

一个新的计算平面有理曲线 μ 基的算法

韩伟珍, 孙维昆*

天津职业技术师范大学理学院, 天津

收稿日期: 2026年7月15日; 录用日期: 2026年8月8日; 发布日期: 2026年8月21日

摘要

μ 基作为研究平面有理曲线的重要工具, 不仅能够刻画参数曲线的动直线结构, 还可用于曲线隐式化、奇异点分析以及参数形式与隐式形式之间的相互转化。现有 μ 基的计算方法大多以多项式向量的合冲模为基本对象, 并通过高斯消元法、矩阵行化简或多项式矩阵分解等方法实现计算, 具有较强的一般性和较好的算法效率。本文从另一角度出发, 利用重新定义的向量多项式, 提出一种构造 μ 基的算法, 并辅以实例说明。

关键词

μ 基算法, 隐式化, 平面有理曲线, 动直线

A New Algorithm for Computing μ -Bases of Planar Rational Curves

Weizhen Han, Weikun Sun*

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

Received: July 15, 2026; accepted: August 8, 2026; published: August 21, 2026

Abstract

As an important algebraic tool for studying planar rational curves, a μ -basis not only captures the moving-line structure of a parametric curve, but also plays an essential role in deriving implicit equations, analyzing singularities, and converting between parametric and implicit forms. Existing methods for computing μ -bases usually take the syzygy module of a polynomial vector as the underlying object, and compute them using Gaussian elimination, matrix row reduction, or polynomial matrix factorization. These methods have broad applicability and high computational efficiency. In this paper, we approach the problem from a different perspective. By introducing a new class of vector polynomials,

*通讯作者。

文章引用: 韩伟珍, 孙维昆. 一个新的计算平面有理曲线 μ 基的算法[J]. 应用数学进展, 2026, 15(8): 206-216.
DOI: 10.12677/aam.2026.158346

we propose an algorithm for constructing μ -bases and illustrate the method with examples.

Keywords

μ -Bases Algorithm, Implicitization, Planar Rational Curve, Moving Line

Copyright © 2026 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

平面有理曲线既可以由参数方程表示,也可以由隐式方程表示.参数形式便于曲线造型和几何设计,而隐式形式则更适合处理点在曲线上的判定、曲线求交以及奇异点分析等问题.因此,如何从平面有理曲线的参数表示有效地求出其隐式表示,是计算机辅助几何设计和符号计算中的一个基本问题.

传统的隐式化方法通常依赖结式、Gröbner 基或消元理论.虽然这些方法具有一般性,但在实际计算中常常会产生较大的行列式或较复杂的中间表达式.为了提高隐式化效率, Sederberg 和 Chen 提出了动曲线与动曲面方法,将隐式化问题转化为构造伴随参数对象动曲线或动曲面的问题[1].在此基础上, Cox 等人系统研究了平面有理曲线的动直线理想,并引入了 μ 基理论,从而使 μ 基成为连接参数表示、动直线表示与隐式表示的重要工具[2].围绕 μ 基的计算,已有许多算法被提出. Zheng 和 Sederberg 给出了一种基于 Gröbner 基思想的直接算法,用于计算平面有理曲线的 μ 基[3]. Chen 和 Wang 进一步研究了平面有理曲线 μ 基的性质与计算,并给出了基于向量消元和高斯消元的计算方法[4]. Song 和 Goldman 将相关方法推广到任意维有理曲线情形,提出了适用于一般多项式向量的算法(本文称其为 SG 算法)[5]. Hong 等人提出的算法则将多个一元多项式 syzygy 模的 μ 基计算转化为一个 Sylvester 型矩阵的部分行化简问题(本文称其为 HHK 算法)[6].随后, jia 等对 μ 基理论及其在平面有理曲线和曲面中的应用进行了系统总结,为相关研究提供了全面的理论框架[7].近年来,随着多项式矩阵理论的发展, μ 基的研究逐渐与多项式矩阵最小基问题相结合. Huang 和 Chen 从多项式矩阵分解的角度出发,将 μ 基推广到一元多项式矩阵情形,并提出了适用于多项式向量和多项式矩阵的高效算法[8].进一步地, Chen 和 Yao 针对一元矩阵最小基计算问题提出了两种改进算法,提高了最小基计算效率,为 μ 基及相关矩阵基计算问题提供了新的研究思路[9].

上述算法具有较强的一般性,能够处理不同维数、不同次数的多项式向量.然而,对于某些特殊类型的平面有理曲线,尤其是低次数曲线,如果仅仅使用一般的线性代数或矩阵分解方法,往往难以直接看出 μ 基与曲线系数向量之间的内在联系.如何直接利用这些系数向量构造伴随曲线的动直线,是本文研究的出发点.本文提出的 QF 算法,通过利用重新定义的向量多项式 Q_i [10],构造出一组次数不超过 $n-1$ 且伴随 $F(t)$ 的动直线 $u_i(t)$,并通过首向量消元进行降次化简后得到曲线的一组 μ 基.最后进一步证明该算法的正确性和终止性,并通过具体实例说明算法的有效性及其在隐式化计算中的应用.

2. 基础知识

2.1. μ 基

设 $R[t]$ 为实系数一元多项式环,一条平面有理参数曲线通常可以表示为

$$(x, y) = \left(\frac{f_1(t)}{f_0(t)}, \frac{f_2(t)}{f_0(t)} \right),$$

其中 $f_0(t), f_1(t), f_2(t) \in \mathbb{R}[t]$, 且 $\gcd(f_0, f_1, f_2) = 1$ 。为了便于讨论, 通常在实射影平面 $\mathbb{P}^2(\mathbb{R})$ 中考虑其齐次形式:

$$F(t) = (f_0(t), f_1(t), f_2(t)), \quad (2.1)$$

记 $\deg(F(t)) \triangleq \max(\deg(f_0), \deg(f_1), \deg(f_2))$, 并将(2.1)写成向量多项式形式

$$F(t) = \sum_0^n p_i t^i, p_i \in \mathbb{R}^3, p_0, p_n \neq 0, n \geq 2,$$

其中系数矩阵为 $P = [p_0, p_1, \dots, p_n] \in \mathbb{R}^{3 \times (n+1)}$ 。若(2.1)中满足 $\gcd(f_0, f_1, f_2) = 1$, 则称 $F(t)$ 为本原参数向量; 若 $\text{rank}(P) = 3$, 则称参数曲线为非退化曲线。

在 $\mathbb{P}^2(\mathbb{R})$ 中, 设 $X = (x, y, w)$ 为齐次坐标。一条关于参数 t 的动直线是指系数随 t 变化的一族直线, 其方程可写为

$$L(t, X) = A(t)x + B(t)y + C(t)w = 0, \quad (2.2)$$

其中 $A(t), B(t), C(t) \in \mathbb{R}[t]$, 且 $\deg(L) = \max(\deg(A(t)), \deg(B(t)), \deg(C(t)))$ 为该动直线的次数, 为便于书写, 将式(2.2)的系数向量记为 $L(t) = (A(t), B(t), C(t))$ 。若满足

$$L(t) \cdot F(t) = A(t)f_0(t) + B(t)f_1(t) + C(t)f_2(t) \equiv 0, \quad (2.3)$$

则称 $L(t, X)$ 是伴随 $F(t)$ 的动直线, 或称 $L(t)$ 是 $F(t)$ 的一个合冲。所有满足式(2.3)的动直线系数向量组成 $F(t)$ 的合冲模, 记为

$$\text{Syz}(F) = \{L(t) \in \mathbb{R}[t]^3 \mid L(t) \cdot F(t) = 0\}.$$

平面有理曲线的 μ 基即为该合冲模的一组特殊基, 其定义如下[11]。

定义 2.1 设 $u_1(t), u_2(t) \in \text{Syz}(F)$, 若 $u_1(t)$ 和 $u_2(t)$ 构成 $\text{Syz}(F)$ 的一组自由基, 并且其首向量 $LV(u_1)$ 和 $LV(u_2)$ 线性无关, 则称 $u_1(t)$ 和 $u_2(t)$ 为平面有理曲线 $F(t)$ 的一组 μ 基。

引理 2.2 设 $u_1(t), u_2(t) \in \text{Syz}(F)$, 若存在非零常数 $\lambda \in \mathbb{R}^*$, 使得 $u_1(t) \times u_2(t) = \lambda F(t)$, 则 $u_1(t)$ 和 $u_2(t)$ 构成曲线 $F(t)$ 的一组 μ 基, 特别地, 此时 $\deg(u_1) + \deg(u_2) = \deg(F)$ 。

定理 2.3 设 $u_1(t), u_2(t)$ 为 $F(t)$ 的一组 μ 基, 令 $L_i(t, X) = u_i \cdot X, i = 1, 2$, 则曲线的隐式方程为

$$F(X) = \text{Res}_t(L_1(t, X), L_2(t, X)) = 0. \quad (2.4)$$

定理 2.4 给定非退化曲线 $F(t) = (f_0, f_1, f_2)$, 则 $\text{Syz}(F)$ 是 $\mathbb{R}[t]$ 上秩为 2 的自由模。

证明 考虑 $\mathbb{R}[t]$ -模同态

$$\begin{aligned} \varphi: \mathbb{R}[t]^3 &\rightarrow \mathbb{R}[t] \\ (h_0, h_1, h_2) &\mapsto \varphi((h_0, h_1, h_2)) = h_0 f_0 + h_1 f_1 + h_2 f_2 \end{aligned},$$

则有 $\ker(\varphi) = \text{Syz}(F) = \{(h_0, h_1, h_2) \mid h_0 f_0 + h_1 f_1 + h_2 f_2 = 0\}$ 。由于 $\gcd(f_0, f_1, f_2) = 1$, 即 $1 \in \text{Im}(\varphi)$, 所以 φ 为满射, 并有如下短正合序列:

$$0 \rightarrow \text{Syz}(F(t)) \rightarrow \mathbb{R}[t]^3 \rightarrow \mathbb{R}[t] \rightarrow 0$$

则 $\text{rank}(\mathbb{R}[t]^3) = \text{rank}(\text{Syz}(F)) + \text{rank}(\mathbb{R}[t])$, 所以 $\text{rank}(\text{Syz}(F)) = 3 - 1 = 2$ 。

因为 $\mathbb{R}[t]$ 是主理想整环, $\mathbb{R}[t]^3$ 的任意子模仍为自由模, 因此 $\text{Syz}(F)$ 是 $\mathbb{R}[t]$ 上秩为 2 的自由模。

引理 2.5 设 $G(t) = d(t)F(t)$, $0 \neq d(t) \in \mathbb{R}[t]$, 则 $\text{Syz}(G) = \text{Syz}(F)$ 。

证明 对任意 $h(t) \in \mathbb{R}[t]^3$, 由于 $\mathbb{R}[t]$ 是整环且 $d(t) \neq 0$, 所以

$$h(t) \cdot G(t) = 0 \Leftrightarrow d(t)(h(t) \cdot F(t)) = 0 \Leftrightarrow h(t) \cdot F(t) = 0,$$

从而 $\text{Syz}(G) = \text{Syz}(F)$ 。

2.2. 向量多项式 $Q_k(t)$

对于 $k = 0, 1, \dots, n-1$, 定义向量多项式

$$Q_k(t) = p_{n-k} + p_{n-k+1}t + \dots + p_nt^k, \quad (2.5)$$

于是 $F(t)$ 可以分解为 $F(t) = t^{n-k}Q_k(t) + (p_{n-k-1}t^{n-k-1} + \dots + p_1t + p_0)$ 。根据外积性质有:

$$u_k(t) = Q_k(t) \times F(t) = (p_n \times p_{n-k-1})t^{n-1} + \text{低次项}, k = 0, \dots, n-1,$$

且 $u_k(t) \cdot F(t) = 0$, $\deg(u_k(t)) \leq n-1$, 说明 $u_k(t)$ 是 n 个伴随 $F(t)$ 的动直线。

引理 2.6 设 $Q_k(t)$ 由式(2.5)定义, 则有

$$\langle Q_0(t), Q_1(t), \dots, Q_{n-1}(t), F(t) \rangle_{\mathbb{R}[t]} = \langle p_0, p_1, \dots, p_n \rangle_{\mathbb{R}[t]}.$$

特别地, 若 $F(t)$ 非退化, 则 $\langle Q_0(t), Q_1(t), \dots, Q_{n-1}(t), F(t) \rangle_{\mathbb{R}[t]} = \mathbb{R}[t]^3$ 。

证明 因为

$$\begin{cases} Q_i = p_{n-i} + p_{n-i+1}t + \dots + p_nt^i \\ F = p_0 + tQ_{n-1}, i = 0, \dots, n-1 \end{cases} \Leftrightarrow \begin{cases} p_{n-i} = Q_i - tQ_{i-1} \\ p_0 = F - tQ_{n-1} \end{cases},$$

所以 $\langle Q_0(t), Q_1(t), \dots, Q_{n-1}(t), F(t) \rangle_{\mathbb{R}[t]} = \langle p_0, p_1, \dots, p_n \rangle_{\mathbb{R}[t]}$ 。当 $F(t)$ 非退化时, 即 $p_0, p_1, \dots, p_n$ 张成整个 $\mathbb{R}[t]^3$, 所以 $\langle Q_0(t), Q_1(t), \dots, Q_{n-1}(t), F(t) \rangle_{\mathbb{R}[t]} = \mathbb{R}[t]^3$ 。

定理 2.7 $\text{Syz}(F) = \langle u_0(t), u_1(t), \dots, u_{n-1}(t) \rangle_{\mathbb{R}[t]}$ 。

证明 由 $u_k(t)$ 的定义可知 $\langle u_0(t), u_1(t), \dots, u_{n-1}(t) \rangle_{\mathbb{R}[t]} \subseteq \text{Syz}(F)$ 。下面证明反向包含关系。

任取 $h(t) \in \text{Syz}(F)$, 则 $h(t) \cdot F(t) = 0$ 。因为 $\gcd(f_0, f_1, f_2) = 1$, 所以存在

$$a(t) = (a_0(t), a_1(t), a_2(t)) \in \mathbb{R}[t]^3,$$

使得 $a \cdot F = a_0f_0 + a_1f_1 + a_2f_2 = 1$ 。由向量三重外积公式可得

$$(a(t) \times h(t)) \times F(t) = h(t)(a(t) \cdot F(t)) - a(t)(h(t) \cdot F(t)) = h(t)。$$

令 $v(t) = a(t) \times h(t)$, 则 $h(t) = v(t) \times F(t)$, 说明 $\text{Syz}(F) = \{v(t) \times F(t) \mid v \in \mathbb{R}[t]^3\}$ 。

由引理 2.6 可知, 对任意 $v \in \mathbb{R}[t]^3$, 存在 $b_0, \dots, b_{n-1}, b_n$, 使得 $v = b_0Q_0 + \dots + b_{n-1}Q_{n-1} + b_nF$, 并且 $h(t) = b_0(Q_0 \times F) + \dots + b_{n-1}(Q_{n-1} \times F) = b_0u_0 + \dots + b_{n-1}u_{n-1}$, 故 $\text{Syz}(F) \subseteq \langle u_0(t), \dots, u_{n-1}(t) \rangle_{\mathbb{R}[t]}$, 从而 $\text{Syz}(F) = \langle u_0(t), u_1(t), \dots, u_{n-1}(t) \rangle_{\mathbb{R}[t]}$ 。

注: 该定理说明, 非零的 $u_k(t)$ 构成 $\text{Syz}(F)$ 的一组生成元。然而这组生成元一般不是 μ 基, 因为它们的数量可能多于两个, 并且次数分布也不一定最小。后续算法需要通过本原化、排序以及首向量消元, 从这些 $u_k(t)$ 中提取一组 μ 基。

2.3. 示例

为直观说明操作过程, 现给出下面的一个例子。

例 考虑平面有理曲线 $G(t) = (1+t, t+t^2+t^3+t^4, t^2+2t^3+t^4)$, 计算其一组 μ 基。

解 计算三个分量的首一最大公因式为 $d(t) = t + 1$, 并记约去公因式后的本原参数向量为 $F(t) = (1, t + t^3, t^2 + t^3)$, 将其写为 $F(t) = p_0 + p_1 t + p_2 t^2 + p_3 t^3$, 其中 $p_0 = (1, 0, 0)$, $p_1 = (0, 1, 0)$, $p_2 = (0, 0, 1)$, $p_3 = (0, 1, 1)$, 系数矩阵为 $P = [p_0, p_1, p_2, p_3]$, $\text{rank}(P) = 3$, 故 $F(t)$ 本原且非退化。

根据向量多项式 $Q_k(t)$ 的定义, 得 $Q_0(t) = p_3 = (0, 1, 1)$, $Q_1(t) = p_2 + p_3 t = (0, t, 1 + t)$, 并生成动直线 $u_0(t) = Q_0(t) \times F(t) = (-t + t^2, 1, -1)$, $u_1(t) = Q_1(t) \times F(t) = (-t - t^2, 1 + t, -t)$ 。因为首向量满足 $LV(u_0) = -LV(u_1) = (1, 0, 0)$, 所以 $LV(u_0)$ 与 $LV(u_1)$ 线性相关, 需进行首向量消元。计算

$$\tilde{u}_0(t) = u_0(t) - (-1)t^{2-2}u_1(t) = (-2t, 2 + t, -1 - t),$$

此时 $LV(\tilde{u}_0) = (-2, 1, -1)$ 与 $LV(u_1)$ 线性无关, 且 $\deg(\tilde{u}_0) + \deg(u_1) = \deg(F)$, $\tilde{u}_0(t) \times u_1(t) = F(t)$, 所以 $\tilde{u}_0(t)$ 和 $u_1(t)$ 是曲线 $F(t)$ 的一组 μ 基。根据引理 2.5 可知 $\tilde{u}_0(t)$ 和 $u_1(t)$ 也是曲线 $G(t)$ 的一组 μ 基。

本例首先通过预处理约去公因式后, 得到本原且系数矩阵秩为 3 的参数向量 $F(t)$ 。随后依次构造 $Q_0(t)$ 、 $Q_1(t)$ 及动直线 $u_0(t)$ 、 $u_1(t)$ 。由于 $u_0(t)$ 、 $u_1(t)$ 的首向量线性相关, 对 $u_0(t)$ 进行首向量消元后得到 $\tilde{u}_0(t)$, 从而得到一组次数分别为 1 和 2 的 μ 基。所以下面将上述操作过程以算法形式呈现。

3. QF 算法

在 2.2 节中, 通过 $Q_k(t)$ 构造出一组动直线 $u_k(t)$, 其中 $k = 0, 1, \dots, n-1$, 并且在曲线非退化条件下, 这些 $u_k(t)$ 能够生成整个合冲模 $\text{Syz}(F)$ 。因此计算 $F(t)$ 的一组 μ 基问题, 可以转化为从有限个 $u_k(t)$ 中提取出一组首向量线性无关且次数之和等于 $F(t)$ 次数的基。

基于这一思想, 本文提出 QF 算法。该算法首先利用 $Q_k(t) \times F(t)$ 构造候选动直线 $u_k(t)$, 并将非零向量本原化。随后按照次数对候选向量重新排序, 并通过首向量消元不断降低候选向量的次数。当候选动直线集合 U 存在两个向量 $u_a(t), u_b(t)$ 满足引理 2.2, 则 $u_a(t), u_b(t)$ 即为 $F(t)$ 的一组 μ 基。

但需要指出的是, QF 算法的主要结论建立在参数向量本原且系数矩阵秩为 3 的条件下。具体地, 设原始输入参数向量为 $G(t) = (g_0(t), g_1(t), g_2(t)) \in \mathbb{R}[t]^3$, 对其首先执行表 1 中的预处理, 对于通过预处理的参数向量, 再执行表 2 所示的 QF 算法。

Table 1. Parameter vector preprocessing algorithm

表 1. 参数向量预处理算法

Algorithm: Preprocessing Algorithm

Input: $G(t) = (g_0(t), g_1(t), g_2(t))$, $G(t) \neq 0$ 。

Output: 预处理后的本原参数向量 $F(t)$ 以及输入状态。

Step 0. 计算 $d(t) = \gcd(g_0(t), g_1(t), g_2(t))$, 并令 $F(t) = G(t)/d(t) = (f_0(t), f_1(t), f_2(t))$ 。

Step 1. 令 $n = \max\{\deg(f_0), \deg(f_1), \deg(f_2)\}$, 并将 $F(t)$ 写成

$$F(t) = p_0 + p_1 t + \dots + p_n t^n, \quad p_i \in \mathbb{R}^3, \quad p_n \neq 0,$$

构造系数矩阵 $P = [p_0, p_1, \dots, p_n]$, 并计算 $r = \text{rank}(P)$ 。

Step 2. 若 $r = 3$, 则输出 $F(t)$, 并转入表 2;

若 $r = 2$, 则输出“退化输入: 曲线位于一条固定直线上”, 并停止;

若 $r \leq 1$, 则输出“退化输入: 参数像为一个固定点或输入无效”, 并停止。

Table 2. QF algorithm**表 2.** QF 算法**Algorithm:** QF Algorithm

Input: $F(t) = (f_0(t), f_1(t), f_2(t))$, 其中 $\gcd(f_0, f_1, f_2) = 1$, $\text{rank}([p_0, p_1, \dots, p_n]) = 3$ 且

$F(t) = p_0 + p_1 t + \dots + p_n t^n$, $p_i \in \mathbb{R}^3$, $p_n \neq 0$, $n \geq 2$ 。

Output: 两个非零向量多项式, 构成 $\text{Syz}(F)$ 的一组 μ 基。

Step 0. 初始化 $U = \emptyset$, $i = 0$ 。

Step 1. 构造 $Q_i(t) = p_{n-i} + p_{n-i} t + \dots + p_{n-i} t^i$, $0 \leq i \leq n-1$, 并令

$u_i(t) = Q_i(t) \times F(t) = (u_{i,1}(t), u_{i,2}(t), u_{i,3}(t))$,

若 $u_i(t) = 0$, 转到 **Step 7**。

若 $u_i(t) \neq 0$, 令 $g(t) = \gcd(u_{i,1}(t), u_{i,2}(t), u_{i,3}(t))$, $u_i(t) = u_i(t)/g(t)$, 将 $u_i(t)$ 加入 U 。

Step 2. 将 U 中元素重新编号为 $U = \{u_1, \dots, u_s\}$, 使得 $d_1 \geq d_2 \geq \dots \geq d_s$, $d_i = \deg(u_i)$ 。

Step 3. 若当前首向量集合 $\{LV(u_1), \dots, LV(u_s)\}$ 线性相关, 转到 **Step 4**, 否则转到 **Step 6**。

Step 4. 取非零向量 $\alpha = (\alpha_1, \dots, \alpha_s) \in \mathbb{R}^s$, 使得 $\alpha_1 LV(u_1) + \dots + \alpha_s LV(u_s) = 0$ 。

Step 5. 设 j 是满足 $\alpha_i \neq 0$ 的最小指标, 即 $\alpha_1 = \dots = \alpha_{j-1} = 0$, $\alpha_j \neq 0$ 。定义

$$\tilde{u}_j = \sum_{k=j}^s \alpha_k t^{d_j - d_k} u_k = (\tilde{u}_{j,1}(t), \tilde{u}_{j,2}(t), \tilde{u}_{j,3}(t)),$$

若 $\tilde{u}_j(t) = 0$, 从 U 中删除 u_j , 并令 $s = s - 1$ 。

若 $\tilde{u}_j(t) \neq 0$, 则 $\deg(\tilde{u}_j) < \deg(u_j)$, 令

$$\tilde{g}(t) = \gcd(\tilde{u}_{j,1}(t), \tilde{u}_{j,2}(t), \tilde{u}_{j,3}(t)), \quad u_j(t) = \tilde{u}_j(t)/\tilde{g}(t),$$

并回到 **Step 2**。

Step 6. 若 U 中存在两个向量 $u_a(t)$ 、 $u_b(t)$, 满足

$$\text{rank}(LV(u_a), LV(u_b)) = 2, \quad u_a(t) \times u_b(t) = \lambda F(t), \quad \lambda \in \mathbb{R}^*.$$

则输出 $u_a(t)$ 、 $u_b(t)$ 并停止, 否则转到 **Step 7**。

Step 7. 若 $i < n-1$, 则令 $i = i + 1$, 并回到 **Step 1**; 若 $i = n-1$, 则算法停止。

与已有 μ 基计算方法相比, QF 算法的出发点有所不同。传统方法通常以多项式向量的合冲模为基本对象, 进而利用 Gröbner 基、高斯消元、矩阵行化简或多项式矩阵分解等工具进行计算。这些方法具有较强的一般性, 能够适用于更广泛的多项式向量情形。相比之下, QF 算法直接利用平面有理曲线参数多项式的系数向量构造候选动直线, 不需要一开始设定一般形式的高次动直线, 也不用在初始阶段构造类似 $(2d+1) \times 3(d+1)$ 的稀疏矩阵, 从而减少计算过程中的系数膨胀问题。该算法先得到次数不超过 $n-1$ 的候选动直线, 再在有限个三维向量多项式上进行本原化、次数排序、首向量消元以及 μ 基验证。因此, QF 算法的优势主要体现在构造过程较为直接、候选动直线来源明确, 并且能够较好地反映 μ 基与曲线系数向量之间的联系。特别是对于低次数平面有理曲线, 该方法不仅可以用于计算 μ 基, 也有助于进一步推导 μ 基的显式表达式以及曲线的隐式方程。

4. 算法的正确性与终止性

引理 4.1 算法运行过程中集合 U 中的每个非零向量均属于 $\text{Syz}(F)$ 。

证明 首先 Step 1 中构造的 $u_i(t) = Q_k(t) \times F(t)$ 满足 $u_i(t) \cdot F(t) = 0$, 因此若 $u_i(t) \neq 0$, 则有 $u_i(t) \in \text{Syz}(F)$ 。

若在 Step 1 中对 $u_i(t)$ 进行本原化, 设 $g_i(t) = \gcd(u_{i,1}(t), u_{i,2}(t), u_{i,3}(t))$ 且 $\bar{u}_i(t) = \frac{u_i(t)}{g_i(t)}$, 由于 $u_i(t) \cdot F(t) = 0$, 于是 $g_i(t) \bar{u}_i(t) \cdot F(t) = 0$, 又因为 $\mathbb{R}[t]$ 是整环且 $g_i(t) \neq 0$, 所以 $\bar{u}_i(t) \cdot F(t) = 0$ 。因此, Step 1 中加入集合 U 的非零向量均属于 $\text{Syz}(F)$ 。

下面考虑 Step 5, 假设当前集合 U 中的非零向量都属于 $\text{Syz}(F)$, Step 5 构造 $\tilde{u}_j = \sum_{k=j}^s \alpha_k t^{d_j-d_k} u_k$, 由于 $\text{Syz}(F)$ 是 $\mathbb{R}[t]$ -模, 而 $\tilde{u}_j(t)$ 是若干个合冲的 $\mathbb{R}[t]$ -线性组合, 所以 $\tilde{u}_j(t) \in \text{Syz}(F)$ 。若 $\tilde{u}_j(t) \neq 0$, 设 $\tilde{g}_j(t) = \gcd(\tilde{u}_{j,1}(t), \tilde{u}_{j,2}(t), \tilde{u}_{j,3}(t))$ 且 $\tilde{u}$ 。同理由 $\tilde{u}_j(t) \cdot F(t) = 0$ 可得 $\tilde{u}$ 。因此, 本原化后的 $\tilde{u}$ 仍属于 $\text{Syz}(F)$ 。若 $\tilde{u}_j(t) = 0$, 算法只删除一个向量, 不会产生新的非零向量。

综上, Step 1 和 Step 5 都不会使集合 U 中的非零向量离开 $\text{Syz}(F)$, 因此算法运行过程中集合 U 中的非零向量均属于 $\text{Syz}(F)$ 。

定理 4.2 (正确性) 输入非退化曲线 $F(t) = (f_0(t), f_1(t), f_2(t)) \in \mathbb{R}[t]^3$ 且 $\gcd(f_0, f_1, f_2) = 1$, 则 QF 算法在 Step 6 输出两个向量 $u_a(t), u_b(t)$, 则 $u_a(t), u_b(t)$ 构成 $F(t)$ 的一组 μ 基。

证明 由引理 4.1 可知, 算法运行过程中集合 U 中的非零向量均属于 $\text{Syz}(F)$ 。因此, 若 QF 算法在 Step 6 输出两个向量 $u_a(t), u_b(t)$, 则 $u_a(t) \in \text{Syz}(F)$, $u_b(t) \in \text{Syz}(F)$ 。

另一方面, Step 6 的检验条件给出 $\text{rank}(LV(u_a), LV(u_b)) = 2$, 并且存在非零常数 $\lambda \in \mathbb{R}^*$, 使得 $u_a(t) \times u_b(t) = \lambda F(t)$ 。因此 $u_a(t), u_b(t)$ 构成 $F(t)$ 的一组 μ 基。

引理 4.3 Step 5 中的消元过程会严格降低某个向量 $u_j(t)$ 的次数。

证明 经过 Step 2 重新编号后, 设 $U = \{u_1, u_2, \dots, u_s\}$, 且 $d_1 \geq d_2 \geq \dots \geq d_s$, $d_j = \deg(u_j)$, 若首向量集合 $\{LV(u_1), LV(u_2), \dots, LV(u_s)\}$ 线性相关, 则存在非零向量 $\alpha = (\alpha_1, \dots, \alpha_s) \in \mathbb{R}^s$, 使得 $\alpha_1 LV(u_1) + \dots + \alpha_s LV(u_s) = 0$ 。设 j 是满足 $\alpha_j \neq 0$ 的最小指标, 即 $\alpha_1 = \dots = \alpha_{j-1} = 0$, $\alpha_j \neq 0$ 。Step 5 中定义 $\tilde{u}_j = \sum_{k=j}^s \alpha_k t^{d_j-d_k} u_k(t)$, 因为 $d_j \geq d_k$, 所以每一项 $t^{d_j-d_k} u_k(t)$ 的次数都不超过 d_j 。当 $u_k(t)$ 的次数为 d_k 时, $t^{d_j-d_k} u_k(t)$ 的最高次项次数为 d_j , 其最高次项系数为 $LV(u_k)$ 。因此, $\tilde{u}_j(t)$ 中 t^{d_j} 项的系数为

$$\sum_{k=j}^s \alpha_k LV(u_k) = \sum_{k=1}^s \alpha_k LV(u_k) = 0,$$

即 $\tilde{u}_j(t)$ 的 d_j 次项被消去, 从而 $\deg(\tilde{u}_j) < d_j = \deg(u_j)$ 。

若 $\tilde{u}_j(t) \neq 0$, 令 $u_j(t) = \frac{\tilde{u}_j(t)}{\gcd(\tilde{u}_{j,1}(t), \tilde{u}_{j,2}(t), \tilde{u}_{j,3}(t))}$, 即用本原化后的结果替换 $u_j(t)$, 该过程不会增

加向量多项式的次数, 所以替换后 $u_j(t)$ 的次数仍然严格小于替换前的次数。因此, Step 5 中的消元过程会严格降低某个向量 $u_j(t)$ 的次数。

定理 4.4 (终止性) QF 算法经过有限步后终止。

证明 首先算法中由 $Q_k(t) \times F(t)$ 构造动直线的指标 i 只取 $0, 1, \dots, n-1$, 因此动直线的构造过程至多进行 n 次。下面证明首向量消元过程也只能执行有限次。

设当前非零候选集合为 U , 定义非负整数

$$M(U) = \sum_{u \in U} (\deg(u) + 1),$$

若 Step 5 中 $\tilde{u}_j(t) = 0$, 则算法从 U 中删去 $u_j(t)$, $M(U)$ 至少减少 1。若 Step 5 中 $\tilde{u}_j(t) \neq 0$, 则由引理 4.3

可知, $\tilde{u}_j(t)$ 的次数严格小于 $u_j(t)$ 的次数, 并且去除三个分量的公因子后得到的 $\tilde{u}_j(t)$ 替换 $u_j(t)$ 后, $M(U)$ 也严格减少。因此在 **Step 2** 到 **Step 5** 的每一次循环中, 非负整数 $M(U)$ 都严格下降, 但非负整数不可能无限严格下降, 所以降次消元过程必定在有限步内结束。综上, 向量 $u_i(t)$ 的生成过程最多执行 n 次, 而每次的降次消元过程又只能执行有限步, 因此, 整个算法必定在有限步内终止。

5. 算法实验比较

为了进一步说明本文 QF 算法的有效性, 将 QF 算法与三元多项式组情形下的 SG 算法和 HHK 算法进行比较。下面分别选取三次、四次、五次、六次平面有理曲线作为测试例子。为了更加全面地比较三种算法, 本文从以下几个方面进行统计:

- 1) 初始候选向量次数, 该指标按各算法实际执行过程统计。对 QF 算法, 统计其 **Step 1** 中构造 $u_i(t) = Q_k(t) \times F(t)$ 并本原化之后的非零候选向量次数; 对 SG 算法, 统计其 **Step 1** 中构造的三条合冲向量 $(-f_1, f_0, 0)$ 、 $(-f_2, 0, f_0)$ 、 $(0, -f_2, f_1)$ 的次数, SG 算法在此处不约去分量公共因子;
- 2) 降次次数, 该指标反映算法在首向量消去和次数降低过程中的计算量;
- 3) 矩阵规模和行变换次数, 该指标反映的是 HHK 算法在矩阵方法下求解过程中的计算量;
- 4) 表达式规模, 为衡量最终 μ 基表达式的简洁程度, 引入指标:

$$\text{表达式规模} = (N, M),$$

其中 N 表示一组 μ 基向量所有分量中非零单项式的总数, M 表示所有系数绝对值中的最大值。 N 越小, 说明表达式项数越少; M 越小, 说明系数膨胀越弱。

为便于比较, 在表 3 中先给出不同次数曲线下三种算法的整体运行结果, 再在附录中分别列出每个例子中三种算法得到的 μ 基表达式(见表 A1~A4)。

从表 3 可以看到, 在进入降次过程之前, QF 算法得到的候选向量次数整体低于 SG 算法。从降次次数来看, QF 算法在所有例子中均少于 SG 算法, 说明 QF 算法构造出的初始候选向量更接近最终 μ 基, 后续降次过程更短。与 HHK 算法相比, QF 算法不需要构造 HHK 方法中规模为 $(2d+1) \times 3(d+1)$ 的系数矩阵, 从而避免了较大规模的矩阵行化简。从最终表达式规模看, QF 算法在多数例子中得到的 μ 基表达式也更简洁。例如 $F_{4,1}$ 中 QF 算法的表达式规模为 $(15, 70)$, 小于 SG 算法的 $(16, 452)$ 和 HHK 算法的 $(16, 210)$ 。因此, QF 算法的优势主要体现在进入降次前候选向量次数较低、降次次数少、避免大规模矩阵计算, 并且在部分例子中能够得到更加简洁的 μ 基表达式。

Table 3. Comparison of three algorithms

表 3. 三种算法比较

曲线	初始候选向量次数		降次次数		HHK 算法		表达式规模		
	QF	SG	QF	SG	矩阵规模	行变换次数	QF	SG	HHK
$F_{3,1}$	(2, 1, 1)	(3, 3, 3)	0	5	7×12	17	(12, 16)	(12, 16)	(12, 320)
$F_{3,2}$	(2, 2, 2)	(3, 3, 3)	1	5	7×12	19	(12, 6)	(14, 6)	(11, 6)
$F_{4,1}$	(3, 3, 3, 3)	(4, 4, 4)	3	6	9×15	24	(15, 70)	(16, 452)	(16, 210)
$F_{4,2}$	(3, 3, 2, 3)	(4, 4, 4)	2	5	9×15	27	(13, 4)	(14, 9)	(14, 9)
$F_{5,1}$	(4, 2, 2, 2, 4)	(5, 5, 5)	3	7	11×18	31	(19, 128)	(19, 128)	(18, 837)
$F_{5,2}$	(4, 4, 4, 4, 4)	(5, 5, 5)	5	7	11×18	47	(19, 337)	(19, 337)	(17, 337)
$F_{6,1}$	(5, 5, 5, 4, 5, 5)	(6, 6, 6)	6	9	13×21	45	(21, 5)	(22, 28)	(22, 10)
$F_{6,2}$	(5, 5, 4, 4, 5, 5)	(6, 6, 6)	5	9	13×21	63	(23, 42)	(23, 254)	(22, 84)

6. 进一步讨论

本文通过构造向量多项式 $Q_k(t) \times F(t)$, 给出了计算平面有理曲线 μ 基的一种新算法。未来, 对于更高维多项式向量组或者有理曲面情形, 如何重新构造类似的向量多项式, 并保持动直线较好的次数控制和生成性, 仍需要进一步讨论。其次, 对于退化曲线或者特殊系数关系下的情形, 仍需进一步分析算法的适用性。此外, 如何将 QF 算法推广到一元多项式组 μ 基的计算上也是后续需要继续研究的问题。

参考文献

- [1] Sederberg, T.W. and Chen, F. (1995) Implicitization Using Moving Curves and Surfaces. In: Mair, S.G. and Cook, R., Eds., *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, Association for Computing Machinery, 301-308. <https://doi.org/10.1145/218380.218460>
- [2] Cox, D.A., Sederberg, T.W. and Falai Chen, (1998) The Moving Line Ideal Basis of Planar Rational Curves. *Computer Aided Geometric Design*, **15**, 803-827. [https://doi.org/10.1016/s0167-8396\(98\)00014-4](https://doi.org/10.1016/s0167-8396(98)00014-4)
- [3] Zheng, J. and Sederberg, T.W. (2001) A Direct Approach to Computing The μ -Basis of Planar Rational Curves. *Journal of Symbolic Computation*, **31**, 619-629. <https://doi.org/10.1006/jSCO.2001.0437>
- [4] Chen, F. and Wang, W. (2002) The μ -Basis of a Planar Rational Curve—Properties and Computation. *Graphical Models*, **64**, 368-381. [https://doi.org/10.1016/s1077-3169\(02\)00017-5](https://doi.org/10.1016/s1077-3169(02)00017-5)
- [5] Song, N. and Goldman, R. (2009) M-Bases for Polynomial Systems in One Variable. *Computer Aided Geometric Design*, **26**, 217-230. <https://doi.org/10.1016/j.cagd.2008.04.001>
- [6] Hong, H., Hough, Z. and Kogan, I.A. (2017) Algorithm for Computing M-Bases of Univariate Polynomials. *Journal of Symbolic Computation*, **80**, 844-874. <https://doi.org/10.1016/j.jsc.2016.08.013>
- [7] Jia, X., Shi, X. and Chen, F. (2018) Survey on the Theory and Applications of M-Bases for Rational Curves and Surfaces. *Journal of Computational and Applied Mathematics*, **329**, 2-23. <https://doi.org/10.1016/j.cam.2017.07.023>
- [8] Huang, B. and Chen, F. (2021) Computing M-Bases of Univariate Polynomial Matrices Using Polynomial Matrix Factorization. *Journal of Systems Science and Complexity*, **34**, 1189-1206. <https://doi.org/10.1007/s11424-020-9314-6>
- [9] Chen, L. and Yao, S. (2025) Two Improved Algorithms to Compute the Minimal Bases of Univariate Matrices. *Journal of Systems Science and Complexity*, **38**, 1312-1329. <https://doi.org/10.1007/s11424-024-3306-x>
- [10] Goldman, R.N., Sederberg, T.W. and Anderson, D.C. (1984) Vector Elimination: A Technique for the Implicitization, Inversion, and Intersection of Planar Parametric Rational Polynomial Curves. *Computer Aided Geometric Design*, **1**, 327-356. [https://doi.org/10.1016/0167-8396\(84\)90020-7](https://doi.org/10.1016/0167-8396(84)90020-7)
- [11] Wang, X., Goldman, R. and Sederberg, T.W. (2016) Explicit M-Bases for Conic Sections and Planar Rational Cubic Curves. *Computer Aided Geometric Design*, **41**, 62-75. <https://doi.org/10.1016/j.cagd.2015.11.003>

附录

下面分别列出第 5 节中每个例子中三种算法得到的 μ 基表达式。

Table A1. μ -basis expressions for cubic curves

表 A1. 三次曲线 μ 基表达式

算法	$F_{3,1} = (-t^3 + 4t^2, 4t^3 - 4, -3t^3 - 4t^2 - 1)$	$F_{3,2} = (2t^3 + 2t^2 + 2t - 2, t^3 + 2t^2 - t, 2t^3 - t^2 - t + 1)$
QF	$u_a = (16t + 16, t - 4, -4t + 16)$	$u_a = (3t + 2, -6t - 6, 4)$
	$u_b = (16t^2 + 16, 16t^2 + 1, 16t^2 - 4)$	$u_b = (t^2 + 2t - 1, -6t^2, 2t^2 + 4t - 2)$
SG	$u_a = (16t + 16, t - 4, -4t + 16)$	$u_a = (3t + 2, -6t - 6, 4)$
	$u_b = (16t^2 + 16, 16t^2 + 1, 16t^2 - 4)$	$u_b = (5t^2 - t - 1, -6t^2 - 6t + 6, -2t^2 + 4t - 2)$
HHK	$u_a = (16t + 16, t - 4, -4t + 16)$	$u_a = (3t + 2, -6t - 6, 4)$
	$u_b = (80t^2 + 320t + 272, 20t^2 - 63, 252)$	$u_b = (2t^2 - 3t - 1, 6, -2t^2 - 2)$

Table A2. μ -basis expressions for quartic curves

表 A2. 四次曲线 μ 基表达式

算法	$F_{4,1} = (t^4, 3t^4 + 2t - 3, 2t^4 - 2t^3 - 4t^2 + 2t - 2)$	$F_{4,2} = (-2t^4 - t^3 - 2t, -2t^4 - t^3 - 2t^2, -2t - 2, 2t^4 - t^3 - t^2 + 2t)$
QF	$u_a = (26t^2 - 6, -10t^2 + 6t - 4, 2t^2 - 7t + 6)$	$u_a = (t^2 - t + 4, -2t, t^2 - 2t + 2)$
	$u_b = (2t^2 - 32t - 70, 10t^2 + 2t, -16t^2 - 3t)$	$u_b = (4t + 1, -2t^2 - t, -2t^2 + t)$
SG	$u_a = (26t^2 - 6, -10t^2 + 6t - 4, 2t^2 - 7t + 6)$	$u_a = (t^2 - t + 4, -2t, t^2 - 2t + 2)$
	$u_b = (452t^2 + 48t, -190t^2 + 102t - 70, 59t^2 - 118t + 105)$	$u_b = (t^2 - 9t + 2, 4t^2, 5t^2 - 4t + 2)$
HHK	$u_a = (210t^2 - 32t - 118, -70t^2 + 50t - 32, -59t + 48)$	$u_a = (2t^2 + 2t + 9, -2t^2 - 5t, -3t + 4)$
	$u_b = (14t^2 - 16t - 38, 4t - 2, -7t^2 - 5t + 3)$	$u_b = (t^2 - t + 4, -2t, t^2 - 2t + 2)$

Table A3. μ -basis expressions for quintic curves

表 A3. 五次曲线 μ 基表达式

算法	$F_{5,1} = (t^5 - 2, -4t^5 - 4t^4 - t, 3t^5 + 3t^4 - 4)$	$F_{5,2} = (2t^5 + t^4 - 2t^3 - 2t^2 - 2t + 1, t^5 + 2t^4 + t^3 - 2t^2 - t, -2t^5 + t^4 + 2t^3 + 2t^2)$
QF	$u_a = (128t^3 + 128t^2 + 3t + 10, -7t^3 - 48t^2 - 2, -52t^3 - 64t^2 - t - 5)$	$u_a = (9t^3 + 18t^2 + 5t, -4t^3 - 8t^2 - 10t + 5, 7t^3 + 18t^2 + 6t - 4)$
	$u_b = (3t^2 + 19t + 16, -2t - 6, -t^2 - 8t - 8)$	$u_b = (337t^2 + 80t, -42t^2 - 305t + 80, 316t^2 + 212t - 161)$
SG	$u_a = (128t^3 + 128t^2 + 3t + 10, -7t^3 - 48t^2 - 2, -52t^3 - 64t^2 - t - 5)$	$u_a = (11t^3 - 28t^2 - 13t, 12t^3 - 16t^2 + 20t - 13, 17t^3 - 10t^2 - 26t - 2)$

续表

	$u_b = (3t^2 + 19t + 16, -2t - 6, -t^2 - 8t - 8)$	$u_b = (337t^2 + 80t, -42t^2 - 305t + 80, 316t^2 + 212t - 161)$
HHK	$u_a = (4t^3 - 28t^2 - 837t - 806, t^3 - 8t^2 + 56t + 302, 343t + 403)$	$u_a = (79t^3 + 105t^2 + 84t, -158t^3 + 27t^2 + 55t + 84, 33t + 143)$
	$u_b = (3t^2 + 19t + 16, -2t - 6, -t^2 - 8t - 8)$	$u_b = (337t^2 + 80t, -42t^2 - 305t + 80, 316t^2 + 212t - 161)$

Table A4. μ -basis expressions for sextic curves**表 A4.** 六次曲线 μ 基表达式

算法	$F_{6,1} = (-t^6 + t^4 + t^2, t^6 - t^4 + t^3 - 1, 2t^6 + t^5 + t^3 - t^2 - t - 1)$	$F_{6,2} = (-t^6 + t^5 - t^4 + t^3 + t^2 - t - 1, t^6 + t^5 + t^4 - t^2 + t, 3t^6 + t^5 - t^4 - t^3 + t^2 - t + 1)$
QF	$u_a = (2t^3 + t^2 + 4t - 2, 4t^3 + 2t^2 + 4t + 5, -t^3 + t - 5)$	$u_a = (4t^2 + 11t - 18, -12t^3 + 24t^2 + 30t - 16, 4t^3 - 4t^2 - 9t - 18)$
	$u_b = (3t^3 + t^2 - t - 1, t^3 + 2t^2 + 2t, t^3 - t^2 - 2t)$	$u_b = (9t^3 + 7t^2 - 19t - 9, 42t^3 + 42t^2 - 6t - 10, -11t^3 - 25t^2 - 27t - 9)$
SG	$u_a = (t^3 - 5t + 1, -3t^3 - 2t - 5, 2t^3 - t^2 - 3t + 5)$	$u_a = (8t^3 + 4t^2 - 23t + 2, 44t^3 + 24t^2 - 22t, -12t^3 - 20t^2 - 19t + 2)$
	$u_b = (28t^3 + 9t^2 - 14t - 8, 6t^3 + 18t^2 + 16t - 5, 11t^3 - 10t^2 - 21t + 5)$	$u_b = (53t^3 + 39t^2 - 118t - 43, 254t^3 + 234t^2 - 52t - 50, -67t^3 - 145t^2 - 154t - 43)$
HHK	$u_a = (5t^3 + 2t^2 + 3t - 3, 5t^3 + 4t^2 + 6t + 5, -t^2 - t - 5)$	$u_a = (4t^3 + 8t^2 + 5t - 26, 4t^3 + 48t^2 + 34t - 24, -16t^2 - 23t - 26)$
	$u_b = (10t^3 + 3t^2 - 8t - 2, 6t^2 + 4t - 5, 5t^3 - 4t^2 - 9t + 5)$	$u_b = (6t^3 + 14t^2 + 13t - 48, 84t^2 + 66t - 44, 2t^3 - 26t^2 - 39t - 48)$