s'-1\}$ 是整数 $2^\kappa(1+2j)$，其中 $j = 1, \ldots, 2^{k-\kappa-1}-1$，因此我们可以将和 $\sum_{i=1}^{s'-1}(m_{i+\ell_i} - m_i)$ 重写为

$$\sum_{\kappa=0}^{k-1}\sum_{j=0}^{2^{k-\kappa-1}-1}(m_{2^\kappa(1+2j)+2^\kappa} - m_{2^\kappa(1+2j)}) = \sum_{\kappa=0}^{k-1}\sum_{j=0}^{2^{k-\kappa-1}-1}(d_{2^\kappa(1+2j)+1} + \cdots + d_{2^\kappa(1+2j)+2^\kappa}),$$

其中，对于 $i > s$，我们令 $d_i = 0$。特别地，对于固定的 κ，求和步骤 j 中出现的最后一个指标小于 $j+1$ 中出现的第一个指标，因此内层和的上界为 $\sum_{i=1}^{s'} d_i = m_s$。由此可得 $\sum_{i=1}^{s-1}(m_{i+\ell_i} - m_i) \leq \sum_{\kappa=0}^{k-1} m_s \in O(m_s \log(s))$。由于$s \leq m_s$，我们的结论得证。 □

4.2 混合基数表示

在本小节中，我们讨论多项式的一种替代基。我们的动机如下：如果 $\mathcal{G} = (g_0, \ldots, g_s)$ 是我们打算在约化算法中使用的最小约化字典序 Gr¨obner 基，那么我们已经看到，对于 $i = 0, \ldots, s$，g_i 可以写成 $M_i G_i$，其中 M_i 在 $\mathbb{K}[x]$ 中的次数为 m_i，且在 y 中关于 $G_i \in \mathbb{K}[x, y]$ 首一，次数为 n_i（在 y 中）。还记得，对于 $i = 1, \ldots, s$，我们写作 $D_i = M_i/M_{i-1}$，它是一个在 $\mathbb{K}[x]$ 中次数为 $d_i = m_i - m_{i-1}$ 的多项式。

主要的约化算法将对多项式 $M_1, \ldots, M_s$ 执行许多一元约化。处理 $\langle x, y\rangle$-主理想时，所有 M_i 都是 x 的幂，因此这些运算不产生算术成本。不过一般而言，如果输入以单项式基表示，情况就不再如此。本段将介绍一种混合基数表示，在这种表示中，关于 M_i 的约化是免费的，并讨论转换算法。

给定整环 $\mathbb{K}[x]$ 中的多项式 $\boldsymbol{K} = (K_1, \ldots, K_t)$，其相应次数为 $k_1, \ldots, k_t$，记 $h = k_1 + \cdots + k_t$，我们考虑 $\mathbb{K}$-线性映射

$$\begin{aligned}\Phi_{\boldsymbol{K}} : \mathbb{K}[x]_{<k_1} \times \cdots \times \mathbb{K}[x]_{<k_t} &\to \mathbb{K}[x]_{<h} \\ (F_1, \ldots, F_t) &\mapsto F_1 + K_1F_2 + K_1K_2F_3 + \cdots + K_1 \cdots K_{t-1}F_t.\end{aligned}$$

整环与陪域的维数均为 h；由此可轻易推出，$\Phi_{\boldsymbol{K}}$ 是一个$\mathbb{K}$-向量空间同构。对于 $\mathbb{K}[x]_{<h}$ 中的 F，我们称 $(F_1, \ldots, F_t) = \Phi_{\boldsymbol{K}}^{-1}(F)$ 为其关于基 $\boldsymbol{K}$的混合基表示。

我们将依赖以下事实：给定 $(F_1, \ldots, F_t) = \Phi_{\boldsymbol{K}}^{-1}(F)$，对于 i中的 $\{1, \ldots, t\}$，F整除 $K_1 \cdots K_i$ 相对于基 $(K_{i+1}, \ldots, K_t)$ 的混合基表示为 $(F_{i+1}, \ldots, F_t)$，因此我们可以免费获得该表示。类似地，F 余数 $K_1 \cdots K_i$相对于基 $(K_1, \ldots, K_i)$ 的混合基表示为 $(F_1, \ldots, F_i)$。特别地，如果 F

以混合基表示给出，则除以乘积 $K_1 \cdots K_i$所得的商和余数是免费的；我们仍将这些运算记作 div 和 rem。

反过来，对于次数小于 $k_{i+1} + \cdots + k_t$ 的 F，若其在与 $(K_{i+1}, \ldots, K_t)$ 相关联的混合基数基下表示为向量 $(F_{i+1}, \ldots, F_t)$，则其在基 $(K_1, \ldots, K_t)$ 下的混合基表示为 $(0, \ldots, 0, F_{i+1}, \ldots, F_t)$，因此可以免费计算。

为完整起见，我们给出运行时间为软线性的算法，用于计算 $\Phi_{\boldsymbol{K}}$ 及其逆变换。这些算法是中国剩余定理相关算法的初等变体，参见 [41,第 10 章。3], 或广义泰勒展开相关算法，参见 [41, 第 9 章。2]我们先从混合基表示到单项式表示的转换开始。

算法 4.1 FromMixedRadix$((F_1, \ldots, F_t), (K_1, \ldots, K_t))$

Input: $(F_1, \ldots, F_t)$ 中 $\mathbb{K}[x]_{<k_1} \times \cdots \times \mathbb{K}[x]_{<k_t}$, $\boldsymbol{K} = (K_1, \ldots, K_t)$ 的相应次数 $k_1, \ldots, k_t$

Output: $\Phi_{\boldsymbol{K}}(F_1, \ldots, F_t) \in \mathbb{K}[x]_{<h}$, 其中 $h = k_1 + \cdots + k_t$

1: 如果 $t = 1$，则返回 F_1
2: $t' \leftarrow \lceil t/2 \rceil$
3: $L \leftarrow$ FromMixedRadix$((F_1, \ldots, F_{t'}), (K_1, \ldots, K_{t'}))$
4: $R \leftarrow$ FromMixedRadix$((F_{t'+1}, \ldots, F_t), (K_{t'+1}, \ldots, K_t))$
5: 如果 $R = 0$ 则
6: 返回 L
7: 否则
8: 返回 $L + K_1 \cdots K_{t'} R$

正确性显而易见：若记 $F = \Phi_{\boldsymbol{K}}(F_1, \ldots, F_t)$，则前述讨论表明，$L = F$余数 $K_1 \cdots K_{t'}$，且 $R = F$整除 $K_1 \cdots K_{t'}$，因此输出确实是 F。若进入第8行，计算 P需要 $O\tilde{}(k_1 + \cdots + k_{t'})$ 次运算（$+, \times$），参见引理10.4]；然而，在此情况下 R非零，因此 F的次数至少为 $k_1 + \cdots + k_{t'}$，且 $O\tilde{}(k_1 + \cdots + k_{t'})$ 为 $O\tilde{}(\deg(F))$。由此可知，不计递归调用，算法 FromMixedRadix 的一次调用开销为 $O\tilde{}(\deg(F))$（若 $\deg(F) \geq k_1 + \cdots + k_{t'}$），否则为零。

递归树中有 $O(\log(\deg(F)))$ 层会产生非零成本，并且在这些层中计算出的多项式的次数总和至多为 $\deg(F)$。因此，总成本为 $O\tilde{}(\deg(F))$ 次运算（$+, \times$），在 $\mathbb{K}$ 中。

对于逆运算，该算法同样是递归的。利用第 3 行的测试，如果 F的次数小于 $k_1 + \cdots + k_{t'}$，我们就无需进行任何计算。讨论与上文相同，因此运行时间为 $O\tilde{}(\deg(F))$ 次运算（$+$, $\times$），其中 $\mathbb{K}$。

算法 4.2 ToMixedRadix(F, $(K_1, \ldots, K_t)$)

INPUT: F 在 $\mathbb{K}[x]_{<h}$, $\boldsymbol{K} = (K_1, \ldots, K_t)$，次数分别为 $k_1, \ldots, k_t$，其中 $h = k_1 + \cdots + k_t$
OUTPUT: $(F_1, \ldots, F_t) = \Phi_{\boldsymbol{K}}^{-1}(F)$
1: 若 $t = 1$ 则返回 (F)
2: $t' \leftarrow \lceil t/2 \rceil$
3: 若 $\deg(F) < k_1 + \cdots + k_{t'}$ 则
4: 返回 ToMixedRadix$(F, (K_1, \ldots, K_{t'}))$ cat $(0, \ldots, 0)$ ▷ $t - t'$ 零
5: 否则
6: $P \leftarrow K_1 \cdots K_{t'}$
7: $Q, R \leftarrow F$整除 P, F余数 P
8: 返回 ToMixedRadix$(R, (K_1, \ldots, K_{t'}))$ cat ToMixedRadix$(Q, (K_{t'+1}, \ldots, K_t))$

在接下来的几段中，我们将这些算法应用于 $\mathbb{K}[x, y]$ 中的多项式（算法名称仍沿用相同的名称）。在这种情况下，我们只需相对于 y 逐系数进行处理；$F \in \mathbb{K}[x, y]$ 的混合基数表示此时是一个二维数组。如果 K_1、.....、K_t 的次数之和为 h，且对于 $\mathbb{K}[x, y]$ 中支撑集为初始段 U 的 F，还有 $\deg(F, x) < h$，那么这两种算法的运行时间均为 $O\tilde{}(|\mathsf{U}|)$。

4.3 主算法

现在，我们可以利用前面各小节的结果，给出一个算法，用于将多项式 $f \in \mathbb{K}[x, y]$ 对极小约化字典序 Gr¨obner 基 $\mathcal{G} = (g_0, \ldots, g_s)$ 约化。目前，我们只考虑“平衡”情形，即 f 已经对 g_0 和 g_s 约化。照例，将 $\mathcal{G}$ 的首项写作

$$\boldsymbol{E} = (y^{n_0}, x^{m_1} y^{n_1}, \ldots, x^{m_{s-1}} y^{n_{s-1}}, x^{m_s})$$

其中 m_i 递增，n_i 递减，并令 S 为矩形 $\{0, \ldots, m_s - 1\} \times \{0, \ldots, n_0 - 1\}$。那么，我们的假设是 f 属于 $\mathbb{K}[x, y]_{\mathsf{S}}$。更一般的输入可以先对 (g_0, g_s) 进行约化来处理；本节最后一段将对此进行讨论。

在下文中，我们令 T 为由 $\mathcal{G}$ 确定的初始段，并令 $\delta = \dim_{\mathbb{K}}(\mathbb{K}[x, y]/\mathcal{G})$ 为 $\mathcal{G}$ 的次数。

算法概述 给定 f 属于 $\mathbb{K}[x, y]$，其中 $\deg(f, x) < m_s$ 且 $\deg(f, y) < n_0$，我们的主算法 Reduction 通过调用一个名为 PartialReduction 的过程 $s - 1$ 次来计算 $r = f$ rem $\mathcal{G}$，该过程将在下文进一步介绍。主算法返回余式 r，以及商 $Q_1, \ldots, Q_{s-1}$，使得 $f = Q_1 g_1 + \cdots + Q_s g_s + r$。虽然本文不需要这些商，但我们将它们作为副产品一并返回；它们在其他情境中可能会派上用场（算法不计算最后一个商 Q_s，但如有需要，从输出中推导出它并不困难）。由于我们假设 $\deg(f, y) < n_0$，g_0 不会出现在约化等式中。

算法全程使用混合基数基来处理中间数据；输入和输出则采用通常的单项式基。

算法 4.3 REDUCTION$(f, \mathcal{G})$

INPUT: f 在 $\mathbb{K}[x, y]$, $\mathcal{G} = (g_0, \ldots, g_s)$ 中，如上所述
ASSUMPTIONS: 次数$(f, x) < m_s$，次数$(f, y) < n_0$
OUTPUT: f 余数 $\mathcal{G}$ 和商 $Q_1, \ldots, Q_{s-1}$

$M_0 \leftarrow 1$, $G_0 \leftarrow g_0$
</style>对 $i = 1, \ldots, s$ **do**
    $M_i \leftarrow$ POLYNOMIALCOEFFICIENT$(g_i, y^{n_i}) \in \mathbb{K}[x]$
    $G_i \leftarrow g_i$ 整除 M_i
    $D_i \leftarrow M_i$ 整除 M_{i-1}
$f^{(0)} \leftarrow$ TOMIXEDRADIX$(f, (D_1, \ldots, D_s))$ ▷ $f^{(0)}$ 是混合基数基
针对 $i = 1, \ldots, s-1$ 执行
    $f_{(i)} Q_i \leftarrow$ PARTIALREDUCTION$(f^{(i-1)}, i)$ ▷ 所有 $f^{(i)}$ 均采用混合基数基
返回 FROMMIXEDRADIX$(f^{(s-1)}, D_1, \ldots, D_s)$ $Q_1, \ldots, Q_{s-1}$

为简化记号，在调用算法PARTIALREDUCTION时，我们假定多项式 $g_0, \ldots, g_s$, $G_0, \ldots, G_s$, $M_0, \ldots, M_s$ 和 $D_1, \ldots, D_s$ 已知；后者是在主算法开始时计算得到的，而不是作为参数传入。

本节的主要结果是以下命题。这里给出的运行时间关于 $n_0 m_s$ 和 $s\delta$ 是软线性的：前者表示输入多项式 f 的大小，后者是表示 $\mathcal{G}$ 所需系数个数的上界，相关讨论见第 3.1 节。我们尚不清楚是否存在更好的算法（例如，不需要 $\mathcal{G}$ 的所有系数，而只需其 Gröbner 参数）。

命题 4.4. 给定 f 和 $\mathcal{G}$，其中 $\deg(f, x) < m_s$ 且 $\deg(f, y) < n_0$，算法REDUCTION 返回 f 余数 $\mathcal{G}$，使用 $O^{\sim}(n_0 m_s + s\delta)$ 次运算 $(+, \times)$，运行时间为 $\mathbb{K}$。

在证明该命题之前，我们先介绍一个重要的特殊情形，在此情形下可以得到简化的运行时间。假设对于所有 i 都有 $e_i = 1$，也就是说，阶梯中的所有步高均为 1。在这种情况下，$n_0 = s$，并且由于我们有 $m_s \leq \delta$，可得 $n_0 m_s \leq s\delta$。换言之，该算法的运行时间就是简单的 $O^{\sim}(s\delta)$。

单个约化步骤 我们首先描述关键子程序 AlgorithmPARTIALREDUCTION。

在第 4.1 节中，我们介绍了一种用矩形 $\mathsf{R}_i = \{m_i, \ldots, m_{i+\ell_i} - 1\} \times \{n_i, \ldots, n_{i-h_i} - 1\}$ 覆盖 $\mathsf{S} = \{0, \ldots, m_s - 1\} \times \{0, \ldots, n_0 - 1\} - \mathsf{T}$ 的方法，其中 $h_i = 2^{\mathrm{val}_2(i)}$，并且对于 $i = 1$, ..., $s - 1$，有 $\ell_i = \min(h_i, s - i)$。

在算法 PARTIALREDUCTION 中，给定 $f \in \mathbb{K}[x, y]_{\mathsf{S}}$ 和 $\{1, \ldots, s-1\}$ 中的一个指标 i。核心操作是关于变量 y 进行欧几里得除法，并将系数相对于 x 适当地约化；使用混合基数基来控制

在 x 中进行这种约化的代价。我们将在下文证明，输出 r 与 f 模 $\mathcal{G}$ 具有相同的余数；我们还返回部分商 Q，其支撑集是 R_i 的一个平移。

算法 4.4 PartialReduction(f, i)

Input: f 属于 $\mathbb{K}[x, y]$, i 属于 $\{1, \ldots, s-1\}$ Assumptions: $\deg(f, x) < m_s$, $\deg(f, y) < n_0$, i 属于 $\{1, \ldots, s-1\}$。 f 基于与 $D_1, \ldots, D_s$ Output: r 相关联的混合基数基给出, Q 属于 $\mathbb{K}[x, y]$。 r 基于与 $D_1, \ldots, D_s$ 相关联的混合基数基给出 1: $h_i \leftarrow 2^{\mathrm{val}_2(i)}$, $\ell_i \leftarrow \min(h_i, s-i)$ 2: $F_1 \leftarrow f$ 在混合基数基中整除 M_i ▷, 除法▷ F_1 基于与 $D_{i+1}, \ldots, D_s$ 相关联的混合基数基给出 3: $F_2 \leftarrow F_1$ 在混合基数基中取余 $D_{i+1} \cdots D_{i+\ell_i}$ ▷, 除法▷ F_2 基于与 $D_{i+1}, \ldots, D_{i+\ell_i}$ 相关联的混合基数基给出 4: $F_3 \leftarrow F_2$ 取余 $y^{n_i - h_i}$ 5: $F_4 \leftarrow$ FromMixedRadix$(F_3, (D_{i+1}, \ldots, D_{i+\ell_i}))$ ▷ F_4 基于单项式基 6: $q \leftarrow F_4$ 整除 G_i, 其中 $\mathbb{A}[y]$ ▷ G_i 满足 $g_i = M_i G_i$, $\mathbb{A} = \mathbb{K}[x]/\langle D_{i+1} \cdots D_{i+\ell_i} \rangle$ 7: 令 Q 为 q 在 $\mathbb{K}[x, y]$ ▷$\deg(Q, x)$ $< m_{i+\ell_i} - m_i$ 中的典范提升 8: $V \leftarrow$ Multiply$(Q,$ $\{0, \ldots, m_{i+\ell_i} - m_i - 1\} \times \{0, \ldots, n_{i-h_i} - n_i - 1\}, G_i,$ T)▷ $V = QG_i$ 基于单项式基 9: $V_1 \leftarrow V$ 取余 $(D_{i+1} \cdots D_s)$ ▷ $V_1 = QG_i$ 基于单项式基取余 $(D_{i+1} \cdots D_s)$ 10: $V_2 \leftarrow$ ToMixedRadix$(V_1, (D_{i+1}, \ldots, D_s))$▷ $V_2 = QG_i$ 取余 $(D_{i+1} \cdots D_s)$, 基于与 $D_{i+1}, \ldots, D_s$ 相关联的混合基数基给出 11: $V_3 \leftarrow M_i V_2$ ▷ 在混合基数基中的乘法▷ $V_3 = Qg_i$ 取余 M_s, 基于与 $D_1, \ldots, D_s$ 相关联的混合基数基给出 12: $r \leftarrow f - V_3$ ▷ 在混合基数基中的减法▷ $r = (f - Qg_i)$ 取余 M_s, 基于与 $D_1, \ldots, D_s$ 相关联的混合基数基给出 13: 返回 r, Q

引理 4.5。调用 PartialReduction(f, i) 需要

$$O\tilde{}(n_0(m_{i+\ell_i} - m_i) + m_s(n_{i-h_i} - n_i) + \delta)$$

在 $\mathbb{K}$ 中进行 $+, \times$ 次运算，其中 $h_i = 2^{\mathrm{val}_2(i)}$ 以及 $\ell_i = \min(h_i, s-i)$。输出 r, Q 满足以下性质：

1. $\deg(r, x) < m_s$ and $\deg(r, y) < n_0$ 2. r 余数 $\mathcal{G} = f$ 余数 $\mathcal{G}$ 3. r 余数 $M_i = f$ 余数 M_i 4. r 整除 $y^{n_i - h_i} = f$ 整除 $y^{n_i - h_i}$ 5. $((r$ 整除 $M_i)$ 整除 $y^{n_i})$ 余数 $(D_{i+1} \cdots D_{i+\ell_i}, y^{n_i - h_i - n_i}) = 0$

6. $\deg(Q, x) < m_{i+\ell_i} - m_i$ 和 $\deg(Q, y) < n_{i-h_i} - n_i$

证明。我们首先验证所有步骤都有明确的定义，并讨论算法中多项式的次数性质。

按照我们在前言中的讨论，第 2 行和第 3 行的除法和余式输出一个二元多项式 F_2，它基于与 $D_{i+1}, \ldots, D_{i+\ell_i}$相关联的混合基数基。多项式 F_3 也基于同一个基表示； F_4 表示同一个多项式，不过这次是基于单项式基表示。

该多项式的 y 次数小于 n_{i-h_i}；由于 G_i 的 y 次数为 n_i， q 因而具有小于 $n_{i-h_i} - n_i$ 的 y 次数。由于 Q 的 x 次数也小于 $m_{i+\ell_i} - m_i$，它的支撑集位于矩形 $\{0, \ldots, m_{i+\ell_i} - m_i - 1\} \times \{0, \ldots, n_{i-h_i} - n_i - 1\}$ 中（该矩形是将 $\mathbf{R}_i$ 平移至原点所得）。这就证明了引理中的最后一个断言。

另一方面， G_i 的支撑集包含于T （这是因为 $i \geq 1$；对于 $i = 0$， G_0的首项 y^{n_0}不属于T），因此，对MULTIPLY 第8 行的调用是合理的。变量 V_1 和 V_2 表示同一个多项式，即 $QG_i \operatorname{rem} (D_{i+1} \cdots D_s)$，但分别基于两种不同的基（单项式基和混合基数基）。由此可知， V_3 表示多项式

$$\begin{aligned} M_i(QG_i \text{ rem } (D_{i+1} \cdots D_s)) &= M_i QG_i \text{ rem } (M_i D_{i+1} \cdots D_s) \\ &= Qg_i \text{ rem } M_s. \end{aligned}$$

正如我们在上一小节中所指出的，由于 V_2 是基于与（D_{i+1}，……， D_s）相关联的混合基数基表示的， V_3是基于与（D_1，……， D_s）相关联的混合基数基表示的。由于 f也具有这一性质，第12 行的减法是逐系数进行的，结果得到多项式（$f - Qg_i$）rem M_s，它仍然基于同一混合基数基表示。

话虽如此，我们来证明性质 1-5。第一项：我们有 $\deg(f, y) < n_0$。另一方面， Q 的次数界意味着 Qg_i 的 y次数小于 n_{i-h_i}。由于$n_{i-h_i} \leq n_0$，乘积 Qg_i 的 y次数也小于 n_0，因此r也满足这一条件。根据构造，$\deg(r, x) < m_s$ 这一界成立。

第二项：我们可以写成 $r = f - Qg_i + hM_s = f - Qg_i + hg_s$，其中某个 h 属于 $\mathbb{K}[x, y]$，使得 $r - f$ 属于理想 $\langle \mathcal{G} \rangle$。

第三项：再次考虑表达式 $r = f - Qg_i + hM_s$，并注意到 M_i 整除 g_i 和 M_s。

第四项：因为 $\deg(f, x) < m_s$，关系式 $r = f - Qg_i + hM_s$中的商 h 为 $-Qg_i$ 整除 M_s。由于 Qg_i 的 y-次数小于 n_{i-h_i}，因此 h也满足这一点。这表明 r 整除 $y^{n_{i-h_i}} = f$ 整除 $y^{n_{i-h_i}}$，如所述。

第五项：由于 $r = f - Qg_i + hM_s = f - QM_iG_i + hM_s$，我们有 r 整除 $M_i = F_1 - QG_i + hD_{i+1} \cdots D_s$。根据定义，对于某些 K、 $L \in \mathbb{K}[x, y]$，有 $F_1 = F_2 + LD_{i+1} \cdots D_{i+\ell_i}$ 和 $F_2 = F_3 + Ky^{n_{i-h_i}}$。 F_4 与 F_3是同一个多项式，只是写在不同的基下，并满足 $F_4 = QG_i + P + L'D_{i+1} \cdots D_{i+\ell_i}$；其中， P和 L'属于 $\mathbb{K}[x, y]$，且 P的 y-次数小于 n_i。综上，我们得到 r整除 $M_i = P + (L + L')D_{i+1} \cdots D_{i+\ell_i} + hD_{i+1} \cdots D_s + Ky^{n_{i-h_i}}$。因此，

$$(r \text{ div } M_i) \text{ div } y^{n_i} = ((L + L') \text{ div } y^{n_i})D_{i+1} \cdots D_{i+\ell_i} + (h \text{ div } y^{n_i})D_{i+1} \cdots D_s + Ky^{n_{i-h_i} - n_i}.$$

由于 $i+\ell_i \le s$，这个表达式模 $(D_{i+1}\cdots D_{i+\ell_i}, y^{n_{i-h_i}-n_i})$ 后为零，正如所要证明的。

接下来需要估算该算法的开销。第2行和第3 行的带余除法无需成本（因为我们在合适的混合进位基中进行运算）；第4行也是如此，因为它只涉及 y的幂。

由于 $D_{i+1}\cdots D_{i+\ell_i}$的次数为 $m_{i+\ell_i}-m_i$，第5 行的转换需要 $O^{\sim}(n_{i-h_i}(m_{i+\ell_i}-m_i))$ 次运算（$+$，$\times$），在 $\mathbb{K}$中，即 $O^{\sim}(n_0(m_{i+\ell_i}-m_i))$。

在第 6 行进行除法之前，G_i必须先对 $D_{i+1}\cdots D_{i+\ell_i}$ 取模；在 y 中逐系数处理，需要 $O^{\sim}(|\mathsf{T}|)=O^{\sim}(\delta)$ 次运算（$+,\times$），在 $\mathbb{K}$ 中。然后，在 $\mathbb{A}[y]$ 中进行除法需要 $O^{\sim}(n_{i-h_i})$ 次运算（$+,\times$），在 $\mathbb{A}$ 中；这相当于 $O^{\sim}(n_{i-h_i}(m_{i+\ell_i}-m_i))$ 次运算（$+,\times$），在 $\mathbb{K}$ 中。对于这个表达式，使用与上文相同的上界 $O^{\sim}(n_0(m_{i+\ell_i}-m_i))$ 就足够了。

接下来，我们考虑计算 V 在 $\mathbb{K}[x,y]$中的乘积的代价。输入 Q 的 x 次数小于 $m_{i+\ell_i}-m_i$，且 y 次数小于 $n_{i-h_i}-n_i$；而 G_i 的支撑集包含于高度为 n_0、宽度为 m_s、基数为 δ 的初始段 T 中。因此，利用命题 2.4（以及命题后关于 QG_i 支撑集大小的说明），可知 QG_i 能在 $\mathbb{K}$ 中用 $O^{\sim}((m_{i+\ell_i}-m_i)(n_{i-h_i}-n_i)+n_0(m_{i+\ell_i}-m_i)+m_s(n_{i-h_i}-n_i)+\delta)$ 次运算计算得出（$+$，$\times$）。这同样是 $O^{\sim}(n_0(m_{i+\ell_i}-m_i)+m_s(n_{i-h_i}-n_i)+\delta)$。

第 9 行的欧几里得除法是在单项式基下进行的，并在 y 中逐系数处理。计算 $D_{i+1}\cdots D_s$ 需要在 $\mathbb{K}$ 中进行 $O^{\sim}(m_s)$ 次运算（$+$，$\times$）。随后，约化所需的时间与 V 支撑集的大小呈准线性关系，也就是再次需要 $O^{\sim}(n_0(m_{i+\ell_i}-m_i)+m_s(n_{i-h_i}-n_i)+\delta)$。回顾一下，对于支撑集包含于初始段 U 的多项式，转换为混合基数基所需的时间与 U 的大小呈准线性关系。此处，支撑集 U 包含于 $V=QG_i$ 的支撑集中，因此第 10 行的转换同样需要时间 $O^{\sim}(n_0(m_{i+\ell_i}-m_i)+m_s(n_{i-h_i}-n_i)+\delta)$。

在混合基数基中乘以 M_i 无需代价，因为只需在每个条目前添加一个零向量，即可得到 V_3。最后一步的多项式减法中，V_3的每个非零系数都需要在 $\mathbb{K}$ 中进行一次减法，因此总共需要 $O^{\sim}(n_0(m_{i+\ell_i}-m_i)+m_s(n_{i-h_i}-n_i)+\delta)$。 □

主算法的正确性 上述性质使我们能够证明算法 Reduction 能够正确计算 f 除以 $\mathcal{G}$的余式。

我们如下定义 $\{1,\dots,s\}$ 中的指标 $(b_{i,j})_{0\le i<s,0\le j<n_0}$。对于 $i=0,\dots,s-1$，令 $\mathsf{T}_i\subset\mathbb{N}^2$为初始段 T 与矩形 $\mathsf{R}_1,\dots,\mathsf{R}_i$的并集；特别地，$\mathsf{T}_0=\mathsf{T}$，且 T_{s-1}是矩形 $\{0,\dots,m_s-1\}\times\{0,\dots,n_0-1\}$。接着，对于 $i=0,\dots,s-1$和$j=0,\dots,n_0-1$，我们令 $b_{i,j}\in\{1,\dots,s\}$ 为满足 (m_k,j) 不属于 T_i的最小指标 k。特别地，对于所有 $j<n_0$，都有 $b_{s-1,j}=s$。另一方面，对于 $i=0$，我们看到，任何满足 $u<m_{b_{0,j}}$ 的数对 (u,j) 都属于 T，因此 x^uy^j 对 $\mathcal{G}$取余。

令 $f_{(0)}\dots,f^{(s-1)}$为整个算法过程中计算得到的多项式（引理 4.5 的第一项证明了这些多项式定义良好，且都支撑在矩形 $\{0,\dots,m_s-1\}\times\{0,\dots,n_0-1\}$上）。我们通过对 $A(i$进行归纳法，证明下文记作 $i=0,\dots,s-1$ 的以下断言：对于 $n_i\le j<n_0$，多项式

PolynomialCoefficient($f^{(i)}, y^j$) 对 $M_{b_i \text{取余}, j} \in \mathbb{K}[x]$ 其次数小于 $m_{b_0, j}$。对于 $i = 0$，无需证明（因为没有需要考虑的指标 j）。假设对于某个 i 属于 $\{1, \ldots, s-1\}$， $A(i-1)$ 成立；我们证明 $A(i)$。

对于 $j \geq n_{i-h_i}$，引理 4.5 的第 4 项表明， PolynomialCoefficient($f^{(i)}, y^j$) = PolynomialCoefficient($f_{(i-1)} y^j$)。由于在该情况下我们还得到 $b_{i,j} = b_{i-1,j}$，因此我们的断言成立。现在，假设 j 属于 $\{n_i, \ldots, n_{i-h_i} - 1\}$；在这种情况下，同一引理的第 3 项和第 5 项推出， PolynomialCoefficient($f^{(i)}, y^j$) 余数 $M_{i+\ell_i}$ 等于PolynomialCoefficient($f^{(i-1)}, y^j$) 余数 M_i. 另一方面，我们还得到 $b_{i-1,j} = i$以及 $b_{i,j} = i+\ell_i$，因此左侧是项 PolynomialCoefficient($f^{(i)}, y^j$) 余数 $M_{b_i,j}$，该项正是我们断言中出现的项。因此，要完成归纳证明，只需证明 PolynomialCoefficient($f^{(i-1)}, y^j$) 余数 M_i 的次数小于 $m_{b_0,j}$。我们通过进一步的分类讨论来完成这一点：

- 若 $j \geq n_{i-1}$，我们可以使用归纳假设。这意味着余式PolynomialCoefficient($f_{(i-1)} y^j$) rem $M_{b_{i-1}, j}$的次数小于 $m_{b_0, j}$。由于我们已知 $b_{i-1, j} = i$，因此得证。

- 若 $j < n_{i-1}$，则有 $b_{0, j} = i$，因此$m_{b_0, j} = m_i = \deg(M_i)$，我们的断言也成立。

既已确立归纳断言，我们可以取 $i = s-1$。于是， $A(s-1)$表明，对于 j属于 n_{s-1}， ....., $n_0 - 1$， PolynomialCoefficient($f_{(s-1)} y^j$) rem M_s的次数小于 $m_{b_0, j}$。根据构造， $f^{(s-1)}$ 模 M_s 是约化的，因此 PolynomialCoefficient($f_{(s-1)} y^j$) 本身的次数小于 $m_{b_0, j}$。现在，对于 j属于 0， ....., $n_{s-1} - 1$，有 $b_{0, j} = s$，因此在这种情况下， PolynomialCoefficient($f^{(s-1)}$, y^j) 的次数也小于 $m_{b_0, j}$。总之，正如我们在引入 $m_{b_0, j}$ 时指出的，这证明了 $f^{(s-1)}$ 模 $\mathcal{G}$ 是约化的。

引理 4.5的第二项最终表明， f余数 $\mathcal{G} = f_{(s-1)}$余数 $\mathcal{G}$，因此 $f^{(s-1)}$确实是 f 模 $\mathcal{G}$的标准形。至此，正确性证明完成。

成本分析 进行成本分析时，我们先考察主算法开始时多项式 M_i、G_i 和 D_i的计算。由于除以首一单变量多项式需要软线性时间， Reduction 的第 2 行处循环的每次迭代需要 $O\tilde{}(\delta)$ 次运算，总计需要 $O\tilde{}(s\delta)$ 次运算。

转换到混合基数基以及从混合基数基转换回来所需的时间，对于 f支撑集的大小而言是准线性的，也就是需要 $O\tilde{}(n_0 m_s)$ 次运算。接下来，只需将调用 PartialReduction的成本相加。根据引理 4.5，从 $f^{(i-1)}$推导出 $f^{(i)}$需要在 $\mathbb{K}$中进行 $O\tilde{}(n_0(m_{i+\ell_i} - m_i) + m_s(n_{i-h_i} - n_i) + \delta)$ 次运算，其中 $\delta = |\mathsf{T}|$，因此只需将此数量对 $i = 1$ 至 $s-1$ 求和。前两项相加后总计为 $O\tilde{}(n_0 \sum_{i=1}^{s-1}(m_{i+\ell_i} - m_i) + m_s \sum_{i=1}^{s-1}(n_{i-h_i} - n_i))$。命题 4.3 表明，这个和为 $O\tilde{}(n_0 m_s)$，因此将每个加数中的 $O\tilde{}(\delta)$ 项计入后，总计为 $O\tilde{}(n_0 m_s + s\delta)$，如所述。

对任意输入的推广与讨论：如果输入 f 不满足条件 $\deg(f, x) < m_s$且 $\deg(f, y) < n_0$，我们就通过对多项式对进行约化，转而处理这种情况

模去多项式对 (g_0, g_s)，其首项分别为 y^{n_0} 和 x^{m_s}。下面这个直接的算法可以实现这一点；我们将在下文讨论可能的改进。

算法 4.5 ReductionGeneralInput$(f, \mathcal{G})$

Input: f 在 $\mathbb{K}[x, y]$, $\mathcal{G} = (g_0, \ldots, g_s)$
Output: f 余数 $\mathcal{G}$
1: $f_1 \leftarrow f$ 余数 g_s
2: $f_2 \leftarrow f$ 余数 g_0 属于 $\mathbb{A}[y]$ ▷ $\mathbb{A} = \mathbb{K}[x]/\langle g_s \rangle$
3: 令 f_3 为 f_2 在 $\mathbb{K}[x, y]$ 中的典范提升 ▷ $\deg(f_3, x) < m_s$
4: 返回 Reduction$(f_3, \mathcal{G})$

命题 4.6. 给定 f 和 $\mathcal{G}$，其中 $\deg(f, x) < d$ 且 $\deg(f, y) < e$，算法 ReductionGeneralInput 在 $\mathbb{K}$中使用 $O^{\sim}(ed + em_s + n_0 m_s + s\delta)$ 次运算，返回 f rem $\mathcal{G}$ $(+, \times)$。若 $\mathcal{G}$生成一个 $\langle x, y \rangle$-主理想，则运行时间为 $O^{\sim}(\delta m_s)$ 次运算 $(+, \times)$，在 $\mathbb{K}$中。

证明。将 f 模 g_s 需要 $O^{\sim}(ed)$ 次运算（如果 $d < m_s$，实际上无需运算）。然后，在 $\mathbb{A}[y]$ 中用 g_0 进行欧几里得除法 $O^{\sim}(e)$ 步，耗时 $\mathbb{A}$，即 $O^{\sim}(em_s)$ 步，耗时 $\mathbb{K}$。最后，命题 4.4 给出最后一步的代价为 $O^{\sim}(n_0 m_s + s\delta)$。

若 $\mathcal{G}$ 生成一个 $\langle x, y \rangle$-主理想，则在约化过程中，y-次数至少为 δ 的所有项都会消失（因此我们可以将 e 替换为 δ）；同样，x-次数至少为 m_s 的所有项也会消失（因此我们可以将 d替换为 m_s）。 □

在一般情形的运行时间中，ed 是输入 f的支撑集大小，而 $s\delta$是我们对 $\mathcal{G}$大小的界，因此它们基本上是不可避免的（当然，除非能够避免使用 $\mathcal{G}$ 本身，而只使用其Gröbner参数）。运行时间中还包含额外的项 em_s 和 $n_0 m_s$，但要消除它们并无条件地将运行时间改进为 $O^{\sim}(ed + s\delta)$，似乎极具挑战性。

事实上，考虑模复合问题：给定 $\mathbb{K}[x]$中的 F、G、H，其中 F是次数为 n 的首一 G，且 H的次数小于 n，问题就归结为计算 $G(H)$ rem F。直接方法需要二次时间，而 Brent-Kung 婴儿步/巨人步算法使用 $O(n^{1.69})$次运算（并依赖快速矩阵运算）。自1978年以来，将其降至拟线性运行时间一直是一个悬而未决的问题：迄今为止，已知只有在有限 $\mathbb{K}$ [53]上才能实现这一点；对于任意 $\mathbb{K}$，目前最优算法的拉斯维加斯代价为 $O(n^{1.43})$ [54]。

事实证明，模复合是我们在此考虑的约化问题的一个特例。沿用上文的 F, G, H，若考虑 $\mathcal{G} = (y - H(x)\ F(x))$ 和多项式 $f = G(y)$，那么余式 f rem $\mathcal{G}$ 恰好就是 $G(H)$ rem F。这里，我们有$n_0 = 1$, $s = 1$, $m_s = n$, $\delta = n$, $d = 1$ 以及 $e = \deg(G, y) + 1$，因此一般情况下 $e = n$；对于此类输入，我们的算法运行时间为 $O^{\sim}(n^2)$。若能将结果改进至 $O^{\sim}(ed + s\delta)$，便可得到软线性模组合成算法，从而解决一个长期悬而未决的问题。

另一方面，当 f 在 x 和 y中的次数都很大时，也就是 $m_s \le d$和 $n_0 \le e$，这种情况特别有利，因为此时运行时间确实会变为 $O\tilde{}(ed + s\delta)$。另一种有利情况是所有 e_i 都等于 1，因为如前所述，此时我们有$n_0 m_s \le s\delta$，因此运行时间为 $O\tilde{}(ed + em_s + s\delta)$。

最后，我们指出命题 4.6 在模乘法中的一个应用：给定A， B属于 $\mathbb{K}[x,y]_{\mathsf{T}}$，其中 T 是由 $\mathcal{G}$确定的初始段，计算 $f = AB$ 余数 $\mathcal{G} \in \mathbb{K}[x,y]_{\mathsf{T}}$。此时，我们有 $d < 2m_s$ 且 $e < 2n_0$，因此运行时间为 $O\tilde{}(n_0 m_s + s\delta)$；当所有 e_i 都等于 1 时，这就变为 $O\tilde{}(s\delta)$。据我们所知，此问题此前尚无相关结果。

5 从 Gr¨obner 参数到 Gr¨obner 基

本节中，我们固定一个给定的 Gr¨obner 胞（或等价地，单项式 $\boldsymbol{E}$）。我们说明如何显式构造映射 $\Phi_{\boldsymbol{E}}$：$\mathbb{K}^N \to \mathcal{C}(\boldsymbol{E})$，该映射以 Gr¨obner 参数为输入，并输出相应的约化 Gr¨obner 基（见第 3.2 节）。

首先，我们固定一种为以下多项式的 N 系数编索引的方法 $(\sigma_{i,\ j})_{0\le i\le s-1,\ i\le j\le s}$ 这些多项式出现在协 syzygy（3）中；这将通过下面给出的互逆过程来完成。这里，为简单起见，我们假设给定单项式 $\boldsymbol{E}$后，可以直接访问整数 s， $(d_i)_{1\le i\le s}$ 和 $(e_i)_{1\le i\le s}$。

算法 5.1 SigmaFromParameters($\boldsymbol{E}$, $(\lambda_1, \ldots, \lambda_N)$)

Input: 单项式 $\boldsymbol{E}$， $(\lambda_1, \ldots, \lambda_N)$ 在 $\mathbb{K}^N$
Output: 多项式 $(\sigma_{i,\ j})_{0\le i\le s-1,\ i\le j s}$ 在 $\mathbb{K}[x,y]$

$k \leftarrow 1$
对于 $i = 0, \ldots, s-1$ 执行
  $\sigma_{i,i} \leftarrow \sum_{0\le \ell < d_{i+1}} \lambda_{k+\ell} x^\ell$
  $k \leftarrow k + d_{i+1}$
  对于 $j = i+1, \ldots, s$， 执行
    $\sigma_{i,j} \leftarrow 0$
    对于 $m = 0, \ldots, e_{j-1}$ **do**
      $\sigma_{i,j} \leftarrow \sigma_{i,j} + \sum_{0\le \ell < d_{i+1}} \lambda_{k+\ell} x^\ell y^m$
      $k \leftarrow k + d_{i+1}$
返回 $(\sigma_{i,j})_{0\le i\le s-1, i\le j\le s}$

算法 5.2 ParametersFromSigma($\boldsymbol{E}$, $(\sigma_{i,j})_{i,j}$)

Input: 单项式 $\boldsymbol{E}$，多项式 $(\sigma_{i,j})_{i,j}$ 属于 $\mathbb{K}[x,y]$
Output: $(\lambda_1,\dots,\lambda_N)$ 在 $\mathbb{K}^N$

```
k ← 1
对于 i = 0, ..., s − 1 执行
    对于 ℓ = 0, ..., d_{i+1} − 1 执行 λ_{k+ℓ} ← Coefficient(σ_{i,i}, x^ℓ)
    k ← k + d_{i+1}
    对于 j = i + 1, ..., s 执行
        σ_{i,j} ← 0
        对于 m = 0, ..., e_{j−1} do
            对于 ℓ = 0, ..., d_{i+1} − 1 执行 λ_{k+ℓ} ← Coefficient(σ_{i,j}, x^ℓ y^m)
            k ← k + d_{i+1}
返回 (λ_1, ..., λ_N)
```

为处理特殊的零维 Gröbner 参数情形，需要做一些显而易见的修改，例如将$\sigma_{0,\,0}$，……，$\sigma_{s-1,\,s-1}$设为零，并确保 x 整除$\sigma_{1,\,0}$，……，$\sigma_{s,\,s-1}$ 在 SigmaFromParameters中。我们将 SigmaFromPunctualParame-ters 和 PunctualParametersFromSigma 称为由此得到的过程。

现在可以给出一个称为 ReducedBasisFromParameters 的算法，它描述了映射 $\Phi_{\boldsymbol{E}}:\ \mathbb{K}^N \to \mathcal{C}(\boldsymbol{E})$。这个过程相当直接；用于逆运算的算法称为 ParametersFromReducedBasis，稍微复杂一些，将在下一节中介绍。我们仍采用第 3.2 节的记号，特别地，将 $M_i \in \mathbb{K}[x]$ 记作 y^{n_i} 在 g_i 和 h_i 中的多项式系数，其中所有i均成立，并将 m_i 记作其次数。

我们按降序计算 h_i，然后计算 g_i。为得到前者，我们只需使用公式（3）。对于任意 $i = s-1,\dots,0$，假设已知 h_i 和 $g_{i+1},\dots,g_s$，下面说明如何通过约化 g_i 来得到它（对于 h_i，我们有 $i=s$），并使用上一节中的过程 $g_s = h_s$。

关于 x 使用欧几里得除法，多项式 h_i 可写为 $h_i = A_i M_{i+1} + B_i$，其中 A_i 和 B_i 属于 $\mathbb{K}[x,y]$，且 $\deg(B_i,x) < m_{i+1}$。

现在回想一下，所有多项式 $g_{i+1},\dots,g_s$ 都是 M_{i+1} 的倍数，并且族$\mathcal{G}_i = (g_{i+1}/M_{i+1},\dots,g_s/M_{i+1})$ 是一个零维 Gröbner 基（如¯式 (2) 后所述）。令 $h_i = (A_i$ 余数 $\mathcal{G}_i)M_{i+1} + B_i$；我们断言 $h_i = g_i$。首先，确定其首项：A_i 余数 $\mathcal{G}_i$ 中的所有单项式，因而 $(A_i$ 余数 $\mathcal{G}_i)$ M_{i+1} 中的所有单项式，其 y-次数都小于¯ n_{i+1}；而 B_i 包含 h_i 的首项 $x^{m_i}y^{n_i}$。因此，h_i 的首项仍然是¯ $x^{m_i}y^{n_i}$。接下来，我们验证 h_i 对 $g_1,\dots,g_{i-1},g_{i+1},\dots,g_s$ 取模后是约化的。

- g_1、……、g_{i-1} 中没有任何一个能约化 h_i中的项，因为这个多项式的 y次数为 n_i。
- 由于 A_i 对 $\mathcal{G}_i$ 取余后模 $\mathcal{G}_i$已约化，因此（A_i 对 $\mathcal{G}_i$取余后）M_{i+1} 模 g_{i+1}、……、g_s已约化。
- 由于 B_i 的 x-次数小于 m_{i+1}，因此它对 $g_{i+1},\dots,g_s$ 取模后也已经约化。

¯最后一点是，差 $h_i - h_i$ 属于理想 $\langle g_{i+1}, \ldots, g_s \rangle$。总而言之，¯这就证明了 $h_i = g_i$。

据此，我们可以给出计算 g_0，……，g_s的算法。要将二元多项式 A_i 模 $\mathcal{G}_i$约化，我们使用过程Reduction。注意，该过程的次数假设均已满足：多项式 A_i 的 x次数小于 $m_s - m_{i+1}$，且 y次数小于 n_{i+1}，这恰好分别是 x次数和 y次数的最大值，这些元素属于 $\mathcal{G}_i$。

和之前一样，我们假设给定 $\boldsymbol{E}$ 后，可以直接访问整数 s、$(d_i)_{1 \le i \le s}$ 和 $(e_i)_{1 \le i \le s}$，并在伪代码中自由使用它们。

算法 5.3 ReducedBasisFromParameters($\boldsymbol{E}$, $(\lambda_1, \ldots, \lambda_N)$)

Input: 单项式 $\boldsymbol{E}$, $(\lambda_1, \ldots, \lambda_N)$ 属于 $\mathbb{K}^N$Output: 约化 Gr¨obner 基的 $\Phi_{\boldsymbol{E}}(\lambda_1, \ldots, \lambda_N)$
1: $(\sigma_{i,j})_{i,j} \leftarrow$ SigmaFromParameters($\boldsymbol{E}$, $(\lambda_1, \ldots, \lambda_N)$) 2: $M_0 \leftarrow 1$3: 对于 $i = 1, \ldots,$ s，执行 $M_i \leftarrow (x^{d_i} - \sigma_{i-1,i-1})M_{i-1}$4: $h_s \leftarrow M_s$; $g_s \leftarrow M_s$5: 对于 $i = 0, \ldots, s-1$，执行 6: $T_i \leftarrow$ KroneckerMultiply($y^{e_{i+1}}, h_{i+1}$) $+ \cdots +$ KroneckerMultiply($\sigma_{s,i}, h_s$)
7: $h_i \leftarrow T_i$ 整除 $(x^{d_{i+1}} - \sigma_{i,i})$ 8: $\mathcal{G}_i \leftarrow (g_{i+1}$ 整除 $M_{i+1}, \ldots, g_s$ 整除 $M_{i+1})$ 9: $A_i, B_i \leftarrow h_i$ 整除 M_{i+1}, h_i 余数 M_{i+1}¯10: $A_i \leftarrow$ Reduction($A_i, \mathcal{G}_i$)¯11: $g_i \leftarrow A_i M_{i+1} + B_i$12: 返回 ($g_0, \ldots, g_s$)

命题 5.1. 给定 $\mathbb{K}$ 中的单项式 $\boldsymbol{E}$ 和 $(\lambda_1, \ldots, \lambda_N)$，ReducedBasisFromPara-meters($\boldsymbol{E}$, $(\lambda_1, \ldots, \lambda_N)$) 使用 $O\tilde{}(s^2 n_0 m_s)$ 次运算，在 $\mathbb{K}$ 中求得 $\Phi_{\boldsymbol{E}}(\lambda_1, \ldots, \lambda_N)$ 的约化 Gr ¨obner 基。

证明. 正确性由前述讨论得出。关于运行时间，第一步不进行任何算术运算，而计算每个多项式 M_i 需要 $O\tilde{}(\delta)$ 次运算，总计 $O\tilde{}(s\delta)$。

对于给定的指标 i，计算 T_i 最多涉及 s 次多项式乘法，每次乘法在 $\mathbb{K}$ 中需要 $O\tilde{}(n_0 m_s)$ 次运算（$+, \times$）；由此可知，h_i 也能在相同的渐近时间内求得。计算 $\mathcal{G}_i$ 所需的欧几里得除法需要 $O\tilde{}(s\delta)$ 次运算（因为 $\mathcal{G}$ 中的所有多项式的支撑集都位于一个大小为 δ 的初始段上）；出于同样的原因，计算 A_i 和 B_i 所需的欧几里得除法需要 $O\tilde{}(n_0 m_s)$、¯。命题 4.4 表明，我们能在 $O\tilde{}(n_0 m_s + s\delta)$ 次运算（$+, \times$）内计算 A_i。最后，得到 g_i 的乘积和同样需要 $O\tilde{}(n_0 m_s)$ 次运算（$+, \times$）。

总的来说，步骤 i 的代价为 $O\tilde{}(s n_0 m_s + s\delta)$，即 $O\tilde{}(s n_0 m_s)$，由此可得整体运行时间估计。 □

值得注意的是，该算法不进行除法，因此，如果输入参数属于环 $\mathbb{A} \subset \mathbb{K}$，则输出多项式 $\mathcal{G}$ 的系数都属于 $\mathbb{A}$。

整个过程可以通过使用¨，以直接明了的方式调整为处理定点 Gr SigmaFromPunctualParameters 在第1行。由此得到的函数称为 ReducedBasisFromPunctualParameters，其运行时间也类似。

6 计算 Gr¨obner 参数

现在，我们可以给出用于计算零维理想 I 的 Gr¨obner 参数的算法。

我们在两种不同的情境下进行此操作。第一种情形是从 I 的约化 Gr¨obner 基出发恢复这些参数（即计算前文定义的映射 $\Phi_{\boldsymbol{E}}^{-1}$ ）。这相对直接，只需依次进行一系列欧几里得除法。

我们介绍的第二种变体是主算法的核心部分。在这里，我们考虑一个理想 J，它由生成元 f_1、……、 f_t生成；这是一个零维理想 I，包含 J，我们将描述一个多项式系统，其解之一是¨的 Gr I obner 参数，且重数为一。在此方面，我们沿用了 Hauenstein、Mourrain 和 Szanto 先前的工作[37]，该工作研究的是边界基表示。

后面这些方程通常过于复杂，无法直接处理。在下一节中，我们将用它们来描述主算法，即一种牛顿迭代，用于计算上述 I 的 Gr¨obner 参数。

6.1 从约化基开始

在本小节中，假设给定了零维理想 I 的约化 Gr¨obner 基 $\mathcal{G} = (g_0, \ldots, g_s)$，我们将说明如何计算其 Gr¨obner 参数。我们还将指出，当 I 为 $\langle x, y\rangle$-primary 时，该过程如何略作简化。

记号与前文相同：多项式 $(g_0, \ldots, g_s)$ 的首项记作 $\boldsymbol{E} = (y^{n_0}, x^{m_1}y^{n_1}, \ldots, x^{m_{s-1}}y^{n_{s-1}}, x^{m_s})$，$\mathcal{G}$ 的次数为 δ，而 $N = \delta + m_s$ 是 Gr¨obner 参数的个数。接下来，我们计算 Syzygy (3) 中出现的多项式 $(\sigma_{i,j})_{i,j}$ ，其系数即为 I 的 Gr¨obner 参数。回忆一下，我们将 $D_i = x^{d_i} - \sigma_{i-1,i-1}$ 记作 $i = 1, \ldots, s$，并将 $M_i = (x^{d_1} - \sigma_{0,0}) \cdots (x^{d_i} - \sigma_{i-1,i-1})$ 记作 $i = 0, \ldots, s$，其中空乘积等于 1。

推导算法 已知约化 Gr¨obner 基 $\mathcal{G} = (g_0, \ldots, g_s)$，一些多项式 $(\sigma_{i,\ j})$ 很容易计算：对于 $i = 1, \ldots, s$，我们在上一节看到，y^{n_i} 在 g_i 中的多项式系数恰好就是 M_i。已知 $M_1, \ldots, M_s$ 可得 $D_1, \ldots, D_s$，进而通过连续除法得到 $\sigma_{0,\ 0}, \ldots, \sigma_{s-1,\ s-1}$。

现在令 h_0，…，h_s 为此前使用过的非约化格罗布纳基¨，它满足公式（3），并回想一下，对于 $i = 0$，…，s，M_i 整除 h_i。定义 $H_i = h_i/M_i$，并且

再考虑公式（4），它是对（3）的改写：

$$D_{i+1}h_i - y^{e_{i+1}}h_{i+1} = \sum_{j=i+1}^{s} \sigma_{j,i}h_j,$$

其中等式两边都可以被 M_{i+1} 整除。进行除法后，我们得到

$$H_i - y^{e_{i+1}}H_{i+1} = \sum_{j=i+1}^{s} \sigma_{j,i}D_{i+2}\cdots D_jH_j. \tag{6}$$

固定 i in $\{0,\ldots,s-1\}$，并假设我们已经计算出 $H_{i+1},\ldots,H_s$；下面说明如何计算 $\sigma_{i+1,i},\ldots,\sigma_{s,i}$，然后计算 H_i。

根据构造，多项式 $(g_0,\ldots,g_i,h_{i+1},\ldots,h_s)$ 也构成 I 的一个极小 Gr¨obner 基。多项式 $h_i - g_i$ 属于 I，因此经由这些多项式作除法可将其约化为零。由于 g_i 和 h_i 都以 y^{n_i} 的多项式系数 M_i 为系数，$h_i - g_i$ 在 y 中的次数小于 n_i。这意味着列表中唯一可能约化它的多项式是 $h_{i+1},\ldots,h_s$。我们依次用 h_{i+1}、再用 h_{i+2} 等约化 $h_i - g_i$；对于 $j = i,\ldots,s$，记 $R_{i,j}$ 为经由 $h_{i+1},\ldots,h_j$ 约化后得到的余式，因此 $R_{i,i} = h_i - g_i$。

引理 6.1. 对于 $j = i,\ldots,s$, $R_{i,j}$的 y-次数小于 n_j。

证明。我们指出，对于 $j = i$，结论成立，因此假设结论对某个指标 $j < s$ 成立，并证明它对指标 $j+1$也成立。为了得到 $R_{i,j+1}$，我们用 h_{j+1}约化 $R_{i,j}$；其首项为 $x^{m_{j+1}}y^{n_{j+1}}$，因此可以写成 $R_{i,j+1} = A_{j+1} + B_{j+1}$，其中 $\deg(B_{j+1},y) < n_{j+1}$，$\deg(A_{j+1},x) < m_{j+1}$，且 A_{j+1} 中的所有项的 y-次数至少为 n_{j+1}。最后，我们证明 $A_{j+1} = 0$。

由于我们使用字典序 $y \succ x$，用 h_{j+1} 约化一项不会增加其 y-次数；根据假设，$R_{i,j}$ 的 y-次数小于 n_j，因此 A_{j+1}也满足这一点。特别地，A_{j+1} 模 $\mathcal{H}$ 已约化。由于 $R_{i,j}$ 模 $\mathcal{H}$ 约化后为零，因此A_{j+1} +(B_{j+1} 余数 $\mathcal{H}$) $= 0$。现在，出于同样的理由，(B_{j+1} 余数 $\mathcal{H}$) 的 y- 次数小于 n_{j+1}，所以 A_{j+1} 与 (B_{j+1} 余数 $\mathcal{H}$) 的支撑集不相交。这就说明 A_{j+1} = (B_{j+1} 余数 $\mathcal{H}$) $= 0$，证毕。 □

这个引理表明，$h_i - g_i$ 的约化会导出如下形式的等式

$$h_i - g_i = \sum_{j=i+1}^{s} q_{j,i}h_j,$$

对于满足以下条件的某些多项式 $q_{j,\,i}$ 其中 $\mathbb{K}[x,y]$ 对所有 j均有 $\deg(q_{j,\,i},y) < n_{j-1} - n_j = e_j$。等价地，我们可以将其改写为

$$h_i = g_i + \sum_{j=i+1}^{s} q_{j,i}M_jH_j,$$

因此，将两边除以 M_i 后，

$$H_i = G_i + \sum_{j=i+1}^{s} q_{j,i} D_{i+1} \cdots D_j H_j. \tag{7}$$

结合式（6）和式（7），得到

$$G_i - y^{e_{i+1}} H_{i+1} = \sum_{j=i+1}^{s} Q_{j,i} H_j, \quad \text{with} \quad Q_{j,i} = (\sigma_{j,i} - q_{j,i} D_{i+1}) D_{i+2} \cdots D_j. \tag{8}$$

特别要注意，对于所有 j，都有 $\deg(Q_{j,i}, y) < e_j$，因此 $\deg(Q_{j,i} H_j, y) < n_{j-1}$。在本段中，对于 F in $\mathbb{K}[x, y]$，我们用 F 表示它在 $\mathbb{B}[y]$中的剩余类，其中 $\mathbb{B} = \mathbb{K}[x]/\langle D_{i+1} \cdots D_s \rangle$。取 j in $i+1, \ldots, s-1$，并假设我们已知 $Q_{i+1,i}, \ldots, Q_{j-1,i}$。将式（8）中的和拆分为 $A = Q_{j,i} H_j + R$，其中

$$A = G_i - y^{e_{i+1}} H_{i+1} - \sum_{k=i+1}^{j-1} Q_{k,i} H_k \quad \text{and} \quad R = \sum_{k=j+1}^{s} Q_{k,i} H_k.$$

在 $\mathbb{B}[y]$上，R 的次数（关于 y）小于 n_j；由于 H_j 是次数为 n_j的首一多项式，关系式 $A = Q_{j,i} H_j + R$ 描述了已知多项式 A 被同样已知的 H_j 除法的欧几里得除法。若令 $Q^*_{i,j}$ 为 $Q_{i,j}$ 在 $\mathbb{K}[x, y]$中的典范提升，则得到

$$\begin{aligned} Q^*_{j,i} &= Q_{j,i} \text{ rem } D_{i+1} \cdots D_s \\ &= (\sigma_{j,i} - q_{j,i} D_{i+1}) D_{i+2} \cdots D_j \text{ rem } D_{i+1} \cdots D_s. \end{aligned}$$

由此可知，$Q^*_{i,j}$可被 $D_{i+2} \cdots D_j$整除，并且

$$Q^*_{i,j} \text{ div } (D_{i+2} \cdots D_j) = (\sigma_{j,i} - q_{j,i} D_{i+1}) \text{ rem } D_{i+1} D_{j+1} \cdots D_s.$$

由于 $\deg(\sigma_{j,i}, x) < d_{i+1}$，将其模 D_{i+1} 化简后，最终得到 $\sigma_{j,i}$。另注意到余式 R 给出了 A的下一个值，因此我们得到算法 ParametersFromReducedBasis。

在下面的命题中，为下一小节的讨论作准备，我们特别指出，该算法不执行任何除法。

命题 6.2. 给定一个最小约化 Gröbner 基 $\mathcal{G} = (g_0, \ldots, g_s)$，在 $\mathbb{K}[x, y]$ 中，ParametersFromReducedBasis($\mathcal{G}$) 通过 $O\tilde{}(s^2 n_0 m_s)$ 次运算（$+, \times$）在 $\mathbb{K}$ 中返回 $\mathcal{G}$ 的 Gröbner 系数。

算法 6.1 ParametersFromReducedBasis($\mathcal{G}$)

Input: $\mathcal{G} = (g_0, \ldots, g_s)$ 属于 $\mathbb{K}[x, y]^s$

Assumptions: $\mathcal{G}$ 是一个极小约化 Gr¨obner 基，其首项 $(y^{n_0}, \ldots, x^{m_s})$ 按降序排列

Output: $(\lambda_1, \ldots, \lambda_N)$ 中的 $\mathbb{K}^N$

对于 $i = 0, \ldots, s$，执行 $x^{m_i} y^{n_i} \leftarrow$ InitialTerm(g_i)
$M_0 \leftarrow 1$, $G_0 \leftarrow g_0$
</style}针对 $i = 1$, $\ldots, s$ **do**
    $M_i \leftarrow$ PolynomialCoefficient(g_i, y^{n_i}) ▷ M_i 在 $\mathbb{K}[x]$ 中首一
    $G_i \leftarrow g_i$ 整除 M_i
    $D_i \leftarrow M_i$ 整除 M_{i-1} ▷ D_i 在 $\mathbb{K}[x]$ 中首一
    $n_{i-1,i-1} \leftarrow x^{d_i} - D_i$ ▷ $d_i = m_i - m_{i-1}$
$H_s \leftarrow 1$
对 $i = s - 1$, $\ldots, 0$ **do**
    $H_i \leftarrow y^{e_{i+1}} H_{i+1}$ ▷ $e_{i+1} = n_i - n_{i+1}$
    $\bar{A} \leftarrow \bar{G}_i - y^{e_{i+1}} \bar{H}_{i+1}$ ▷在 $\mathbb{B}[y]$ 中完成计算，使用 $\mathbb{B} = \mathbb{K}[x]/\langle D_{i+1} \cdots D_s \rangle$
    对 $j = i + 1$, $\ldots$, s 执行
        $\bar{Q}_{j,i} \leftarrow \bar{A}$ 整除 $\bar{H}_j$, $\bar{A} \leftarrow \bar{A}$ 余数 $\bar{H}_j$ ▷ 在 $\mathbb{B}[y]$中进行的欧几里得除法
        $Q^*_{j,\ i} \leftarrow$ 的典范提升 $\bar{Q}_{j,\ i}$ 至 $\mathbb{K}[x, y]$
        $\sigma_{j,i} \leftarrow (Q^*_{j,i}$ 整除 $D_{i+2} \cdots D_j)$ 余数 D_{i+1}
        $H_i \leftarrow H_i +$ KroneckerMultiply($\sigma_{j,i}, D_{i+2} \cdots D_j H_j$)
返回 ParametersFromSigma($(y^{n_0}, \ldots, x^{m_s})$, $(\sigma_{j,i})_{0 \le i \le s-1, i \le j \le s}$)

与之前一样，处理定点 Gr¨obner 胞所需的修改很简单；只需在最后一步调用 PunctualParametersFromSigma 。所得过程记作 PunctualParametersFromReducedBasis。在证明该命题之前，我们先举例说明定点 Gr¨obner 系数的计算。

例 6.3. 给定 $\mathcal{G}$ 如引言中所述，见 例 1.1，

$$\begin{aligned} &y^4 + \tfrac{17}{14} xy - \tfrac{17}{7} x^2, \\ &xy^3 - \tfrac{10}{9} x^3, \\ &x^2 y - 2x^3, \\ &x^4, \end{aligned}$$

算法 PunctualParametersFromReducedBasis计算

$$\begin{aligned} \sigma_{0,0} = \sigma_{1,0} = 0, \quad \sigma_{2,0} = \frac{17}{14}, \quad \sigma_{3,0} = \frac{40}{9}, \\ \sigma_{1,1} = \sigma_{2,1} = 0, \quad \sigma_{3,1} = -\frac{10}{9}, \\ \sigma_{2,2} = 0, \quad \sigma_{3,2} = -2x \end{aligned}$$

因此

$$\lambda_1 = 0, \quad \lambda_2 = \frac{17}{14}, \quad \lambda_3 = \frac{40}{9}, \quad \lambda_4 = -\frac{10}{9}, \quad \lambda_5 = -2. \tag{9}$$

证明。我们已经确立了算法的正确性。通过检查可以看出，所有步骤在 $\mathbb{K}$ 中仅涉及加法和乘法，并且只使用整数常数，因为所进行的操作仅是首一多项式的乘法或欧几里得除法，这些首一多项式或者属于 $\mathbb{K}[x]$，或者属于 $\mathbb{B}[y]$，其中 $\mathbb{B}$ 的形式为 $\mathbb{K}[x]/\langle D_{i+1}\cdots D_s\rangle$（而这又归结为 $\mathbb{K}$ 中的加法和乘法）。

接下来需要确定算法的运行时间。循环在第 3 行的每次迭代使用 $O^{\sim}(\delta)$ 次运算（$+$、$\times$），总计为 $O^{\sim}(s\delta)$。为了继续分析，我们首先注意到，对于所有 i，算法计算出的多项式的 x-次数小于 $d_{i+1}+\cdots+d_s$，而后者小于 m_s，并且 y-次数为 n_i。对于 $\deg(Q^*_{j,i}, x)$，同样的界也成立（根据构造）；如算法推导过程中所述，该多项式的 y-次数小于 e_j。

由于 G_i 满足与 H_i 相同的次数界 $\deg(G_i, x) < d_{i+1}+\cdots+d_s$，因此将 $G_i - y^{e_{i+1}}H_{i+1}$ 模 $D_{i+1}\cdots D_s$ 约化是免费的。在每次执行第 13 行时，欧几里得除法在 $\mathbb{B}$中需要 $O^{\sim}(n_{j-1}) \subset O^{\sim}(n_0)$ 次运算（$+,\times$），也就是在 $\mathbb{K}$ 中需要 $O^{\sim}(n_0 m_s)$ 次运算（$+,\times$）。上述次数界表明，计算 $\sigma_{j,i}$ 并更新H_i 的代价具有相同的上界 $O^{\sim}(n_0 m_s)$。由于我们在第 12 行进入内层循环 For 时，共执行 $O(s^2)$ 次，因此总成本为 $O^{\sim}(s^2 n_0 m_s)$。 □

现在我们来看看如何将这一观察形式化：算法 ParametersFromReducedBasis计算出的系数是 $\mathcal{G}$的系数的多项式表达式。

假设项 $\boldsymbol{E}$ 固定，令 $\mu_1, \ldots, \mu_\delta$ 为不属于 $\langle \boldsymbol{E}\rangle$的单项式，并按任意顺序排列；再令 $\Gamma_{0,1}, \ldots, \Gamma_{s,\delta}$ 为 $\mathbb{Z}$上的 $(s+1)\delta$ 个新变量。我们设定$\mathbb{A}_{\boldsymbol{E}} = \mathbb{Z}[\Gamma_{0,1}, \ldots, \Gamma_{s,\delta}]$。

由于该算法只进行加法和乘法，并使用来自 $\mathbb{Z}$ 在 $\mathbb{K}$ 中的像的常数，我们可以推断存在 $P_{1,\boldsymbol{E}}, \ldots, P_{N,\boldsymbol{E}}$ 属于 $\mathbb{A}_{\boldsymbol{E}} = \mathbb{Z}[\Gamma_{0,1}, \ldots, \Gamma_{s,\delta}]$，使得对于任意给定的、初始项为 $\boldsymbol{E}$ 且系数属于 $\mathbb{K}$ （或属于其任意扩张，如下文所选）的约化 Gröbner 基 $\mathcal{G} = (g_0, \ldots, g_s)$，$\mathcal{G}$ 的 Gröbner 参数可通过将 $P_{1,\boldsymbol{E}}, \ldots, P_{N,\boldsymbol{E}}$ 在 $\mathcal{G}$ 的系数处求值得到。

因此，算法的正确性可以重述如下。设 $\Lambda_1, \ldots, \Lambda_N$ 是 $\mathbb{K}$ 上另一组新的变量，用来表示“通用”的 Gröbner 参数，并定义 $\mathbb{L} = \mathbb{K}(\Lambda_1, \ldots, \Lambda_N)$。再设 $g_{0,\mathbb{L}}, \ldots, g_{s,\mathbb{L}}$ 是由 ReducedBasisFromParameters$(\boldsymbol{E}, (\Lambda_1, \ldots, \Lambda_N))$ 输出得到的多项式。由于该算法同样只执行加法和减法（命题 5.1），这些多项式的系数实际上属于 $\mathbb{K}[\Lambda_1, \ldots, \Lambda_N] \subset \mathbb{L}$。对于 $i = 0, \ldots, s$ 和 $j = 1, \ldots, \delta$，令 $R_{i,j} \in \mathbb{K}[\Lambda_1, \ldots, \Lambda_N]$ 表示单项式 μ_j 在 $g_{i,\mathbb{L}}$ 中的系数。根据上述讨论，我们可以推得，对于所有 i，$P_{i,\boldsymbol{E}}(R_{0,1}, \ldots, R_{s,\delta}) = \Lambda_i$ 成立。我们将在下一小节中使用这一观察结果。

6.2 Gröbner 参数的多项式方程

现在设 $f_1, \ldots, f_t$ 是 $\mathbb{K}[x, y]$ 中的多项式；在本小节中，它们是我们的输入，我们将它们生成的理想记为 J，其中 J 属于 $\mathbb{K}[x, y]$。再设 I 是 $\mathbb{K}[x, y]$ 中的一个理想，并满足以下性质：

A_1. I 的维数为零；

A_2. 存在一个理想 $I' \subset \mathbb{K}[x, y]$，使得 $I + I' = \langle 1 \rangle$ 且 $II' = J$。

等价地，I 是 J 的若干零维主分量的交（或乘积）。例如，当原点 (0, 0) 是 $V(J)$ 中的孤立点，且 I 是 J 的 $\langle x, y \rangle$-主分量时，或者当 $I = J$ 且 $V(J)$ 是有限集时，便属于这种情况。

设 $\mathcal{G} = (g_0, \ldots, g_s) \subset \mathbb{K}[x, y]$ 是 I 的约化字典序 Gröbner 基。我们用 $\boldsymbol{E}$ 表示 $\mathcal{G}$ 中多项式的首项，像之前一样写作

$$\boldsymbol{E} = (y^{n_0}, x^{m_1} y^{n_1}, \ldots, x^{m_{s-1}} y^{n_{s-1}}, x^{m_s}).$$

接下来，我们假设已知 $\boldsymbol{E}$，但未知 $\mathcal{G}$；我们将说明如何求出 I 的 Gröbner 参数（从而得到其本身 $\mathcal{G}$）。

令 δ 为 I 的次数，μ_1、……、μ_δ 为不属于 $\langle \boldsymbol{E} \rangle$ 的单项式，并以任意顺序排列。进一步令 $N = \delta + m_s$ 为 Gr¨obner 胞 $\mathcal{C}(\boldsymbol{E})$ 的参数个数，并令 $(\lambda_1, \ldots\ldots, \lambda_N) = \phi_{\boldsymbol{E}}^{-1}(I) \in \mathbb{K}^N$ 为与 I 相关联的 Gr¨obner 参数。在本小节中，我们定义一个由 $t\delta$ 个方程 $\mathscr{E}_{1,1}$、……、$\mathscr{E}_{t,\delta}$ 组成的方程组，其定义在 $\mathbb{K}[\Lambda_1, \ldots, \Lambda_N]$ 中，其中 Λ_1、……、Λ_N 是新变量；并证明 $(\lambda_1, \ldots\ldots, \lambda_N)$ 是这些方程的一个重数为 1 的解。

与上一小节一样，令 $\mathbb{L} = \mathbb{K}(\Lambda_1, \ldots\ldots, \Lambda_N)$，并令 $g_{0,\mathbb{L}}$、……、$g_{s,\mathbb{L}}$ 为由 ReducedBasisFromParameters$(\boldsymbol{E}, (\Lambda_1, \ldots\ldots, \Lambda_N))$ 给出的 $\mathcal{C}(\boldsymbol{E})$ 的参数 Gr¨obner 基。回忆一下，所有多项式 $g_{0,\mathbb{L}}$、……、$g_{s,\mathbb{L}}$ 的系数都属于 $\mathbb{K}[\Lambda_1, \ldots, \Lambda_N]$；特别地，这意味着对于 $\mathbb{K}[x, y]$ 中的 A，余式 A rem $\langle g_{0,\mathbb{L}}, \ldots\ldots, g_{s,\mathbb{L}} \rangle$ 属于 $\mathbb{K}[\Lambda_1, \ldots, \Lambda_N][x, y]$。对于 $j = 1$、……、δ，我们接着将 $\mathcal{N}_i$ 记作以下的 $\mathbb{K}$-线性映射：

$$\begin{aligned} \mathcal{N}_j : \mathbb{K}[x, y] &\to \mathbb{K}[\Lambda_1, \ldots, \Lambda_N] \\ A &\mapsto \operatorname{coeff}(A \text{ rem } \langle g_{0,\mathbb{L}}, \ldots, g_{s,\mathbb{L}} \rangle, \mu_j), \end{aligned}$$

其中 $\mu_1, \ldots, \mu_\delta$ 是上述定义的不属于 $\langle \boldsymbol{E} \rangle$ 的单项式。对于 $i = 1, \ldots, t$，我们接着令

$$\mathscr{E}_{i,1}, \ldots, \mathscr{E}_{i,\delta} = \mathcal{N}_1(f_i), \ldots, \mathcal{N}_\delta(f_i),$$

从而定义 $t\delta$ 多项式 $\mathscr{E}_{1,\,1}$，…，$\mathscr{E}_{t,\,\delta}$ 在 $\mathbb{K}[\Lambda_1, \ldots, \Lambda_N]$中。这些方程的一项关键性质受到 [37, 定理 4。8], 的启发，该定理是在边界基的背景下提出的。

Pr命题 6.4. $(\lambda_1, \ldots, \lambda_N)$ 是 $\mathscr{E}_{1,1}, \ldots, \mathscr{E}_{t,\delta}$的一个重数 1 的解 .

证明。令 $\mathcal{I}$ 为由所有多项式 $\mathcal{N}_i(g_j)$（其中 $i = 1, \ldots, \delta$ 且 $j = 0, \ldots, s$）生成的理想，并令 $R_{0,1}, \ldots, R_{s,\delta} \in \mathbb{K}[\Lambda_1, \ldots, \Lambda_N]$ 为 $(g_{0,\mathbb{L}}, \ldots, g_{s,\mathbb{L}})$ 的系数，与上一小节中一样。那么，对于 $i = 1, \ldots, \delta$ 和 $j = 0, \ldots, s$，多项式 $\mathcal{N}_i(g_j)$ 等于 $R_{j,i}(\lambda_1, \ldots, \lambda_N) - R_{j,i}$。特别地，$(\lambda_1, \ldots, \lambda_N)$ 属于 $\mathcal{I}$ 的零点集。

进一步回顾上一小节中的结论：存在多项式 $P_{1,\boldsymbol{E}}, \ldots, P_{N,\boldsymbol{E}}$，使得对于所有 k，均有 $P_{k,\boldsymbol{E}}(R_{0,1}, \ldots, R_{s,\delta}) = \Lambda_k$。对于所有$i$，$R_{j,i}(\lambda_1, \ldots, \lambda_N) - R_{j,i}$ 都属于 $\mathcal{I}$这一事实意味着

$$P_{k,\boldsymbol{E}}(R_{0,1}(\lambda_1, \ldots, \lambda_N), \ldots, R_{s,\delta}(\lambda_1, \ldots, \lambda_N)) - P_{k,\boldsymbol{E}}(R_{0,1}, \ldots, R_{s,\delta})$$

对于所有 $k = 1, \ldots, N$，它也属于 $\mathcal{I}$。左边是 λ_k，右边是 Λ_k，因此 $\mathcal{I}$包含所有多项式 $\Lambda_1 - \lambda_1, \ldots, \Lambda_N - \lambda_N$。综上，到目前为止的这两段推导表明 $\mathcal{I} = \langle \Lambda_1 - \lambda_1, \ldots, \Lambda_N - \lambda_N \rangle$。

现在令 $\mathcal{J}$ 为由多项式 $\mathscr{E}_{1,1}, \ldots, \mathscr{E}_{t,\delta}$ 在 $\mathbb{K}[\Lambda_1, \ldots, \Lambda_N]$ 中生成的理想。首先注意，对于任意 a，$b \geq 0$ 和 $i = 1, \ldots, t$，

$$(x^a y^b f_i) \text{ rem } \langle g_{0,\mathbb{L}}, \ldots, g_{s,\mathbb{L}} \rangle = \sum_{j=1}^{\delta} \mathcal{N}_j(f_i)(x^a y^b \mu_j \text{ rem } \langle g_{0,\mathbb{L}}, \ldots, g_{s,\mathbb{L}} \rangle).$$

由此可知，对于任意 A属于 $J = \langle f_1, \ldots, f_t \rangle$，且对于 $j = 1, \ldots, \delta$，$\mathcal{N}_j(A)$ 属于 $\mathcal{J}$。同理，对于 A属于 $I = \langle g_0, \ldots, g_s \rangle$，且对于 $j = 1, \ldots, \delta$，$\mathcal{N}_j(A)$ 属于 $\mathcal{I}$。我们还需要如下事实：对于 A属于 I^2，且对所有 j，$\mathcal{N}_j(A)$ 属于 $\mathcal{I}^2$；这可用类似的方法证明。

现在回顾关于 I' 的第二个假设：存在一个理想 $I' \subset \mathbb{K}[x, y]$，使得$I + I' = \langle 1 \rangle$ 且 $II' = J$。由于 J 包含于 I，前一段中的论述意味着 $\mathcal{J}$ 包含于 $\mathcal{I} = \langle \Lambda_1 - \lambda_1, \ldots, \Lambda_N - \lambda_N \rangle$，因此 $(\lambda_1, \ldots, \lambda_N)$ 属于 $\mathcal{J}$ 的零点集。这就证明了命题的第一个断言。

进一步设 K, K' 分别属于 I 和 I'，且满足 $K + K' = 1$。对于 $i = 0, \ldots, s$，g_i 属于 I，因此 $g_i K' = g_i - g_i K$ 属于 $II' = J$。由上面的注释可知，对于 $j = 1, \ldots, \delta$，$A_{j,i} := \mathcal{N}_j(g_i) - \mathcal{N}_j(g_i K)$ 因而属于 $\mathcal{J}$，而 $\mathcal{N}_j(g_i K)$ 属于 $\mathcal{I}^2$。

考虑所有多项式 $A_{j,i}$ 在 $(\lambda_1, \ldots, \lambda_N)$ 处的雅可比矩阵 $\boldsymbol{J}$。由于所有项 $\mathcal{N}_j(g_i K)$ 都属于 $\mathcal{I}^2 = \langle \Lambda_1 - \lambda_1, \ldots, \Lambda_N - \lambda_N \rangle^2$，它们的雅可比矩阵在 $(\lambda_1, \ldots, \lambda_N)$ 处为零，因此 $\boldsymbol{J}$ 就是多项式 $\mathcal{N}_j(g_i)$ 在 $(\lambda_1, \ldots, \lambda_N)$ 处的雅可比矩阵。由于这些多项式生成理想 $\mathcal{I} = \langle \Lambda_1 - \lambda_1, \ldots, \Lambda_N - \lambda_N \rangle$，该矩阵的核是平凡的。因此，$\mathcal{J}$ 在 $(\lambda_1, \ldots, \lambda_N)$ 处的重数为 1。 □

在 $I = J$ 的特殊情形下，我们可以得到一个稍强的结果。

推论 6.5. 假设 $I = \langle f_1, \ldots, f_t \rangle$。那么，$\langle \mathscr{E}_{1,1}, \ldots, \mathscr{E}_{t,\delta} \rangle = \langle \Lambda_1 - \lambda_1, \ldots, \Lambda_N - \lambda_N \rangle$属于 $\mathbb{K}[\Lambda_1, \ldots, \Lambda_N]$。

证明。利用命题证明中的记号，我们可知，若 $I = J$，则 $\mathcal{I} = \mathcal{J}$，而我们已经证明了 $\mathcal{I} = \langle \Lambda_1 - \lambda_1, \ldots, \Lambda_N - \lambda_N \rangle$。 □

在另一种特殊情形下，若 I是 J的 $\langle x, y \rangle$-主分量，我们可以得到类似但更强的结论。回忆一下，局部 Gr¨obner 胞 $\mathcal{C}_0(\boldsymbol{E})$ 的维数为

$N' = \delta - n_0$，并且 $\mathcal{C}_0(\boldsymbol{E})$的参数可通过将 $\mathcal{C}(\boldsymbol{E})$的参数 $\Lambda_1, \ldots, \Lambda_N$中的 $N - N'$个参数设为零得到。

令 $\tau_1, \ldots, \tau_{N-N'}$为这些被设为零的参数的指标，并令 $\Lambda_{\sigma_1}, \ldots, \Lambda_{\sigma_{N'}}$为其余的 N'个参数。对于 $i = 1, \ldots, t$和 $j = 1, \ldots, \delta$，令 $\mathscr{F}_{i,\ j}$为在 $\mathscr{E}_{i,\ j}$中将 $\Lambda_{\tau_1}, \ldots, \Lambda_{\tau_{N-N'}}$设为零后得到的 $\mathbb{K}[\Lambda_{\sigma_1}, \ldots, \Lambda_{\sigma_{N'}}]$中的多项式。于是有如下结果。

推论 6.6. 假设 I是 $\langle x, y\rangle$-准素的。那么，$\langle \mathscr{F}_{1,\ 1}, \ldots, \mathscr{F}_{t,\ \delta}\rangle = \langle \Lambda_{\sigma_1} - \lambda_{\sigma_1}, \ldots, \Lambda_{\sigma_{N'}} - \lambda_{\sigma_{N'}}\rangle$属于 $\mathbb{K}[\Lambda_{\sigma_1}, \ldots, \Lambda_{\sigma_{N'}}]$。

证明。 我们在命题 6.4 中证明了 $\lambda_{\sigma_1}, \ldots, \lambda_{\sigma_{N'}}$是 $\mathscr{F}_{1,\ 1}, \ldots, \mathscr{F}_{t,\ \delta}$的一个解。此外，由于 $\mathscr{E}_{1,\ 1}, \ldots, \mathscr{E}_{t,\ \delta}$的雅可比矩阵在 $(\lambda_1, \ldots, \lambda_N)$ 处的核平凡（因此 $\lambda_{\tau_1} = \cdots = \lambda_{\tau_{N-N'}} = 0$），对于 $\mathscr{F}_{1,\ 1}$，……，$\mathscr{F}_{t,\ \delta}$在 $(\lambda_{\sigma_1}, \ldots, \lambda_{\sigma_{N'}})$ 处的情形也同样如此。因此，唯一缺少的性质是 $(\lambda_{\sigma_1}, \ldots, \lambda_{\sigma_{N'}})$ 是这些方程的唯一公共解。令 $(\lambda^\star_{\sigma_1}, \ldots, \lambda^\star_{\sigma_{N'}}) \in \overline{\mathbb{K}}^{N'}$为这样的一个解，令 $\mathcal{G}^\star$为相应的约化 Gr¨obner 基，并令 $I^\star$为它生成的理想（特别地，$V(I^\star) = \{(0, 0)\}$）。由于根据假设，$\mathcal{G}^\star$将 f_1，……，f_t约化为零，因此有 $J \subset I^\star$。

根据关于 I的假设，存在一个理想 $I' \subset \mathbb{K}[x, y]$，使得 $I + I' = \langle 1\rangle$且 $II' = J$。令 K、K'分别属于 I 和 I'，并满足 $K + K' = 1$；特别地，K' 在 $(0, 0)$ 处不为零。由于 $V(I^\star) = \{(0, 0)\}$，可知 K' 是模 $I^\star$的单位。

回顾一下，我们用 $\mathcal{G} = (g_0, \ldots, g_s)$ 表示 I 的约化字典序 Gröbner 基。那么，对于 $i = 0, \ldots, s$，多项式 $g_i K'$ 属于 II'，因此属于 J，进而属于 $I^\star$。由于 K' 模 $I^\star$ 是单位，这意味着 g_i 属于 $I^\star$。综上，这证明了 I 包含于 $I^\star$。由于这些理想的字典序初始理想相同，因此它们相等。由此又可证明 $(\lambda_{\sigma_1}, \ldots, \lambda_{\sigma_{N'}}) = (\lambda^\star_{\sigma_1}, \ldots, \lambda^\star_{\sigma_{N'}})$。 □

例子。在贯穿全文的例子中，我们只考虑局部的 Gr¨obner 胞，并取 f_1 和 f_2，如引言中所述。为了写出局部 Gr¨obner 参数的方程，我们考虑 $g_{0,\ \mathbb{L}}, \ldots, g_{3,\ \mathbb{L}}$并将上文写出的参数 $\Lambda_{\tau_1}, \ldots, \Lambda_{\tau_{N-N'}}$设为零；所得多项式见式（5），它们以变量 $\lambda_1, \ldots, \lambda_5$表示（回顾一下，此处$N' = 5$）。用这些多项式约化 f_1 和 f_2，并取系数（舍去恒等于零的系数），得到

$$14\Lambda_1, \quad 14\Lambda_2 - 17, \quad -14\Lambda_1\Lambda_5^2 + 14\Lambda_3 - 28\Lambda_4\Lambda_5, \quad 14\Lambda_2\Lambda_5 + 34, \quad -18\Lambda_4 + 10\Lambda_5. \quad (10)$$

如前所述，这些多项式生成该极大理想

$$\Lambda_1, \quad \Lambda_2 - 17/14, \quad \Lambda_3 - 40/9, \quad \Lambda_4 + 10/9, \quad \Lambda_5 + 2.$$

由于输入 f_1，f_2 和 $\mathcal{G}$ 的次数都很小，方程组（10）可以手工求解。当然，一般情况下并非如此（不过，在许多例子中，其中几个方程确实是线性的）。

7 牛顿迭代

最后，我们可以描述主算法，该算法使用牛顿迭代计算 Gr¨obner 参数。为此，我们假设 $\mathbb{K}$ 是整环 $\mathbb{A}$ 的分式域，并考虑 $\mathbb{A}$ 中的极大理想 $\mathfrak{m}$，其剩余域为 $\Bbbk = \mathbb{A}/\mathfrak{m}$。

考虑以下对象：$\mathbb{A}[x,y]$ 中的多项式（$f_1, \ldots, f_t$）以及 $\mathbb{K}[x,y]$中的约化 Gr¨obner 基 $\mathcal{G}$。我们作如下假设：

A′1. 由 $\mathbb{K}[x,y]$ 中的 $\mathcal{G}$ 生成的理想是 $\langle f_1, \ldots, f_t\rangle$的若干主分量的交集，

A′2. $\mathcal{G}$ 中的所有多项式都属于 $\mathbb{A}_{\mathfrak{m}}[x,y]$，

A′3. 由 $\mathcal{G}_{\mathfrak{m}} = \mathcal{G}$ 余 $\mathfrak{m}$ 在$\Bbbk[x,y]$ 中生成的理想是理想 $\langle f_1$ 余 $\mathfrak{m}, \ldots, f_t$ 余 $\mathfrak{m}\rangle$的若干主分量的交集。

最后两项表明 $\mathfrak{m}$ 对 $\mathcal{G}$是良好的，符合定义1.2的含义。满足第一和第三个假设的重要情形与上一小节相同，即当 $\mathcal{G}_{\mathfrak{m}}$和 $\mathcal{G}$生成理想 $\langle f_1$ 对 $\mathfrak{m}$取余，....., f_t 对 $\mathfrak{m}\rangle$取余，分别为 $\langle f_1$, ..., $f_t\rangle$ 这些理想本身时；或者它们描述这些理想的 $\langle x, y\rangle$-主分量时。

给定 $\mathfrak{m}$、$(f_1, \ldots, f_t)$ 和 $\mathcal{G}_{\mathfrak{m}}$，这里我们将展示如何对任意 $K \geq 1$ 计算 $\mathcal{G}$对 $\mathfrak{m}^K$ 取余。

算法 LiftOneStep描述了核心提升过程；它以 $\mathcal{G}$ 的 Gr¨obner 参数（已知模 $\mathfrak{m}^\kappa$，其中 $\kappa \geq 0$ 为某个参数）为输入，并返回这些参数模 $\mathfrak{m}^{2\kappa}$ 的结果（注意，由于根据 A′2，$\mathcal{G}$ 的系数属于 $\mathbb{A}_{\mathfrak{m}}$，其 Gr¨obner 参数也属于 $\mathbb{A}_{\mathfrak{m}}$，因此对它们模 $\mathfrak{m}$ 的幂取余是有意义的）。

该算法只需对上一小节中介绍的方程 $\mathscr{E}_{i,j}$ 应用牛顿迭代。不过请注意，我们从不显式写出这些方程，因为它们可能包含大量项；相反，我们只计算其一阶泰勒展开，因为这足以进行迭代。这也解释了为什么下文的运算是在理想 $\langle \Lambda_1, \ldots, \Lambda_N\rangle^2$ 模意义下进行的。

由于我们希望给出一个统计 $\mathbb{A}_{2\kappa}$ 中运算次数的成本估计，因此这里假设我们已经知道输入方程 $f_1, \ldots, f_t$ 模 $\mathfrak{m}^{2\kappa}$ 的约化结果；这些结果记作 $f'_1, \ldots, f'_t \in \mathbb{A}_{2\kappa}[x,y]$。算法中的某些步骤需要进一步说明，即第 ReducedBasisFromParameters 5 行的调用 5、Reduction 第 8 行的调用以及 LinearSolve第 行的调用。

- 在第 5 行，我们处理的是写作 (¨obner 参数，$\ell_1, \ldots, \ell_N$)，它们属于 $\mathbb{B} = \mathbb{A}_{2\kappa}[\Lambda_1, \ldots, \Lambda_N]/\langle \Lambda_1, \ldots, \Lambda_N\rangle^2$(在算法中，$\mathbb{B}$ 的元素写作 $b_0 + \sum_{i=1}^n b_i\Lambda_i$，其中某些 b_i 属于 $\mathbb{A}_{2\kappa}$)。回想一下，算法 ReducedBasisFromPa-rameters只进行加法和乘法，并使用来自 $\mathbb{Z}$ 的常数，因此我们可以将该算法用于输入属于 $\mathbb{B}$ 的情形；不过要记住，它的性质只针对域中的输入得到了证明。

  同样的说明也适用于第 8 行，对于算法 ReductionGeneralInput。

- 最后一个子程序用于求解 $\mathbb{A}_{2\kappa}$ 上的线性方程组：输入是 $\mathbb{B}$ 的元素。回顾一下，这些元素的形式为 $b_0 + \sum_{i=1}^{n} b_i \Lambda_i$，其中某些 b_i 属于 $\mathbb{A}_{2\kappa}$。过程 LinearSolve 随后会发现，这些元素是关于 Λ_i 的线性方程。我们将证明解存在，并且相应的矩阵有一个模 $\mathfrak{m}$ 不为零的极大子式，因此解实际上是唯一的。

算法 7.1 LiftOneStep$((f'_1, \ldots, f'_t)\ \boldsymbol{E}, (\alpha_1, \ldots, \alpha_N))$

Input: $(f'_1, \ldots, f'_t)$ 属于 $\mathbb{A}_{2\kappa}[x, y]$，单项式 $\boldsymbol{E}$，$(\alpha_1, \ldots, \alpha_N)$ 属于 $\mathbb{A}_{\kappa}^{N}$

Output: $(\alpha''_1, \ldots, \alpha''_N)$ 在 $\mathbb{A}_{2\kappa}^{N}$

将 $(\alpha_1, \ldots, \alpha_N)$ ←提升至 $\mathbb{A}_{2\kappa}^{N}$ 的 $(\alpha'_1, \ldots, \alpha'_N)$

$\mu_1, \ldots, \mu_\delta$ ← 中不属于 $\langle \boldsymbol{E} \rangle$ 的单项式

对于 $i = 1, \ldots, N$ **do**

    $\ell_i \leftarrow \alpha'_i + \Lambda_i$ ▷所有 ℓ_i 属于 $\mathbb{B} = \mathbb{A}_{2\kappa}[\Lambda_1, \ldots, \Lambda_N]/\langle \Lambda_1, \ldots, \Lambda_N \rangle^2$

$\mathcal{G}^* \leftarrow$ ReducedBasisFromParameters$(\boldsymbol{E}, (\ell_1, \ldots, \ell_N))$ ▷在 $\mathbb{B}$ 上完成的计算

$\mathscr{R} \leftarrow [\,]$

对于 $i = 1, \ldots, t$ **do**

    $r_i \leftarrow$ ReductionGeneralInput$(f'_i, \mathcal{G}^*)$ ▷在 $\mathbb{B}$上完成的计算

    对于 $j = 1$、……、δ，执行$r_{i,\,j} \leftarrow$ 系数(r_i, μ_j) ▷ 全部$r_{i,\,j}$ 在 $\mathbb{B}$中

    $\mathscr{R} \leftarrow \mathscr{R}$ 连接 $[r_{i,1}, \ldots, r_{i,\delta}]$ ▷ $\mathscr{R}$ 是一个元素属于 $\mathbb{B}$的数组

$(\epsilon_1, \ldots, \epsilon_N) \leftarrow$ LinearSolve$(\mathscr{R})$ ▷ 所有 ϵ_i 属于 $\mathbb{A}_{2\kappa}$

对于 $i = 1, \ldots, N$执行 $\alpha''_i \leftarrow \alpha'_i + \epsilon_i$ ▷ 所有 α''_i属于 $\mathbb{A}_{2\kappa}$

返回 $(\alpha''_1, \ldots, \alpha''_N)$

命题 7.1。假设 A'_1、A'_2 和 A'_3成立，并令 $(\lambda_1, \ldots, \lambda_N) \in \mathbb{A}_{\mathfrak{m}}^{N}$为 $\mathcal{G}$ 的 Gr¨obner 参数。给定 $(f_1, \ldots, f_t)$ 余数 $\mathfrak{m}^{2\kappa}$ 以及 $(\lambda_1$ 余数 $\mathfrak{m}^{\kappa}, \ldots, \lambda_N$余数 $\mathfrak{m}^{\kappa})$，算法LiftOneStep 正确返回 $(\lambda_1$ 余数 $\mathfrak{m}^{2\kappa}, \ldots, \lambda_N$余数 $\mathfrak{m}^{2\kappa})$。

证明。设 $\lambda = (\lambda_1, \ldots, \lambda_N) \in \mathbb{A}_{\mathfrak{m}}^{N}$ 是与 $\mathcal{G}$ 相关联的 Gröbner 参数。根据假设，向量 $\alpha = (\alpha_1, \ldots, \alpha_N)$ 满足 $\alpha = \lambda$ 余数 $\mathfrak{m}^{\kappa}$，α' 也同样满足。我们证明，输出 $\alpha'' = (\alpha''_1, \ldots, \alpha''_N)$ 等于 λ 余数 $\mathfrak{m}^{2\kappa}$。

这正是牛顿迭代有效性的经典证明。设 δ 为次数 $\mathcal{G}$，并设 $\mathscr{E} = (\mathscr{E}_{1,\,1}$、……、$\mathscr{E}_{t,\,\delta})$ 是上一小节中针对多项式 f_1、……、f_t 和 $\mathcal{G}$在域 $\mathbb{K}$上的方程。由于所有 f_i 的系数都属于 $\mathbb{A}$，且约化过程不会引入新的分母，多项式 $\mathscr{E}$ 属于$\mathbb{A}[\Lambda_1, \ldots, \Lambda_N]$。利用命题 6.4，假设 A'_1 表明 λ 是这些方程的一个解（并且它处的雅可比矩阵 λ 的核为平凡核，但我们不会直接用到这一事实）。

进一步令 $\mathscr{E}_{\mathfrak{m}} = (\mathscr{E}_{\mathfrak{m},1,1}, \ldots, \mathscr{E}_{\mathfrak{m},t,\delta})$ 表示这些相同的方程，但这次针对多项式 f_1 余数 $\mathfrak{m}, \ldots, f_t$ 余数 $\mathfrak{m}$和 $\mathcal{G}_{\mathfrak{m}}$。这些是 $\mathsf{k}[\Lambda_1, \ldots, \Lambda_N]$ 中的多项式，其中 $\mathscr{E}_{\mathfrak{m}} = \mathscr{E}$余数 $\mathfrak{m}$。利用命题 6.4，假设 A'_3 表明 λ余数 $\mathfrak{m}$ 是一个解

这些方程（我们已经可以从前一段推导出这一点），并且它们在 λ 余数 $\mathfrak{m}$ 处的雅可比矩阵的核是平凡的。我们将在下文中用到这一点。

我们声称，对于所有 i, j，在第 9 行计算出的系数 $r_{i,j}$等于在 $\mathbb{B} = \mathbb{A}_{2\kappa}[\Lambda_1, \ldots, \Lambda_N]/\langle \Lambda_1, \ldots, \Lambda_N \rangle^2$ 中计算出的 $\mathscr{E}_{i,j}(\ell_1, \ldots, \ell_N)$。唯一需要注意的是，算法 ReducedBasisFromParameters的输出仅在输入属于域时才被规定为 Gröbner 基。为处理这一点，令 $\ell'_1, \ldots, \ell'_N$为 $\ell_1, \ldots, \ell_N$在整环 $\mathbb{A}[\Lambda_1, \ldots, \Lambda_N]$中的任意提升，并令 $\mathcal{G}'$ 为 ReducedBasisFromParameters$(\boldsymbol{E}, (\ell'_1, \ldots, \ell'_N))$的输出。这些多项式构成了 $\mathbb{K}(\Lambda_1, \ldots, \Lambda_N)[x, y]$中的 Gröbner 基；该基的所有系数恰好都属于 $\mathbb{A}[\Lambda_1, \ldots, \Lambda_N]$，而在第 5 行计算出的$\mathcal{G}^*$是 $\mathcal{G}'$模 $\mathfrak{m}^{2\kappa} + \langle \Lambda_1, \ldots, \Lambda_N \rangle^2$的约化。

同样，在第 8 行，算法 ReductionGeneralInput可以将系数属于 $\mathbb{B}$ 的多项式作为输入，但其输出仅针对系数属于某个域的多项式进行了规定。按照之前的处理方式可得，对于所有 i，r_i 是多项式 f_i除以$\mathfrak{m}^{2\kappa} + \langle \Lambda_1$、……、$\Lambda_N \rangle^2$所得的余数。现在，$f_i$除以 $\mathcal{G}'$所得的余数的系数是多项式 $\mathscr{E}_{i,\ j}$在 $(\ell'_1, \ldots, \ell'_N)$ 处的取值，因此总而言之，对于所有 i，j，$r_{i,\ j} = \mathscr{E}_{i,\ j}(\ell_1, \ldots, \ell_N)$，作为 $\mathbb{B}$ 的元素。将所有 i、j 一并考虑，我们得到以下在 $\mathbb{B}$ 上成立的等式：

$$\begin{aligned}\mathscr{R} &= \mathscr{E}(\alpha'_1 + \Lambda_1, \ldots, \alpha'_N + \Lambda_N) \\ &= \mathscr{E}(\alpha') + \mathrm{jac}(\mathscr{E}, \alpha')[\Lambda_1 \ \cdots \ \Lambda_N]^T,\end{aligned}$$

其中，$\mathrm{jac}(\mathscr{E}, \alpha')$ 是在 α' 处取值的 $\mathscr{E}$ 的雅可比矩阵。首先，我们证明线性方程组 $\mathscr{R}$ 在 $\mathbb{A}_{2\kappa}^N$ 中有唯一解 $\epsilon = (\epsilon_1, \ldots, \epsilon_N)$。事实上，给定 $\mathbb{A}_{2\kappa}^N$ 中的两个解向量 ϵ 和 ϵ'，我们得到关系式

$$\mathrm{jac}(\mathscr{E}, \alpha')[\epsilon_1 - \epsilon'_1 \ \cdots \ \epsilon_N - \epsilon'_N]^T = [0 \ \cdots \ 0]^T$$

在 $\mathbb{A}_{2\kappa}$ 上。我们上面指出，$\mathrm{jac}(\mathscr{E}$ 余数 $\mathfrak{m}, \lambda$ 余数 $\mathfrak{m})$ 的核是平凡的，因此它在 $\mathbb{K} = \mathbb{A}/\mathfrak{m} = \mathbb{A}_{2\kappa}/\mathfrak{m}$ 中存在一个非零的 N-阶子式。现在，根据假设，α' 余数 $\mathfrak{m} = \lambda$ 余数 $\mathfrak{m}$，所以 $\mathrm{jac}(\mathscr{E}, \alpha')$ 本身存在一个模 $\mathfrak{m}$ 可逆的 N-阶子式，因而在 $\mathbb{A}_{2\kappa}$ 中也可逆。这又意味着 $\epsilon = \epsilon'$，作为 $\mathbb{A}'/\mathfrak{m}^{2\kappa}$ 上的向量。我们的第一个论断得证。

其次，我们证明 $\epsilon = (\lambda - \alpha')$ 余数 $\mathfrak{m}^{2\kappa}$ 是这些线性方程的一个解。事实上，模 $\mathfrak{m}^{2\kappa}$，我们有泰勒展开 $\mathscr{E}(\alpha' + \epsilon) = \mathscr{E}(\alpha') + \mathrm{jac}(\mathscr{E}, \alpha')[\epsilon_1 \ \cdots \ \epsilon_N]^T$：高阶项消失，因为根据假设，$\epsilon$ 的所有元素都在 $\mathfrak{m}^\kappa$ 中。现在，$\alpha' + \epsilon = \lambda$ 余数 $\mathfrak{m}^{2\kappa}$，所以 $\mathscr{E}(\alpha' + \epsilon) = 0$ 余数 $\mathfrak{m}^{2\kappa}$，我们的论断得证。

前两段证明了，在 while 循环结束时，值 α'' 满足$\alpha'' = \alpha' + (\lambda - \alpha')$ 余数 $\mathfrak{m}^{2\kappa} = \lambda$ 余数 $\mathfrak{m}^{2\kappa}$，因此证明完成。□

命题 7.2。设 $\boldsymbol{E} = (y^{n_0}, x^{m_1}y^{n_1}, \ldots, x^{m_{s-1}}y^{n_{s-1}}, x^{m_s})$ 是 $\mathcal{G}$ 的首项，并假设所有 f_i 的次数至多为 d。

在假设 A′1、A′2、A′3 下，算法 LiftOneStep 在 $\mathbb{A}_{2\kappa}$ 中使用 $O\tilde{}(s^2 n_0 m_s + t\delta(d^2 + dm_s + s\delta + \delta^{\omega-1}))$ 次运算。

证明。按照惯例（见引言），将每个 α_i 提升到 α'_i 在 $\mathbb{A}_{2\kappa}$中需要进行一次运算，因此总共需要 $O(N) = O(\delta)$ 次运算。

根据命题 5.1，计算 $\mathcal{G}^*$ 需要 $O^{\sim}(s^2 n_0 m_s)$ 次运算（$+$，$\times$）在 $\mathbb{B}$中，其中每次此类运算在 $\mathbb{A}_{2\kappa}$ 中需要 $O(\delta)$ 次运算。

在第 8 行，根据命题 4.6，算法 ReductionGeneralInput 使用 $O^{\sim}(d^2+dm_s+n_0m_s+s\delta)$ 次运算（$+$，$\times$），在 $\mathbb{B}$ 中。这里，我们知道 n_0 至多为 d，因此所有 f_i 的运行时间变为 $O^{\sim}(t(d^2+dm_s+s\delta))$ 次运算，在 $\mathbb{B}$ 中，即 $O^{\sim}(t\delta(d^2+dm_s+s\delta))$ 次运算，在 $\mathbb{A}_{2\kappa}$ 中。

最后，我们需要求解由 $\mathscr{R}=0$ 在 $\mathbb{A}_{2\kappa}$ 上定义的线性方程组。该方程组有 $t\delta$ 个方程和 N 个未知数，并且我们知道它在 $\mathbb{A}_{2\kappa}^N$ 中有唯一解，因为相应的矩阵模 $\mathfrak{m}$ 的核是平凡的。尽管 $\mathbb{A}_{2\kappa}^N$ 不是域，我们仍然可以应用快速算法，例如 [55] (，在 [56] 中扩展)，以可逆性测试替代零测试；这需要 $O^{\sim}(t\delta^{\omega})$ 次运算，在 $\mathbb{A}_{2\kappa}$ 中。 □

与往常一样，如果 $\mathcal{G}$（因此 $\mathcal{G}_{\mathfrak{m}}$）是 $\langle x,\ y\rangle$-主理想，我们可以使用这种提升过程的一个变体，称为 LiftOneStepPunctualParameters，它将 ReducedBasisFromPunctualParameters 和 PunctualParametersFromReducedBasis作为子程序。这样我们就可以处理 N'，而非 N 未知的 Gr¨obner 参数；现在的证明依赖于推论 6.6，运行时间则为 $O^{\sim}(s^2n_0m_s+t\delta^2(m_s+\delta^{\omega-2}))$ 次运算，位于 $\mathbb{A}_{2\kappa}$（见命题 4.6）。

此时，定理 1.4 的证明几乎已经完成：对于 $K=2^k$，该算法只需通过反复调用算法 LiftOneStep 来计算 $\mathcal{G}$ rem $\mathfrak{m}^K$。然而，这一过程以 Gr¨obner 参数作为输入和输出。因此，在首次进入算法 LiftOneStep 之前，我们先计算 $\mathcal{G}$ rem $\mathfrak{m}$ 的 Gr¨obner 参数；而在最后一次调用算法 LiftOneStep之后，我们再使用算法 ReducedBasisFromParameters计算 $\mathcal{G}$ rem $\mathfrak{m}^K$。这项额外工作不会影响渐近运行时间，因此，对于 $i=1,\dots,k$，我们需要在 $\mathbb{A}/\mathfrak{m}^{2^i}$ 中执行 $O^{\sim}(s^2n_0m_s+t\delta(d^2+dm_s+s\delta))$ 次运算。

目前尚未计入的唯一运算，是对多项式进行逐系数约化 f_1、……、f_t 模 $\mathfrak{m}^2$、……、$\mathfrak{m}^{2^k}$。这些运算无法用剩余类环 $\mathbb{A}/\mathfrak{m}^{2^i}$ 中的运算来表示；相反，按照引言中的约定，我们假设每次将系数模 $\mathfrak{m}^{2^i}$ 约化所需的时间为T_{2^i}，因此对每个$td^2T_{2^i}$、……、$i=1$，k总共需要 $\mathfrak{m}^{2^k}$。由此，主定理的证明告一段落。在处理逐点 Gr¨obner 胞时，我们在命题4.6 中看到，每个输入多项式只需要 δm_s 个系数，因此，对所有指标 $t\delta m_sT_{2^i}$，系数约化需要 i步。

注 7.3. 如果希望只以 Gr ¨obner 基作为输入和输出，设计名为 LiftOneStepGroebnerBasis（以及 LiftOn-eStepPunctualGroebnerBasis）的算法很简单：这些算法以 f_1'、……、f_t' 和 $\mathcal{G}$ mod $\mathfrak{m}^{\kappa}$ 为输入，并返回 $\mathcal{G}$ mod $\mathfrak{m}^{2\kappa}$。只需在进入该过程时调用算法 ParametersFromReducedBasis，接着调用算法 LiftOneStep，最后在退出前调用算法 Reduced-BasisFromParameters（或相应的逐点变体）即可。这不会影响渐近运行时间，但对我们的主定理而言并无用处。

注 7.4。当 $\mathfrak{m}$ 为主理想时，我们可以通过使用分治技术（众所周知）或松弛算法，稍微改进提升过程 *[57, 第 4] 节中的方法来求解*

给出 $\epsilon_1, \ldots, \epsilon_N$ 的线性方程组。其缺点是，运行时间不再以 $\mathbb{A}_{2\kappa}$ 中的运算次数表示。取而代之的是，我们给出常见情形 $\mathbb{A} = \mathbb{Z}$ 和 $\mathfrak{m} = \langle p \rangle$，以及 $\mathbb{A} = \mathbb{K}[t]$ 和 $\mathfrak{m} = \langle t - \tau \rangle$ 的运行时间：

- 在前一种情况下，求解该系统需要 $O\tilde{}(t\delta^{\omega}\log(p))$ 次位运算，这是对模 p 进行的一次性计算（矩阵求逆）；随后在模 $p^{2\kappa}$ 下求解该系统需要 $O\tilde{}(\delta^2\kappa\log(p))$ 次位运算。

- 在后一种情况下，一次性计算需要在 $\mathbb{K}$ 中进行 $O\tilde{}(t\delta^{\omega})$ 次运算，之后求解线性方程组需要在 $\mathbb{K}$ 中进行 $O\tilde{}(\delta^2\kappa)$ 次运算。

也就是说，在前一种情况下，每次在 $\mathbb{A}_{2\kappa}$ 中进行的运算，如命题 7.2 所述，需要 $O\tilde{}(\kappa \log(p))$ 次位运算；而在后一种情况下，需要 $O\tilde{}(\kappa\log(p))$ 次运算。总体而言，在这两种情况下，求解线性方程组的成本都可以忽略不计（除了我们在开始时进行的一次性计算）。

例 7.5。 我们以正在使用的例子为例，展示算法的一步（例 1.1），重点考察 punctual Gr¨obner 参数。我们的输入是多项式 f_1、f_2，如引言中所述；此外还包括 ¨obner 基 $\langle x$，它是 $y\rangle$-*primary component of* $\langle f_1 \operatorname{rem} p$、$f_2 \operatorname{rem} p\rangle$ 的，且 $p = 11$；即：

$$\left|\begin{array}{l} y^4 + 2xy + 7x^2, \\ xy^3 + 5x^3, \\ x^2y + 9x^3, \\ x^4. \end{array}\right.$$

我们推导出模 11 下的点式 Gr¨obner 参数，$\alpha = (0, 2, 2, 5, 9) \in \mathbb{Z}/11\mathbb{Z}^5$（回想一下，$N' = 5$）。按照算法，我们令 $(\ell_1, \ldots, \ell_5) = (\Lambda_1, 2+\Lambda_2, 2+\Lambda_3, 5+\Lambda_4, 9+\Lambda_5)$，并计算相应的点式 Gr¨obner 基，系数截断于模 11^2 和 $\langle \Lambda_1, \ldots, \Lambda_N \rangle^2$。我们得到伪代码中写出的多项式：

$$\left|\begin{array}{l} y^4 + \Lambda_1 xy^2 + (\Lambda_2 + 2)xy + (40\Lambda_1 + \Lambda_3 + 103\Lambda_4 + 111\Lambda_5 + 33)x^3 + (9\Lambda_2 + 2\Lambda_5 + 18)x^2, \\ xy^3 + (\Lambda_4 + 5)x^3, \\ x^2y + (\Lambda_5 + 9)x^3, \\ x^4. \end{array}\right.$$

将 f_1 和 f_2 模 $\mathcal{G}^*$ 约化（计算模 11^2 和 $\langle \Lambda_1, \ldots, \Lambda_N \rangle^2$），并保留系数，可得线性方程 $\mathscr{R}$（我们只展示非零项）

$$14\Lambda_1 = 14\Lambda_2 + 11 = 76\Lambda_1 + 14\Lambda_3 + 111\Lambda_4 + 102\Lambda_5 + 99 = 5\Lambda_2 + 28\Lambda_5 + 44 = 103\Lambda_4 + 10\Lambda_5 = 0.$$

它们模 11^2 有如下唯一解：

$$\epsilon_1 = 0,\ \epsilon_2 = 77,\ \epsilon_3 = 110,\ \epsilon_4 = 88,\ \epsilon_5 = 110;$$

正如预期，所有 ϵ_i 都模 11 消失。由此，α 更新为取值 $\alpha + \epsilon = [0, 79, 112, 93, 119]$ 模 11^2。可以验证，这与 11^2 中给出的值模 9 一致。

8 结论

一个自然的问题是，我们的方法是否可以用于多于两个变量的理想。目前仍缺少若干要素：已知的结构结果不如 Lazard 的 [58], 完整，而且目前没有 Gr¨obner 胞的已知显式描述。在算法方面，关键操作（模一个 n元字典序 Gr¨obner 基进行约化）本身似乎就是一个颇具挑战性的问题。

如引言中所述，利用我们的结果来恢复 $\mathcal{G}$ 本身，而非 $\mathcal{G}$ 余数 $\mathfrak{m}^K$，尤其包括对坏极大理想 $\mathfrak{m}$进行量化，是未来工作的主题。除此之外，我们希望实现的主要算法改进是降低总体成本，使其在两种方法均适用的情况下与 [9], 的成本相匹配。

9 致谢

作者感谢：加拿大自然科学与工程研究委员会（探索研究基金和亚历山大·格雷厄姆·贝尔加拿大研究生奖学金）、魁北克自然与技术研究基金以及滑铁卢大学给予的慷慨支持。

参考文献

[1] L. Gonz´alez-Vega，M. E. Kahoui，实代数平面曲线拓扑计算的改进复杂度上界，《复杂性杂志》12 (4) (1996) 527 – 544。[2] I. Z. Emiris，E. P. Tsigaridas，二元多项式系统的实数求解，载于：CASC，施普林格，2005，第 150–161 页。[3] D. Diochnos，I. Emiris，E. Tsigaridas，关于实数域上求解双变量系统的渐近复杂度与实际复杂度，《符号计算杂志》44 (7) (2009) 818–835。[4] L. Alberti，B. Mourrain，J. Wintz，半代数平面曲线的拓扑与排列计算，《计算机辅助几何设计》25 (8) (2008) 631–651。[5] F. Rouillier，关于求解二元多项式系统，载于：ICMS，《计算机科学讲义》卷 6327，施普林格，2010，第 100–104 页。[6] E. Berberich，P. Emeliyanenko，M. Sagraloff，一种求解二元多项式系统的消元法：消除常见缺点，载于：ALENEX，SIAM，2011，第 35–47 页。[7] P. Emeliyanenko，M. Sagraloff，关于求解二元多项式系统的复杂度，载于：ISSAC'12，ACM，2012，第 154–161 页。

[8] Y. Bouzidi，S. Lazard，M. Pouget，F. Rouillier，双变量系统的有理单变量表示及其应用，载于：ISSAC'13，ACM，2013，第 109–116 页。[9] R. Lebreton，E. Mehrabi，´E. Schost，关于求解双变量系统的复杂度：非奇异解的情形，载于：ISSAC'13，ACM，2013，第 251–258 页。[10] Y. Bouzidi，S. Lazard，G. Moroz，M. Pouget，F. Rouillier，计算双变量系统分离线性形式的改进算法，载于：ISSAC'14，ACM，2014，第 75–82 页。[11] A. Kobel，M. Sagraloff，计算平面代数曲线的改进复杂度界，CoRR abs/1401.5690 (2014)。[12]E. Mehrabi，E. Schost，一种用于求解整数域上二元多项式系统的软最优蒙特卡罗算法，《复杂性杂志》34 (2016) 78–128。doi:https://doi.org/10.1016/j.jco.2015.11.009.[13]A. Kobel，M. Sagraloff，关于计算平面代数曲线的复杂度，《复杂性杂志》31 (2) (2015) 206–236。doi:https://doi.org/10.1016/j.jco.2014.08.002.[14] Y. Bouzidi，S. Lazard，G. Moroz，M. Pouget，F. Rouillier，M. Sagraloff，使用有理单变量表示求解双变量系统，《复杂性杂志》37 (2016) 34–75。 doi:https://doi.org/10.1016/j.jco.2016.07.002.[15] D. N. Diatta，S. Diatta，F. Rouillier，M.-F. Roy，M. Sagraloff，代数数上的多项式界及其在曲线拓扑中的应用 (2021)。 arXiv:1807.10622.[16]X. Dahan，二元多项式关于一个单变量多项式取模的字典序 Gr¨obner 基，《符号计算杂志》110 (2022) 24–65。doi:https://doi.org/10.1016/j.jsc.2021.10.001.[17] F. Rouillier，通过有理单变量表示求解零维系统，《工程、通信与计算中的应用代数》9 (5) (1999) 433–461。[18] G. L. Ebert，关于 Gr¨obner 基的模方法的一些评论，ACM SIGSAM 公报 17 (2) (1983) 28–32。[19] W. Trinks，利用牛顿法改进布赫贝格算法的近似结果，SIGSAM 公报 18 (3) (1984) 7–11。 doi:10.1145/1089389.1089392.[20] M. Giusti，J. Heintz，J.-E. Morais，L.-M. Pardo，多项式方程组何时能够快速求解？，载于：AAECC-11，《LNCS》卷 948，施普林格，1995，第 205–231 页。[21] M. Giusti，J. Heintz，J.-E. Morais，J. Morgenstern，L.-M. Pardo，几何消元理论中的直线程序，《纯粹与应用代数杂志》124 (1998) 101–146。

[22] M. Giusti, K. H¨agele, J. Heintz, J.-E. Morais, J.-L. Monta˜na, L.-M. Pardo，丢番图逼近的下界，《纯粹与应用代数杂志》117/118 (1997) 277–317。[23] M. Giusti, G. Lecerf, B. Salvy，一种无需 Gr¨obner 基的多项式系统求解替代方法，《复杂性杂志》17 (1) (2001) 154–211。[24] ´E. Schost，参数化几何分解的计算，《工程、通信与计算中的应用代数》13 (5) (2003) 349–393。[25] F. Winkler，一种基于 A p-进的方法，用于计算 Gr¨obner 基，《符号计算杂志》6 (2/3) (1988) 287–304。doi:10.1016/S0747-7171(88)80049-X.[26] F. Pauer，关于 Gr¨obner 基计算中的幸运理想，《符号计算杂志》14 (5) (1992) 471–482。[27] E. A. Arnold，计算 Gr¨obner 基的模算法，《符号计算杂志》35 (4) (2003) 403–419。[28] T. Ojika, S. Watanabe, T. Mitsui，非线性方程组重根的降亏算法，《数学分析与应用杂志》96 (2) (1983) 463–479。 doi:https://doi.org/10.1016/0022-247X(83)90055-0. URL https://www.sciencedirect.com/science/article/pii/0022247X83900550[29] N. Yamamoto，具有奇异雅可比矩阵的非线性方程解的正则化，《信息处理杂志》7 (1984) 16–21。[30] G. Lecerf，具有重数系统的二次牛顿迭代，《计算数学基础》2 (3) (2002) 247–293。[31] A. Leykin, J. Verschelde, A. Zhao，牛顿法结合降亏算法求解多项式系统的孤立奇点，《理论计算机科学》359 (1) (2006) 111–122。doi:https://doi.org/10.1016/j.tcs.2006.02.018.[32] A. Leykin, J. Verschelde, A. Zhao，具有孤立奇异解的多项式系统的高阶降亏，载于：代数几何中的算法，施普林格，2008，第 79–97 页。[33]S. Pope, A. Szanto，具有给定根重数的最近多元系统，《符号计算杂志》44 (6) (2009) 606–625。doi:https://doi.org/10.1016/j.jsc.2008.03.005.[34] B. Dayton, T.-Y. Li, Z. Zeng，非线性系统的多重零点，《计算数学》80 (276) (2011) 2143–2168。[35]A. Mantzaflaris, B. Mourrain，多项式系统奇异零点的降亏与经认证的隔离，载于：ISSAC'11，ACM Press，2011，第 249–256 页。doi:10.1145/1993886.1993925.

[36]X. Wu, L. Zhi，通过符号–数值约化为几何对合形式确定多项式系统的奇异解，《符号计算杂志》47 (3) (2012) 227–238。doi:https://doi.org/10.1016/j.jsc.2011.10.001.[37]J. Hauenstein, B. Mourrain, A. Szanto，关于降亏与重数结构，《符号计算杂志》83 (2017) 228–253，ISSAC 2015 会议专刊：符号计算与计算机代数。doi:https://doi.org/10.1016/j.jsc.2016.11.013.[38] A. Conca, G. Valla，整数环的理想的典范希尔伯特–伯奇矩阵，《密歇根数学杂志》57 (2008) 157–172。 doi:10.1307/mmj/1220879402. URL https://doi.org/10.1307/mmj/1220879402[39] J. van der Hoeven, R. Larrieu，关于充分正则 Gr¨obner 基的二元多项式快速约化，载于：ISSAC '18，ACM Press，2018，第 199–206 页。doi:10.1145/3208976.3209003.[40] D. Lazard，理想基与初等分解：二变量情形，《符号计算杂志》1 (3) (1985) 261–270。[41] J. von zur Gathen, J. Gerhard，《现代计算机代数》，第 3 版，剑桥大学出版社，2013。[42] S. G. Hyun, S. Melczer, E. Schost, C. St-Pierre，一元与二元整数环-主理想的基变换，载于：ISSAC' 19，ACM Press，2019，第 227–234 页。[43] J. Canny, E. Kaltofen, Y. Lakshman，更快地求解非线性多项式方程组，载于：ISSAC'89，ACM，1989，第 121–128 页。[44] J. van der Hoeven, E. Schost，高维多点求值，《工程、通信与计算中的应用代数》24 (1) (2013) 37–52。[45] J. Brian¸con, A. Galligo，点的特定变形，载于：《卡尔热兹的奇点》，《Ast´erisque》第 7–8 号，法国数学学会，1973，第 129–138 页。[46] J. Brian¸con，《Hilb^n $\mathbb{C}\{x,y\}$的描述》，《Inventiones Mathematicae》41 (1977) 45–90。[47] A. Iarrobino，《支撑于一点的希尔伯特概形》，《美国数学学会回忆录》第 188 号，美国数学学会，1977。[48] G. Carr`a Ferro，Gr¨obner 基与希尔伯特概形（一），《符号计算杂志》6 (2) (1988) 219–230。doi:https://doi.org/10.1016/S0747-7171(88)80044-0.[49] R. Notari, M. L. Spreafico，按初始理想对希尔伯特概形进行分层及其应用，《Manuscripta Mathematica》101 (2000) 429–448。

[50] L. Robbiano，关于边界基和 Gr¨obner 基概形，Collectanea Mathematica 60 (2009) 11–25. doi:10.1007/BF03191213.[51] P. Lella, M. Roggero，希尔伯特概形的有理分支，Rend. Semin. Mat. Univ. Padova 126 (2011) 11–45. doi:10.4171/RSMUP/126-2.[52] G. Ellingsrud, S. Strømme，关于平面上点的希尔伯特概形的同调，Inventiones Mathematicae 87 (1987) 343–352. doi:10.1007/BF01389419.[53] K. Kedlaya, C. Umans，快速多项式因式分解与模复合，SIAM计算杂志 40 (6) (2011) 1767–1802. doi:10.1137/08073408X.[54]V. Neiger, B. Salvy, E. Schost, G. Villard，更快的模复合（2021）。doi:10.48550/ARXIV.2110.08354。URL https://arxiv.org/abs/2110.08354[55] O. H. Ibarra, S. Moran, R. Hui，快速 LUP 矩阵分解算法的推广及其应用，J. Algorithms 3 (1) (1982) 45–56.[56] C.-P. Jeannerod，重新审视 LSP 矩阵分解（2006）。[57] J. Berthomieu, R. Lebreton，代数系统的松弛 p-进 Hensel 提升，载于：ISSAC'12，ACM，2012，第 59–66 页。[58] M. G. Marinari, T. Mora，Cerlienco-Mureddu对应与Lazard结构定理，Investigaci´on Operacional 27 (1) (2013) 75–98。