

一类仿射簇的维数计算的程序实现

富庆辉

天津职业技术师范大学理学院, 天津

收稿日期: 2022年10月7日; 录用日期: 2022年11月1日; 发布日期: 2022年11月9日

摘要

计算单项式理想定义的簇的维数与计算任意簇的维数不同, 可以简化为计算理想中最小公共变量集的元素个数。根据单项式理想的这一特性, 本文给出一种算法并利用符号计算软件进行实现, 从而可以自动计算单项式理想定义的簇的维数。

关键词

单项式理想, 仿射簇, 代数集, 维数

Symbolic Implementation of Calculation Dimension of a Class of Affine Varieties

Qinghui Fu

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

Received: Oct. 7th, 2022; accepted: Nov. 1st, 2022; published: Nov. 9th, 2022

Abstract

Calculating the dimension of varieties defined by a monomial ideal is different from calculating the dimension of any varieties, and can be reduced to finding the number of elements of the least common set of variables in the ideal. According to this characteristic of monomials, an algorithm is designed and a code is written by Maple to calculate the dimension of varieties defined by monomialideal.

Keywords

Monomialideal, Affinevarieties, Algebraic Set, Dimension



1. 引言

簇的维数是代数几何中的一个基本概念, 簇的一种刻画. 但是计算一个簇的维数是比较困难的, 其中计算由单项式理想定义的簇的维数是比较简单的. D. Cox 等人在[1]中, 总结了一种计算由单项式理想定义的簇的维数的方法. J. Herzog 在[2]的第二部分中介绍了希尔伯特函数, 可以用来计算簇的维数. K. Hulek 在[3]中介绍了希尔伯特零点定理、克鲁尔维数的概念. 这篇文章总结文献[1] [2] [3]中提到由单项式理想定义的簇的维数问题, 并且介绍一种算法, 用来计算由单项式理想定义的簇的维数. 最后利用 maple 实现程序, 并且通过实例论证整个过程是可以由计算机软件实现.

2. 基础知识

我们定义数域 k 上的 n 维仿射空间为:

$$A^n := \{(a_1, a_2, a_3, \dots, a_n) \mid \forall a_i \in k\}.$$

定义 1: 设子集 $X \in A^n$, 那么 X 生成的消失理想如下:

$$I(X) = \{f \in k[x_1, x_2, \dots, x_n] \mid \forall P \in X, f(P) = 0\}.$$

可以证明上述定义的 $I(X)$ 为 $k[x_1, x_2, \dots, x_n]$ 中的一个理想.

定义 2: 设 $k[x_1, x_2, \dots, x_n]$ 是在数域 k 的 n 个元多项式环, 对于环中的一个子集 J , 定义由 J 生成的簇 (代数集):

$$V(J) = \{P \in A^n \mid \forall f \in J, f(P) = 0\}.$$

根据上面的定义以及希尔伯特零点定理[3], 可以在 $k[x_1, x_2, \dots, x_n]$ 中的理想与 A^n 中子集定义出一个映射. 那么在这个映射下, $k[x_1, x_2, \dots, x_n]$ 中的素理想与 A^n 不可约簇是一一对应的[4].

定理 1: 假设 $A = \langle f_1, f_2, \dots, f_s \rangle$ 和 $B = \langle g_1, g_2, \dots, g_t \rangle$ 是 $k[x_1, x_2, \dots, x_n]$ 中的理想, $V(A)$ 和 $V(B)$ 是 A^n 中的仿射簇, 那么两个簇的交集和并集也是簇, 并且

$$V(A) \cup V(B) = V(f_1 g_1 \mid 1 \leq i \leq s, 1 \leq j \leq t),$$

$$V(A) \cap V(B) = V(\langle f_1, f_2, \dots, f_s, g_1, g_2, \dots, g_t \rangle).$$

定理 2: 假设簇 X 非空, 其对应的消去理想是 $I(X)$, 那么 X 是不可约簇当且仅当 $I(X)$ 是素理想.

其中不可约簇是指不存在 $X = X_1 \cup X_2$, 其中 X_1, X_2 是 X 的两个真子集, $I(X)$ 素理想是指, 若 $a \cdot b \in I(X)$, 那么必有 $a \in I(X)$ 或 $b \in I(X)$ 成立.

根据文献 4 定理 3.9 在 $k[x_1, x_2, \dots, x_n]$ 中素谱的诺特性质, 和素理想和不可约簇的对应关系, 可以证明每个由不可约簇构成的降链具有有限长度. 就像计算线性空间的维数一样. 这里将降链的最大长度定义为簇的维数.

定义 3: 一个簇 V 的克鲁尔维数(krull)是 V 的最长降链的长度 $\text{krdim } V = n$. 其中降链是

$$V = V_0 \supsetneq V_1 \supsetneq V_2 \supsetneq V_3 \supsetneq \dots \supsetneq V_n \neq \emptyset.$$

根据上述定义, 计算簇的维数要找到一条最长不可约簇的降链。利用 $k[x_1, x_2, \dots, x_n]$ 中的素理想与 A^n 中的不可约簇之间的对应关系, 我们可以将上述问题转化为寻找最长素理想构成的降链的问题。

定义 4: 假设 $V(I)$ 是 $k[x_1, x_2, \dots, x_n]$ 中的一个单项式理想, 那么 $I = \langle f_1, f_2, \dots, f_n \rangle$ 中的每一个生成元 f_i ($i=1, 2, \dots, n$) 都是单项式。

由于计算一般簇的维数是比较困难的, 下面本文考虑一类由单项式理想的定义的簇。

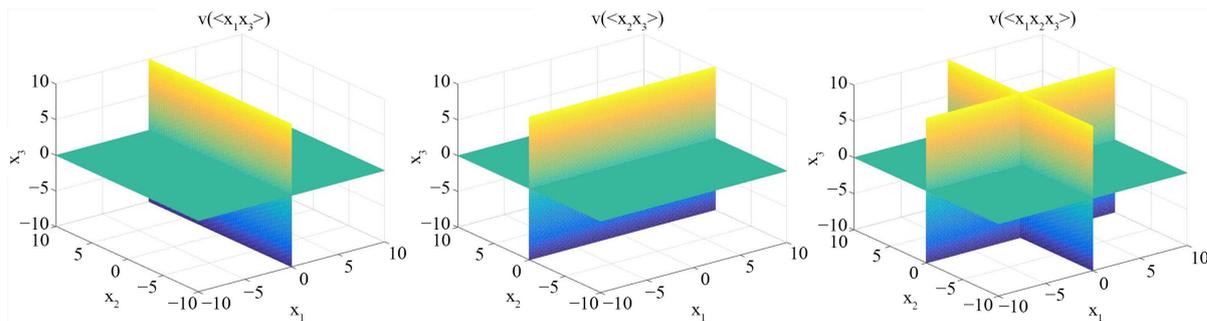
3. 单项式理想所定义的簇维数

由于单项式中的任意非零次数的变量取零时, 单项式就会为零。所以单项式理想的簇我们只需要考虑由类似坐标超平面来构成的。如下例所示:

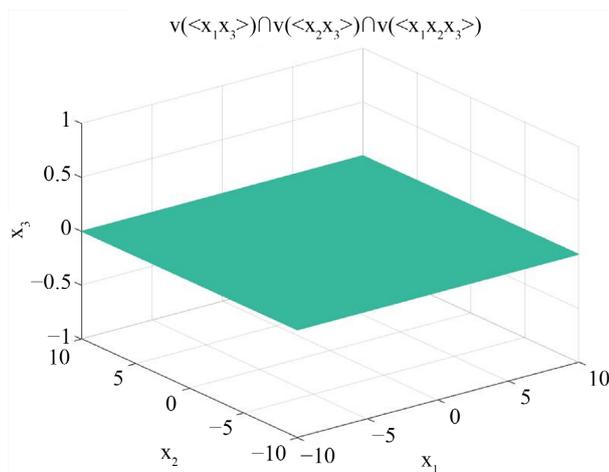
例 1: 假设在 $k[x_1, x_2, \dots, x_n]$ 中有一个单项式理想 $I = \langle x_2^2 x_3^3, x_1^5 x_3^4, x_1^2 x_2 x_3^2 \rangle$, 那么:

$$\begin{aligned} V(I) &= V(\langle x_2^2 x_3^3, x_1^5 x_3^4, x_1^2 x_2 x_3^2 \rangle) \\ &= V(\langle x_2^2 x_3^3 \rangle) \cap V(\langle x_1^5 x_3^4 \rangle) \cap V(\langle x_1^2 x_2 x_3^2 \rangle) \\ &= V(\langle x_2 x_3 \rangle) \cap V(\langle x_1 x_3 \rangle) \cap V(\langle x_1 x_2 x_3 \rangle) \\ &= [V(\langle x_2 \rangle) \cup V(\langle x_3 \rangle)] \cap [V(\langle x_1 \rangle) \cup V(\langle x_2 \rangle)] \\ &\quad \cap [V(\langle x_1 \rangle) \cup V(\langle x_2 \rangle) \cup V(\langle x_3 \rangle)] \end{aligned}$$

簇 $V(\langle x_2 \rangle) \cup V(\langle x_3 \rangle)$, 簇 $V(\langle x_1 \rangle) \cup V(\langle x_2 \rangle)$, 簇 $V(\langle x_1 \rangle) \cup V(\langle x_2 \rangle) \cup V(\langle x_3 \rangle)$ 如下图:



三个簇取交集不难看出是:



上面这个平面计算克鲁尔维数，先找到最长一条由不可约簇构成的链：

$$V_2 \supsetneq V_1 \supsetneq V_0.$$

V_0 表示 $(0,0,0)$ 点， V_1 表示 $(a,0,0)$ 其中 $a \in K$ 。 V_2 表示 $(a,b,0)$ 其中 $a,b \in K$ ，也就是上面的平面。在这个 $V(I)$ 中最长可以找到由三个不可约簇构成的降链。根据克鲁尔维数的定义可知：

$$\text{krdim} \pi(V_2) = 2.$$

与之相对的素理想链：

$$\langle x, y, z \rangle \supsetneq \langle x, y \rangle \supsetneq \langle x \rangle.$$

受上例启发，可以将上述过程简化为计算理想中最小公共变量集的元素个数，一般的簇并不能做到这点。

定理 3： 设单项式理想 $I \in k[x_1, x_2, \dots, x_n]$ ，那么单项式理想的簇 $V(I)$ 的维数和理想 I 公共变量有下述关系：

$$\text{krdim}(V(I)) = \text{krdim}\left(V\left(\langle x_{q_1} \rangle\right) \cup V\left(\langle x_{q_2} \rangle\right) \cup \dots \cup V\left(\langle x_{q_n} \rangle\right)\right),$$

其中 $x_{p_1}, x_{p_2}, \dots, x_{p_q}$ 是在单项式理想 I 中出现的公共变量。

证明： 设 $(m, n \in 0, 1, 2, \dots)$ ，那么

$$\begin{aligned} & V\left(\left\langle x_1^{(1,1)} x_2^{(1,2)} \dots x_n^{(1,n)}, x_1^{(2,1)} x_2^{(2,2)} \dots x_n^{(2,n)}, \dots, x_1^{(m,1)} x_2^{(m,2)} \dots x_n^{(m,n)} \right\rangle\right) \\ &= \bigcap_{i=1}^m V\left(\left\langle x_1^{(i,1)} x_2^{(i,2)} \dots x_n^{(i,n)} \right\rangle\right) \\ &= \bigcap_{i=1}^m \left[\bigcup_{j=1}^n V\left(\langle x_j^{i,j} \rangle\right) \right] \\ &= V\left(\langle x_{p_1}^{k_1} \rangle\right) \cup V\left(\langle x_{p_2}^{k_2} \rangle\right) \cup \dots \cup V\left(\langle x_{p_n}^{k_n} \rangle\right) \\ &= V\left(\langle x_{p_1} \rangle\right) \cup V\left(\langle x_{p_1} \rangle\right) \cup \dots \cup V\left(\langle x_{p_1} \rangle\right), \end{aligned}$$

其中 $x_{p_1}, x_{p_2}, \dots, x_{p_q}$ 是在单项式理想 I 中出现的公共变量。

根据定理 6，把例 5 改成寻找公共变量集，

$$I = \langle x_2^2 x_3^3, x_1^5 x_3^4, x_1^2 x_2 x_3^2 \rangle.$$

公共变量集合为：

$$\{\{x_1, x_2, x_3\}, \{x_1, x_2\}, \{x_1, x_3\}, \{x_2, x_3\}, \{x_3\}\},$$

其中最少的是 $\{x_3\}$ ，个数为 1。那么该簇的维数就等于：

$$\text{krdim}(V(I)) = 3 - 1 = 2.$$

4. 单项式理想簇的维数程序实现

根据上面的寻找最小公共变量集的做法，这里设计一种 maple 程序。先将全部的变元取出来，根据 choose 算法，做成公共变量集合，最后找到其中变元最少的。

Algorithm 1: Calculating the dimension of the varieties of a monomial ideal by this program

Input: A Variety of a Monomial Ideal

Output: The dimension of this the varieties of a monomial ideal

```

1: Str1 := input;
2: while i ≠ length of this variety do
3:   while j ≠ length of Str1 [ i ] / lcoeff Str1 [ i ] do
4:     noset [ i ] := [ noset [ i ], open( [j, 1], Str1 [ I ] / lcoeff( Str1 [ i ] ) ) ];
5:   end while;
6: end while;
7: for i to length of noset do
8:   no1 := set( open( noset [ i ] ), open ( no1 ) );
9: end for;
10: no1 := list( open(no1) );
11: var-choose := choose( no1 );
12: while j to length( noset ) do
13:   while i to length( var-choose ) do
14:     if open( var-choose[ i ] ) intersect open( noset[ j ] ) = [ ] do
15:       var-choose[ i ] := [ ];
16:     end if;
17:   end while;
18: end while;
19: k1 := list ( from 2 to open( var-choose ) ) intersect list( open( var-choose ) );
20: return: length( k1[ length( k1 ) ] ) - length( k1[ nops(1) ] );

```

用变量做成字符串，求全部子集。然后按照子集中元素的个数排列，将元素个数少的子集排在前面。choose 算法如下：

Algorithm 2: choose function

Input: A set of variates with of this the variety of a monomial ideal

Output: All the subsets arranged by the number of elements

```

1: Str2 := input;
2: for i from 1 to length of str2 by 1 do
3:   for j from 1 to length of arr by 1 do
4:     arr := list ( open(arr), list( open( arr [j] ), str2 [i] ) );
5:   end for;
6:   arr := list ( open( arr ), str2[i] );
7: end for;
8: arr [1] := [ arr [1] ];
9: while i ≠ length ( arr ) do
10:   while j ≠ length ( arr ) - 1 do
11:     If length ( arr [ j ] ) > length ( arr [ j+1 ] ) do
12:       t := arr [ j ];
13:       arr [ j ] := arr [ j+1 ];
14:       arr [ j+1 ] := t;
15:     end if;
16:     j := j+1;
17:   end while;
18:   i := i+1;
19: end while;
20: return:arr;

```

下面是利用 maple 执行上述算法得出的结果。

```

> AutoCalc_MonDim([x22·x33, x15·x34, x12·x2·x32]);#本文中例1
2
> AutoCalc_MonDim([x12·x23, x14·x32, x1·x33·x5, x22·x5, x1·x22·x6]);
3
> AutoCalc_MonDim([x1·x32·x73, x34·x62, x2·x33·x5, x22·x5, x1·x22·x6]);
4
> AutoCalc_MonDim([x1·x4, x1·x2, x1·x2·x3·x4, x1·x2·x4, x1·x2·x6, x1·x2·x5]);
5

```

后面两个例子直接计算的话是比较复杂的。上面的几个例子可以看出到，利用 maple 执行上述算法的程序，可以快捷有效的计算出一个由单项式定义的簇的维数。这里 choose 算法中对数据的处理会按照 2^n 增加，本质上求理想中最少公共变量集的元素个数这一做法是不需要 choose 算法中全部的结果。如果可以精准的找到最少公共变量集，是可以对效率有很大的提升。

参考文献

- [1] Cox, D., Little, J. and O'Shea, D. (1997) Ideals, Varieties, and Algorithms. Springer, New York.
<https://doi.org/10.1007/978-1-4757-2693-0>
- [2] Herzog, J. and Hibi, T. (2011) Monomial Ideals. Springer London Ltd., London.
<https://doi.org/10.1007/978-0-85729-106-6>
- [3] Hulek, K. and Helena, V. (2003) Elementary Algebraic Geometry. American Mathematical Society, Providence.
<https://doi.org/10.1090/stml/020>
- [4] Gregor, K. (2011) A Course in Commutative Algebra. Springer, New York.