

基于Groebner基的单项式生成理想仿射代数簇维数计算

马小茸

天津职业技术师范大学理学院, 天津

收稿日期: 2025年1月10日; 录用日期: 2025年2月11日; 发布日期: 2025年2月26日

摘要

研究代数簇的维数可以帮助深入理解代数结构与几何对象。论文研究了从单项式生成理想所对应代数簇的维数计算, 通过找出单项式生成理想的Groebner基, 并在此基础上探讨多种单项式理想对应簇的维数, 得出一个计算单项式生成理想簇的维数的公式, 并利用Maple软件进行算法实现, 得到计算代数簇维数的一种方法。

关键词

单项式生成理想, Groebner基, 代数簇, 维数, Maple

Groebner-Based Monomial Generation of Ideal Affine Algebraic Variety Dimension Calculations

Xiaorong Ma

School of Science, Tianjin University of Technology and Education, Tianjin

Received: Jan. 10th, 2025; accepted: Feb. 11th, 2025; published: Feb. 26th, 2025

Abstract

Studying the dimension of algebraic varieties aids in gaining a deeper understanding of algebraic structures and geometric objects. This paper investigates the computation of the dimension of algebraic varieties corresponding to ideals generated by monomials. By identifying the Groebner basis of the ideal generated by monomials, the paper further explores the dimensions of varieties corresponding to various monomial ideals. It derives a formula for calculating the dimension of the

文章引用: 马小茸. 基于 Groebner 基的单项式生成理想仿射代数簇维数计算[J]. 理论数学, 2025, 15(2): 98-103.

DOI: 10.12677/pm.2025.152051

variety corresponding to an ideal generated by monomials and implements the algorithm using Maple software, thereby providing a method for computing the dimension of algebraic varieties.

Keywords

Monomial Generation Ideals, Groebner Basis, Algebraic Variety, Dimensionality, Maple

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

代数簇作为在代数几何这一数学领域中的研究对象,从 18 世纪代数与数学分析交织在一起,到 20 世纪代数簇的探究取得一定成果[1] [2],其维数的计算问题一直被数学家关注。

随着理想生成元的复杂程度的提升,学者对代数簇维数的研究也日益深入。在前面的研究中,学者计算理想簇的维数问题是从希尔伯特函数或者 Krull 维数着手[3] [4],这需要大量的理论知识以及复杂的计算。本研究从单项式生成理想这一基础概念出发,通过计算 Groebner 基[5],探讨单项式生成理想簇的维数问题。通过深入分析这些理想簇的代数性质,期望能够揭示出它们之间的内在联系和规律,从而为代数簇维数的计算提供新的思路和方法。

本文的主要内容分为三个部分,第一部分内容为基础理论知识[6] [7],为后续研究奠定基础。第二部分内容为给出代数簇维数的公式,第三部分内容在第二部分内容基础上用 Maple 进行算法实现。

2. 基础代数知识

定义 2.1 在 $K[x_1, \dots, x_n]$ 中,称幂积

$$x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdots x_n^{\alpha_n}$$

为关于 $x_1, \dots, x_n$ 的项,简记 x^α 。其中 x 和 α 分别表示 $(x_1, \dots, x_n)$, $(\alpha_1, \dots, \alpha_n)$, 又称 α_i 为 x^α 关于 x_i 的次 $\deg(x^\alpha, x_i)$, 而 $\alpha_1 + \alpha_2 + \cdots + \alpha_n$ 为 x^α 的全次数,记为 $\text{tdeg}(x^\alpha)$ 。

定义 2.2 在 $K[x_1, \dots, x_n]$ 中称有限和

$$f = \sum_{\alpha} a_{\alpha} x^{\alpha}, a_{\alpha} \in K$$

为多项式。其中 a_{α} 是 f 关于项 x^{α} 的系数,记为 $\text{coef}(f, x^{\alpha})$, 若 $a_{\alpha} \neq 0$, 则称 x^{α} 为 f 的项,并称 $a_{\alpha} x^{\alpha}$ 为 f 的单项式。

定义 2.3 若 K 是一个域,多项式 $f_1, \dots, f_s \in K[x_1, \dots, x_n]$, 称

$$V\langle f_1, \dots, f_s \rangle = \{(a_1, \dots, a_n) \in K^n \mid f_i(a_1, \dots, a_n) = 0, 1 \leq i \leq s\}$$

是由 $f_1, \dots, f_s$ 定义的仿射代数簇。

定义 2.4 在 $K[x_1, \dots, x_n]$ 的一个单项式 x^{α} , $\alpha \in \mathbb{Z}_{\geq 0}^n$, $K[x_1, \dots, x_n]$ 中的一个单项式 $>$ 指的是定义在 $\mathbb{Z}_{\geq 0}^n$ 上的一种关系,它满足:

- (1) $>$ 是在 $\mathbb{Z}_{\geq 0}^n$ 的一个全序
- (2) 如果 $\alpha > \beta$ 并且 $\gamma \in \mathbb{Z}_{\geq 0}^n$ 那么有 $\alpha + \gamma > \beta + \gamma$

(3) $>$ 是 $\mathbb{Z}_{\geq 0}^n$ 上的良序

设 $f = \sum_{\alpha} a_{\alpha} x^{\alpha}$ 是在 $K[x_1, \dots, x_n]$ 下的一个非零多项式, $>$ 是一个单项式序, 则可以定义 f 的多重次数为 $\text{multideg}(f) = \max\{\alpha \in \mathbb{Z}_{\geq 0}^n : a_{\alpha} \neq 0\}$, 首项系数为 $LC(f) = a_{\text{multideg}(f)} \in K$, 首单项式为 $LM(f) = x^{\text{multideg}(f)}$, 首项为 $LT(f) = LC(f) \cdot LM(f)$ 。

除非另外说明, 本文默认变元序 $x_1 < x_2 < \dots < x_n$, 下面介绍 $K[x_1, \dots, x_n]$ 上几种常见的全序关系, 设 $x^{\alpha} = x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdot \dots \cdot x_n^{\alpha_n}$, $x^{\beta} = x_1^{\beta_1} \cdot x_2^{\beta_2} \cdot \dots \cdot x_n^{\beta_n}$, 定义

字典序: $x^{\alpha} <_{\text{lex}} x^{\beta}$, 如果存在 $i(1 \leq i \leq n)$, 使得

$$\alpha_j = \beta_j (i+1 \leq j \leq n) \text{ 且 } \alpha_i < \beta_i$$

逆字典序: $x^{\alpha} <_{\text{rlex}} x^{\beta}$, 如果存在 $i(1 \leq i \leq n)$, 使得

$$\alpha_j = \beta_j (i+1 \leq j \leq n) \text{ 且 } \alpha_i > \beta_i$$

分次字典序: $x^{\alpha} <_{\text{grlex}} x^{\beta}$, 如果

$$\text{tdeg}(x^{\alpha}) < \text{tdeg}(x^{\beta}), \text{ 或者 } \text{tdeg}(x^{\alpha}) = \text{tdeg}(x^{\beta}) \text{ 且 } x^{\alpha} <_{\text{lex}} x^{\beta}$$

分次逆字典序: $x^{\alpha} <_{\text{grevlex}} x^{\beta}$, 如果

$$\text{tdeg}(x^{\alpha}) < \text{tdeg}(x^{\beta}), \text{ 或者 } \text{tdeg}(x^{\alpha}) = \text{tdeg}(x^{\beta}) \text{ 且 } x^{\alpha} <_{\text{rlex}} x^{\beta}$$

定义 2.5 在多项式环 $K[x_1, \dots, x_n]$ 上固定一个单项式序, 有一组多项式 $g_1, \dots, g_t$, $g_i = \sum a_{\alpha_i} x^{\alpha_i}$, $a_{\alpha_i} \in K, 1 \leq i \leq t$, 若 $\langle LT(g_1), \dots, LT(g_t) \rangle = \langle LT(I) \rangle$ 成立, 理想 $I \subseteq K[x_1, \dots, x_n]$ 不同于 $\{0\}$ 的有限生成子集 $G = \{g_1, \dots, g_t\}$ 定义为 Groebner 基。

定义 2.6 K 是一个域, $V \subseteq K^n$ 是仿射代数簇, $I = I(V) \subset K[x_1, x_2, \dots, x_n]$, V 的坐标环是商环, 有下面的形式

$$K[V] = K[x_1, x_2, \dots, x_n] / I$$

坐标环也被称为 V 上的正则函数环。

若 M 是一个集合的集合, 子集 C 是 M 的子集被称为 M 中的链, 满足完全按照包含关系 “ $\subseteq$ ” 排序。 C 的长度定义为 $\text{length}(C) = |C| - 1 \in \mathbb{N}_0 \cup \{-1, \infty\}$, 长度为 n 的有限链通常写为

$$X_0 \subsetneq X_1 \subsetneq \dots \subsetneq X_n$$

记为

$$\text{length}(M) = \sup\{\text{length}(C) | C \text{ 是 } M \text{ 中的链}\}$$

若 $M = \emptyset$, 则长度为 -1 。

定义 2.7 对于一个给定的向量空间 V , 其 Krull 维数是满足以下条件的最长子空间链的长度, 即存在一个子空间链 $V_0 \subset V_1 \subset \dots \subset V_n = V$, 其中每个 V 都是 V_i 的子空间, 且 V_{i+1}/V_i 是非零向量空间, n 即为 V 的 Krull 维数。

$$\dim(V) = \text{length}(\{U \subseteq V | U \text{ 是子空间}\})$$

K 是一个代数闭域, K^n 中的子集 X 具有 Zariski 拓扑特点, X 是一个仿射代数簇, 有下式成立

$$\dim(X) = \dim(K[X])$$

在引出定理 3.1 之前, 首先进行如下说明:

考虑理想 $I \subseteq K[x_1, \dots, x_n]$, 集合 $G = \{g_1, g_2, \dots, g_m\}$ 是理想 I 的一个 Groebner 基。如果 G 中的所有单项式都包含相同的变元(即每个单项式都至少包含一个变元 $x_i \in \{x_1, x_2, \dots, x_n\}, 1 \leq i \leq n$), 已知理想 I 不是零理想, 则集合 $A = \{x | x_i = 0\} \subseteq V(I)$, 所以 $\dim A = n-1$, 由定义 $\dim V(I) = n-1$, 所以单项式生成理想代数簇的维数 $\dim V(I) = n-1$ 。

3. 维数公式

定理 3.1 若 G 中的 m 个单项式 ($m \leq n$) 不全部包含相同的变元, 即存在至少一个变元 $x_i \in \{x_1, x_2, \dots, x_n\}, 1 \leq i \leq n$, 使得 $x_i \notin m_j$ 对所有 $1 \leq j \leq m$ 成立, 则由此单项式理想 I 生成的理想簇 $V(I)$ 的维数为 $n-r$ 。

证明: 在 G 的单项式中找到出现次数最多的变元, 记为 x_{i_1} 。有 $A_1 = \{\bar{x}_i | x_{i_1} = 0\}$, $A_2 = \{\bar{x}_i | x_{i_1} = 0, x_{i_2} = 0\}$, $\dots$, $A_r = \{\bar{x}_i | x_{i_1} = 0, x_{i_2} = 0, \dots, x_{i_r} = 0\}$, $A_1 \supsetneq A_2 \supsetneq \dots \supsetneq A_r$, $A_r \subseteq V(I)$, 因为结合 Krull 维数有 $\dim V(I) = \dim(K[V(I)])$, 即表示为不在点集 $V(I)$ 中变元个数 $\dim A_r = n-r$, $\dim V(I) = n-r$ 。

根据上面的证明得到计算单项式生成理想代数簇的维数步骤, 步骤如下:

- (1) 找到 G 中的单项式中出现次数最多的变元。若变元不止一个, 则根据字典序选择第一个变元。
- (2) 将包含该变元的所有单项式组成一组。
- (3) 在余下的单项式中重复步骤(1)和(2), 直到所有单项式被分组, 最终得到 r 组单项式, 得到单项式生成理想代数簇 $V(I)$ 的维数为: $\dim V(I) = n-r$ 。

结合上面定理, 可以把理想 $I \subseteq K[x_1, \dots, x_n]$ 的 Groebner 基整理得到如下形式:

$$G = \left\{ \underbrace{x_1^{p_{11}} y_{11}, \dots, x_1^{p_{1q_1}} y_{1q_1}}_{c_1 \text{ 个单项式含有共同变元 } x_1}, \dots, \underbrace{x_r^{p_{r1}} y_{r1}, \dots, x_r^{p_{rc_r}} y_{rc_r}}_{c_r \text{ 个单项式含有共同变元 } x_r} \right\} \left(c_1 > c_2 > \dots > c_r \right)$$

符号说明:

- (1) 上式中的 y_i ($1 \leq i \leq r$) 是不含有变元 x_i 的单项式
- (2) p_i 是变元 x_i 的次数

4. 算法与示例

算法 4.1: 单项式生成理想簇维数

输入: 单项式序列 monos

输出: monos 序列中单项式生成理想簇维数

- 1: 初始化 iterationCount, singleVarCount, dd 为 0
- 2: 计算 monos 中变元的总数 nn
- 3: 使用 Groebner[Basis] 计算 monos 的基 monoss
- 4: 初始化结果列表 resultt 为空
- 5: **Step 1: 判断每一个变元在 Groebner 基中变元出现的次数是否为 1**
- 6: 确定 expr 的变量集合 vvars
- 7: **if** nops(vvars) = 1

续表

```

8:   set isInOther is false
9: Step 2: 去除 Groebner 基中变元出现一次单变元单项式的, 计数并更新单项式序列
10:  if has(monoss[j], vvar)
11:    set isInOther is true
12:    resultt := [op(resultt), expr]
13:  break
14:    if isInOther is false:
15:      singleVarCount := singleVarCount + 1
16:    else
17:      resultt := [op(resultt), expr]
18:    filteredMonos := resultt
19: Step 3: 遍历 filteredMonos 中的单项式并根据规则重复操作, 更新列表直至序列个数为 0
20:  iterationCount := iterationCount + 1
21:  varOccurrences := table([ ])
22: Step 3.1: 对 filteredMonos 中的变量出现次数进行分组和排序
23:  确定单项式的变量集合
24:  更新 varOccurrences 中每个变量的出现次数
25:  flag_set := {temp1[2], op(flag_set)}
26:  astVar := final_list[-1][1]
27:  filteredMonosdd := [ ]
28: Step 3.2: 去掉出现次数最多的变量所在的单项式
29:  if not has(mono, lastVar):
30:    filteredMonosdd := [op(filteredMonosdd), mono]
31:  filteredMonos := filteredMonosdd
32:    if nops(filteredMonos)=0 :
33:    break
34:  计算 dd = nn - iterationCount - singleVarCount
35: return dd
36: end

```

根据算法下面举出具体的例子进行实现, 在此之前首先对例子进行分析, 计算该单项式生成理想仿射代数簇的维数, 最后用 Maple 进行实现与验证。

例 4.1 在 $K[x_1, x_2, x_3, x_4, x_5, x_6, x_7]$ 中, 有理想 $I = \langle x_1^2 x_3, x_3^4 x_5, x_2 x_3, x_3^3, x_4 x_5, x_4^2 x_6 x_2^5, x_7^3 \rangle$, 其 Groebner 基为 $\{x_7^3, x_4 x_5, x_3^3, x_2 x_3, x_4^2 x_6 x_2^5, x_1^2 x_3\}$, 计算该理想代数簇的维数。

$$\begin{aligned}
V(I) &= V(x_7^3) \cap V(x_4 x_5) \cap V(x_3^3) \cap V(x_2 x_3) \cap V(x_4^2 x_6 x_2^5) \cap V(x_1^3 x_3) \\
&= V(x_7^3) \cap V(x_4^2 x_6 x_2^5) \cap V(x_4 x_5) \cap V(x_3^3) \cap V(x_2 x_3) \cap V(x_1^3 x_3) \\
&= H_{x_7} \cap (H_{x_4} \cup H_{x_6} \cup H_{x_2}) \cap (H_{x_4} \cup H_{x_5}) \cap H_{x_3} \cap (H_{x_2} \cup H_{x_3}) \cap (H_{x_1} \cup H_{x_3}) \\
&= H_{x_7} \cap (H_{x_4} \cup (H_{x_6} \cap H_{x_2} \cap H_{x_5})) \cap (H_{x_3} \cup (H_{x_2} \cap H_{x_3} \cap H_{x_1})) \\
&= H_{x_7} \cap (H_{x_4} \cup H_{x_2 x_5 x_6}) \cap (H_{x_3} \cup H_{x_1 x_2 x_3})
\end{aligned}$$

解集 $V(I)$ 可以表示为多个超平面的交集，具体表达形式为 $V(I) = H_1 \cap H_2 \cap \cdots \cap H_r$ 。其中，超平面 H_1, H_2, H_3 的定义分别为 $H_1 = H_{x_7}$ ， $H_2 = H_{x_4} \cup H_{x_2 x_5 x_6}$ ， $H_3 = H_{x_3} \cup H_{x_1 x_2 x_3}$ 。由定理 3.1 可知 $n=7, r=3$ ，则该单项式生成的理想代数簇的维数 $\dim V(I) = 7 - 3 = 4$ 。

下面运用算法计算例 4.1 的维数。

> RemoveSingleVarMonomials(monos)

维数, 4

5. 结论

本研究在 Groebner 基的基础上，通过研究不同单项式生成理想的所带来的代数簇维数的变化，并总结成定理，提出了一个用于计算单项式生成理想的代数簇维数公式。并利用数学计算软件 Maple 对公式进行了算法实现。

参考文献

- [1] Hartshorne, R. (1977) Algebraic Geometry. Springer. <https://doi.org/10.1007/978-1-4757-3849-0>
- [2] Atiyah, M.F. and MacDonald, I.G. (1969) Introduction To Commutative Algebra. CRC Press.
- [3] Cox, D., Little, J. and O'Shea, D. (1992) Ideals, Varieties, and Algorithms. Springer. <https://doi.org/10.1007/978-1-4757-2181-2>
- [4] 富庆辉. 代数簇维数计算的自动化方法研究[D]: [硕士学位论文]. 天津: 天津职业技术师范大学, 2022.
- [5] Traverso, C. (1996) Hilbert Functions and the Buchberger Algorithm. *Journal of Symbolic Computation*, **22**, 355-376. <https://doi.org/10.1006/jsco.1996.0056>
- [6] Kreuzer, M. and Robbiano, L. (2000) Computational Commutative Algebra. Springer. <https://doi.org/10.1007/978-3-540-70628-1>
- [7] 王东明, 牟晨琪, 李晓亮, 等. 多项式代数[M]. 北京: 高等教育出版社, 2011.