

涉密论文 ☐ 公开论文 ☐

浙 江 大 学

本科生毕业设计



题目 平面上物理区域的三次样条表示及布尔代数

学生姓名 贺思齐

学生学号 3150105217

指导教师 张庆海

年级与专业 2015级数学与应用数学

所在学院 数学科学学院

递交日期 2019/05/29

浙江大学本科生毕业论文（设计）承诺书

1. 本人郑重地承诺所呈交的毕业论文（设计），是在指导教师的指导下严格按照学校和学院有关规定完成的。

2. 本人在毕业论文（设计）中除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得浙江大学或其他教育机构的学位或证书而使用过的材料。

3. 与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

4. 本人承诺在毕业论文（设计）工作过程中没有伪造数据等行为。

5. 若在本毕业论文（设计）中有侵犯任何方面知识产权的行为，由本人承担相应的法律责任。

6. 本人完全了解浙江大学有权保留并向有关部门或机构送交本论文（设计）的复印件和磁盘，允许本论文（设计）被查阅和借阅。本人授权浙江大学可以将本论文（设计）的全部或部分内容编入有关数据库进行检索和传播，可以采用影印、缩印或扫描等复制手段保存、汇编本论文（设计）。

作者签名：

导师签名：

签字日期： 年 月 日 签字日期 年 月 日

致谢

作者对导师张庆海教授致以深深的感谢，感谢他对毕业设计的不断鼓励和指导。同时，他作为作者竺可桢学院的导师，在其大学的各个阶段，不论是学术还是生活，都不吝时间地对其进行启发和教导。

摘要

在本篇论文中作者详细讨论了计算任意一个三次有理数曲线集合的交点的算法^{[1][2]}，并将其在 CGAL 库的基础上，用 C++ 实现了出来。该算法适用于包括非交叉交点和与奇点相交的大多数情况。本论文正文有三个部分。第一部分是解决二元三次多项式方程组的求根问题。结式和无平方分解的方法被用于消去多项式的元以及计算根的重数。Descartes 符号法则被用于检验某个区间内是否存在根。以代数数为形式的根被边界为有理数的区间分离开。第二部分建立在 Bentley-Ottmann 扫描线算法之上，分析了三次曲线的几何性质以及将平面三次曲线分成可以扫描的部分的必要步骤和相关判据。更进一步，第二部分对直线段的扫描线算法进行了一些改进，以支持曲线的输入。第三部分首先简略地介绍了 CGAL 中的程序包，并进行了一些测试和实验，以检验这些方法的可行性和精确性。本次论文的目的在于协助约当曲线空间上的布尔运算。这一概念来自于一个为二维流体建模^[3]以处理多相流问题中的拓扑变化的模型。

Abstract

In this thesis the author discusses an algorithm computing intersection points of an arbitrary set of cubic rational curves^{[1][2]} and have it implemented in C++ via libraries in CGAL(The Computational Geometry Algorithm Library). The algorithm applies to most circumstances including improper intersections and intersections with singularities. There are three major ingredients. The first is a method to compute roots of cubic bivariate polynomial equations. Resultant and square-free factorization is used in eliminating two variables and counting the multiplicity of the roots. Descartes' rule of signs is applied to check if there exists a solution in a certain interval. The roots in the form of algebraic numbers are represented by intervals bounded by rational numbers. The second, based on the Bentley-Ottmann sweep line algorithm, analyzes the geometric features of cubic curves as well as necessary steps and corresponding criteria to break planar cubic curves into x-monotone sweepable components. Furthermore, some adjustments are made to make sweep line algorithm adaptable to the input of curves. A brief introduction of some packages in CGAL is given in the third ingredient. A few tests and experiments have been carried out and elaborated to verify the feasibility and accuracy of such schemes. The aim of this thesis is to facilitate the boolean operations on Jordan space, which is a concept in the mathematical model^[3] characterizing two-dimensional fluids in order to cope with topological changes in multiphase flows.

目录

第一部分 毕业设计

1 绪论	1
1.1 背景	1
2 平面上三次样条求交点算法	7
2.1 曲线交点的求解	7
2.2 利用区间进行实根隔离	10
2.3 曲线的分割	10
2.4 曲线扫描算法	11
3 算法实践与测试	12
3.1 CGAL 程序包	12
3.2 测试	13
4 不足与分析	14
5 参考文献	15
附录	17
作者简历	19
本科生毕业论文（设计）任务书	21
本科生毕业论文（设计）考核	23

第二部分 毕业设计开题报告

一、 文献综述	1
1 文献综述	1
1.1 殷空间上布尔代数的意义及基本理论	1
1.2 直线段交点的扫描算法	4
1.3 三次代数曲线求交点	6
2 参考文献	7

二、 开题报告	13
1 问题提出的背景	13
2 项目的主要内容和技術路线	14
2.1 研究的主要内容和技術路线	14
3 研究计划进度安排及预期目标	18
4 参考文献	19
三、 外文翻译	25
1 外文翻译——准确、有效、完备的曲线计算	25
1.2 代数基础	25
四、 外文原文	39
五、 中期检查	43
1 项目概况	43
2 工作进展情况	43
3 问题与建议	43
4 其他	43
毕业设计开题报告、外文翻译的考核	45
毕业设计中期报告考核	47

第一部分

毕业设计

1 绪论

1.1 背景

1.1.1 实体建模

在科学与工程领域，实体建模（Solid Modeling）是通过数学和计算机的算法，对二维或三维的实体进行模型的建立。与几何建模和计算机图形学不同的是，实体建模侧重于对实体物理性质的精确保留，并在此基础上实现创建实体、计算实体、以及实体可视化等一系列操作。

实体建模在科学与工程领域的应用非常广泛，除了传统工业如金属薄片加工，喷射造型法等的需求外，实体建模还通过利用样本点构建物理模型、利用有限元进行机械分析等手段，为快速成型技术、逆向工程等领域奠定了技术基础。

早期实体建模的重心在于利用几何模型在计算机上替代工程作图，以用来支持工程上的一些自动化的任务，包括几何设计、机械元件的可视化、和积分有关（如质量、体积、面积等）的性质的计算，以及一些机械运动的模拟等^[4]。

Requicha 在论文^[5]中对实体的一些物理性质进行了假设。所有机械加工元件都有有限的尺寸和性质较为良好的边界。最初的研究对象主要局限于由均匀的各向同性的材料制成的刚性物体。这些假定出的性质可以通过三维或二维欧式空间中一些特定子集的性质来表示。

元件有限的尺寸在欧式空间中对应有界的子集，刚性一般可以用旋转和平移下的一致性来表示。^[4]性质较为良好的边界在欧式空间中通常被表示为：该集合的边界可以被有限个分段光滑的映射函数表示。与此同时，集合应该在几种集合运算下封闭，而这些集合运算应该是物理上实物的添加和删除的规范化，如物体之间的相交可以对应于集合的交集。定义半解析集为包括所有可以以 $f_i(x, y, z) \geq 0$ 为形式的不等式的有限个组合，这里 f_i 是解析函数。半解析集合满足了以上所有要求。

1.1.2 殷集和布尔代数

与实体建模形成鲜明对比的是，流体建模（Fluid Modeling）领域相关的研究寥寥无几。而流体的拓扑变化在科学和工程界又起着必不可少的作用。在这样一个背景下，论文^[3]提出了建立在点集拓扑上的一种流体建模的理论。上文实体建模中提到的用半解析集来表示材料区域的数学问题同样也满足了为流体建立模型的需求。在论文中，殷空间和约当曲线偏序集空间的概念被引入，用以表示流体的边界。论文还指出半解析集的条件满足了与多相流、GIS 以及实体建模相关的问题。与此同时，此模型使用解析函数这类可以进行计算的方式来表示材料区域，从而将多相流和几何、拓扑等领域建立起了桥梁。

首先回顾一下正则开集意义下的布尔代数，^[6] 拓扑空间 X 中所有的正则开集可以被当做一个布尔代数，其中布尔代数的运算符为：

$$0 = \emptyset; \quad (1.1.1)$$

$$1 = X; \quad (1.1.2)$$

$$P \wedge Q = P \cap Q; \quad (1.1.3)$$

$$P \vee Q = (P \cup Q)^{\perp\perp}; \quad (1.1.4)$$

$$P' = P^{\perp}; \quad (1.1.5)$$

其中 \perp 指取外部运算，比如对于 P^{\perp} ， P^{\perp} 等于自己补集的内部。

接下来给出半解析集的严格定义^[4]：对于一个集合 S ，如果存在有限个解析函数 $g_i : \mathcal{R}^D \rightarrow \mathcal{R}$ ，使得 S 在 $X_i = \{x \in \mathcal{R}^D : g_i(x) \geq 0\}$ 生成的有限布尔代数中，那么 S 是半解析集。

利用半解析集来表示物体显然可以展现物体的边界表现良好的特点，但是半解析闭集存在一处局限：正则半解析闭集会让边界表示失去唯一性。为了解决这一不足，论文引入了有界的正则半解析开集这一概念，并将其定义为殷集。所有的殷集构成殷空间。而殷空间联合正则集上定义过的布尔运算，构成了一个布尔代数。

约当曲线偏序集空间概念的引入是为了用有限条曲线来刻画殷空间中的集合，论文^[3]定义了约当曲线偏序集空间，并证明了约当曲线偏序集空间上定义的运算和殷空间上的布尔代数运算同构，这样殷空间上的布尔运算就可以用曲

线的计算来代替。

约当曲线的定义：平面曲线是指形如 $\gamma: (0, 1) \rightarrow R^2$ 的连续映射。约当曲线是指满足 $\gamma(1) = \gamma(0)$ 的简单曲线 (simple curve)。

Spadjor 的定义：Spadjor 是指分段的几乎不相交的有向约当曲线组成的非空集合。

Atom Spadjor 的定义：Atom Spadjor 是指一个包含至多一个正向约当曲线 γ^+ 和有限个满足以下条件的负向约当曲线的 spadjor:

- (a). γ_j^- 在包含关系 (inclusion) 下是无法比较的。
- (b). $\gamma_l^- \prec \gamma^+$, $l = 1, 2 \dots n_-$ 。

这里的包含关系 (inclusion) 被定义为: γ_k 包含 γ_l 当且仅当 γ_l 的有界补集是 γ_k 的有界补集的子集。(b) 中的 \prec (covering of Jordan curves) 被定义为: 令 \mathcal{J} 为以包含为关系的约当曲线偏序集, 如果 $\gamma_l < \gamma_k$ 并且不存在 $\gamma \in \mathcal{J}$ 满足 $\gamma_l < \gamma < \gamma_k$, 那么称 γ_k 覆盖 (cover) γ_l 。

Realizable Spadjor 的定义：Realizable Spadjor $\mathcal{J} = \bigcup_k \mathcal{J}_k$ 是有限个不同的 atom spadjor 的并集。这里不同是指对于 $\mathcal{J}_i, \mathcal{J}_j$, $\rho(\mathcal{J}_i) \cap \rho(\mathcal{J}_j) = \Phi$, 运算 ρ 将曲线边界映射为集合内部: $\rho(\mathcal{J}_k) := \bigcap_{\gamma_i \in \mathcal{J}_k} \text{int}(\gamma_i)$ 。

正如上文所述, 引入约当曲线以及以上定义的目的在于将殷空间的表示实体化。论文^[3]中的定理 3.21 和推论 3.22 建立了约当曲线和殷集之间的关系: 不论殷集是连通的还是带有几个连通分支的, 都可以用约当曲线唯一地表示出来。这样不难证明将约当曲线映射为殷集的映射是一个双射。

为了使约当曲线空间与殷空间上的布尔运算构成一个同构, 应该有 complementation 和 meet 运算:

对于 realizable spadjor, 定义 complementation' 运算:

$$\mathcal{J}' := \begin{cases} \hat{1} & \text{if } \mathcal{J} = \hat{0} \\ \hat{0} & \text{if } \mathcal{J} = \hat{1} \\ (S_V^{-1} \circ R \circ S_V)\mathcal{J} & \text{otherwise} \end{cases} \quad (1.1.6)$$

两个 realizable spadjor 之间的运算符 $\text{meet}\wedge$ 定义为：

$$\mathcal{J} \wedge \mathcal{K} := \begin{cases} \hat{0} & \text{if } \mathcal{K} = \hat{0} \\ \mathcal{J} & \text{if } \mathcal{K} = \hat{1} \\ S_V^{-1}(E) & \text{otherwise} \end{cases} \quad (1.1.7)$$

上述两种运算的定义中， V 是约当曲线的非交叉交点。 R 是一个将约当曲线方向反过来的映射。 $S_V(\mathcal{J})$ 指 segmented realizable spadjor。 E 是一个有向多重图的边的集合。关于 E 的具体构建可以参考^[3]。

在这样的运算定义下，约当曲线偏序集空间和运算符共同构成了一个布尔代数。本次研究的主要内容就是建立在布尔代数的构造上的。

1.1.3 直线段交点的扫描算法

直线段交点的扫描算法是求解曲线段扫描算法的基础。在计算几何领域中，用算法求直线交点是一个非常重要的基本问题。如下的算法将输入的直线段集合进行两两遍历，耗时 $O(n^2)$ ，这样的做法显然不可取：

Algorithm 1 寻找直线的交点

Require: 平面上直线集合 S

Ensure: 直线的交点集合

```

for  $\forall e_i, e_j \in S$  do
    if  $e_i$  和  $e_j$  相交 then
        输出交点
    end if
end for
    
```

求解直线段交点的另外一种高效的方法就是扫描算法。历史上直线段扫描算法是 Jon Bentley 和 Thomas Ottmann 于 1979 年提出的。在^[7]介绍的扫描算法具体操作中，一条想象出来的线段从平面的一侧扫描到另一侧，这条扫描线段的状态定义为线段与考察直线段的交点。每次遇到线段的初始端点，就将该线段纳入考虑范围。每次遇到线段的末尾端点，就将线段删除。线段的初始端点、交点和末尾端点被称为线段的事件点。

扫描线段的过程不是连续的。在实际操作中，扫描线段是从一个事件点跳到另一个事件点。每一次到一个新的事件点，程序都会执行相应的操作（如遇到末端点就会删除相应的线段），并更新状态结构。初始状态下，所有直线的端点依照 y 坐标（如果扫描线段是水平线）的大小放入优先队列中。具体步骤如下^[7]：

Algorithm 2 寻找直线的交点

Require: 平面上直线集合 S

Ensure: 直线的交点集合

初始化事件队列 Q ，将线段端点依照 y 坐标顺序放入堆中

初始化状态结构 τ

while Q 非空 **do**

 寻找下一个事件点

 处理事件点

end while

Algorithm 3 处理事件点

令 $U(p)$ 为起始点是 p 的线段

找到存储在 τ 中包含 p 的线段。令 $L(p)$ 表示末端端点是 p 的线段，令 $C(p)$ 代表包含 p 的线段

if $L(p) \cup U(p) \cup C(p)$ 包含不止一个线段 **then**

 那么 p 是一个交点

end if

将 $L(p) \cup C(p)$ 中的线段从 τ 中删除

将 $U(p) \cup C(p)$ 中的线段插入到 τ 。

if $U(p) \cup C(p) = \emptyset$ **then**

 找到下一个事件点

end if

优先队列可以用堆或者平衡二叉搜索树这样的数据结构来实现，从而将运算时间从 $O(n^2)$ 降低到 $O(n \log n)$ 。在一般操作中，往往使用平衡二叉搜索树。因为在执行扫描线算法时，需要找到某个已有事件进行修改。在平衡树中搜索节

点的复杂度是 $O(\log n)$ ，而对于堆来说，查找非最小节点的复杂度是 $O(n)$ 。因此对于有寻找非最小节点需求的扫描线算法来说，平衡二叉搜索树更快。

普通的扫描算法通过 $O(n \log n)$ 的复杂度搜寻直线段的交点，这一点比大多数算法已经快很多了。但是普通的扫描线算法没有处理非交叉相交和线段重合的情况。^[8] 中提供了一种扫描算法的变体，解决了上述问题。该文献将部分重合的直线段用集合表示，并定义其为直线束 (**bundle**)，从而将研究直线交点的问题转化为研究直线束的交点的问题，使新算法对重合的线段也具有适用性。

2 平面上三次样条求交点算法

平面上三次样条求交点是求解殷集交集的一个重要步骤。一般来说表示平面曲线有三种方法：

- 函数形式： $\{(x, y) \in \mathbb{R}^2 : y = f(x)\}, \quad f : \mathbb{R} \rightarrow \mathbb{R}$

这种形式最为特殊，对于每个 x 都对应唯一一个 y ，这样的形式无法用一个方程表示闭合的曲线

- 参数形式： $\{(x, y) \in \mathbb{R}^2 : x = f_1(t), y = f_2(t), t \in (R)\}, \quad f_1, f_2 : \mathbb{R} \rightarrow \mathbb{R}$

这种形式可以表示平面上的闭合曲线，解决一部分函数形式无法解决的问题。但是参数形式的样条无法表示平面上包括圆在内的很多基本几何图形，于是就有了以下的一种表示形式。

- 隐函数形式： $\{(x, y) \in \mathbb{R}^2 : f(x, y) = 0\}, \quad f : \mathbb{R}^2 \rightarrow \mathbb{R}$

出于以上考虑以及实际算法操作的方便性，本次毕业设计的主要研究对象选用隐函数形式的三次样条。即对于三次样条曲线集合 $S := \{f | f \in \mathbb{Q}[x, y], \deg(f) \leq 3\}$ ，可以计算出所有曲线两两之间的交点，同时省去将所有曲线进行 $O(n^2)$ 次遍历的麻烦。为了实现这一目的，在选取曲线进行考察是否相交时，采用与计算直线集合交点相类似的算法 **Bentley-Ottmann 算法**^[9]，假想一条曲线从 $x = -\infty$ 扫描到 $x = \infty$ ，在扫描的过程中记录每一个扫到的曲线对的相交情况以及相交点（如果有）。这样的设计可以保证每一个交点都被扫描到，并且算法的复杂度可以从 $O(n^2)$ 降低到 $O((n + k) \log n)$ ^[9]，这里 n 是输入的曲线数， k 是这些曲线的交点数。

在求解曲线交点的时候，本设计采用了 Arno Eigenwillig 等人总结的对三次样条进行处理的方法^[1]和 CGAL 算法库（The Computational Geometry Algorithms Library）。具体会在下列小节中一一陈述。

2.1 曲线交点的求解

CGAL 算法库的编写者 Eric Berberich 和他的同事们在发表的论文^[10]中对三次二元多项式方程组的求根进行了详尽的介绍。算法的大致思路是将二元多项式

中的一个未知元作为系数来看待，进而将二元多项式投射为一个一元多项式，并在此基础上计算出一个包含多项式解的集合。文章中提出了利用结式（Resultant）计算进行消元，然后对于消元后解出的解集进行检验，以确定其是否为消元前原方程组的解。具体见下文的描述。

2.1.1 曲线的无平方分解

在代数中，无平方多项式（square-free polynomial） $f \in F[x]$ 是指不包含任意一个非常量因式的平方的多项式。对多项式无平方分解是对多项式进行操作的基础。在求解曲线交点的过程中，无平方分解是很重要的一环。对多项式进行无平方分解的算法有很多，Yun 算法（Yun's algorithm）^[11] 通过计算 GCD 和导数，实现对一元多项式进行无平方分解。

Yun 算法的设计思路如下：

如果 $f = a_1 a_2^2 a_3^3 \dots a_k^k$ 是所求得的分解，我们可以得到

$$a_0 = a_2^1 a_3^2 \dots a_k^{k-1}$$

$$\frac{f}{a_0} = a_1 a_2 a_3 \dots a_k$$

以及

$$\frac{f'}{a_0} = \sum_{i=1}^k i a_i' a_1 \dots a_{i-1} a_{i+1} \dots a_k$$

如果令 $b_1 = \frac{f}{a_0}$ ， $c_1 = \frac{f'}{a_0}$ 以及 $d_1 = c_1 - b_1'$ ，可以得到

$$\gcd(b_1, d_1) = a_1$$

$$b_2 = b_1 / a_1 = a_2 a_3 \dots a_k$$

$$c_2 = d_1 / a_1 = \sum_{i=2}^k (i-1) a_i' a_2 \dots a_{i-1} a_{i+1} \dots a_k$$

重复这一步，直到 $b_{k+1} = 1$ ，这样可以得到所有的 a_i 。在这样的思路下，Yun 算法可以表示成：由 Yun 算法将多项式进行无平方分解对下述的消元、求解根的重数以及对根进行分离方面会产生很大的帮助。

Algorithm 4 Yun 无平方分解

 $a_0 := \gcd(f, f'); b_1 := f/a_0; c_1 := f'/a_0; d_1 := c_1 - b'_1; i := 1;$

Repeat

 $a_i := \gcd(b_i, d_i); b_{i+1} := b_i/a_i; c_{i+1} := d_i/a_i; i := i + 1; d_i := c_i - b'_i;$

 直到 $b = 1;$

 输出 a_1, \dots, a_{i-1} 。

2.1.2 用结式矩阵消元

用西尔维斯特矩阵可以对二元多项式进行消元，并将求解二元多项式方程组问题转化为一个求解两次一元方程组的问题。对于一元多项式来说， $f = \sum_{i=0}^m a_i x^i$, $a_m \neq 0$ 和 $g = \sum_{i=0}^n b_i x^i$ 的西尔维斯特 (Sylvester) 矩阵是一个 $(n+m) \times (n+m)$ 形如下的系数矩阵^[1]:

$$\begin{bmatrix} a_m & \dots & \dots & a_0 & & \\ & \dots & & & \dots & \\ & & a_m & \dots & \dots & a_0 \\ b_n & \dots & \dots & b_0 & & \\ & \dots & & & \dots & \\ & & b_n & \dots & \dots & b_0 \end{bmatrix}$$

西尔维斯特矩阵在代数中的作用非常显著，它的行列式 (resultant) 可以用来判断两多项式是否存在共同解。Ideals, Varieties, and Algorithms^[12] 一书中对此进行了详细的描述:

定理: 给定多项式 $f, g \in k[x]$, 那么结式 $\text{res}(f, g, x)$ 是一个由 f 和 g 的系数构成的多项式。进一步地讲, f 和 g 有共同的因子当且仅当 $\text{res}(f, g, x) = 0$ 。

而对于二次多项式, 可以将一个未知元当做多项式系数的一部分带入西尔维斯特矩阵中, 这样依据上述定理求解令西尔维斯特矩阵多项式等于 0 时, 对应的解。论文^[1] 给出了利用结式行列式进行消元的严格定理和证明:

定理: 令 K 为一个域, 令 $f, g \in K[x, y]$ 为非零多项式。令 f 为 y 正规的 (即 f 包含一项 $cy^{\deg f}$, $c \in R, c \neq 0$), 那么下面两个条件等价:

1. $\text{res}(f, g, y)(\xi) = 0$
2. 存在 $\eta \in \bar{K}$ 使得 $f(\xi, \eta) = g(\xi, \eta) = 0$

上述定理可以求出一个包含原方程组解的解集。原方程解可以通过带入该解集来考察。除此之外，该定理要求多项式一定是 y 正规的，但是在实际求解的过程中，可以通过坐标变换的方式将非 y 正规的曲线转化为 y 正规的曲线。

2.2 利用区间进行实根隔离

在计算机中，代数数或者无理数往往要涉及到符号表示，这样往往会消耗内存。对此的一个解决方案是利用以有理数为边界的区间来表示此类数。在求解多项式根的过程中，我们采用实根隔离的方法——Descartes 算法^[1]。

Descartes 符号法则： 令 $f \in R[x]$ 为一个有 p 个正实根（包括重数）和 v 个系数符号变化的非零多项式，那么 $v - p$ 是非负偶数。

Descartes 符号法则可以用来大致估计多项式有几个正解几个负解。而在利用该法则进行区间隔离时，如果 $v = 0$ ，那么此区间上一定无根，如果 $v = 1$ ，那么区间中一定有且仅有一个根。因此进行实根分离时只需要以 v 等于 1 或 0 为判据。用 Descartes 符号法则在研究函数 f 在任意区间 (l, r) 上的根的情况时需要进行莫比乌斯变换 $T(x) = \frac{ax+b}{cx+d}$ ，即寻找 g ，使得 $g \circ T = f$ 且 g 上的 $(0, +\infty)$ 区间被映射到 f 的 (l, r) 区间上，进而在该区间上利用 Descartes 符号法则进行分析。由于多项式的根是有限的，所以在经历一定的迭代次数后，一定可以将所有的根用区间分离开来。

利用区间进行实根分离之前应该先对多项式进行消元和无平方分解，这样在保证可以求得 $v = 0$ 或 $v = 1$ 的区间的同时，还可以计算出每一个根的重数。为下文曲线的分割提供了非常大的帮助。

2.3 曲线的分割

曲线扫描与直线扫描的不同之一是，隐函数表示的曲线可以不是一个 x 对应唯一一个 y 的形式，这样在一条直线扫过一条曲线时，扫描状态可能会发生改变，即直线与这条曲线的交点数可能不唯一，或者可能在扫描的过程中发生变化。因此，让曲线扫描成为可行的算法的一个必要操作是将曲线分割成可以扫描

的曲线段 (arc)。对此, 论文^[1] 提出了将曲线“分成可以扫描的曲线段”的严格定义和操作步骤。

将曲线分割需要找到让扫描状态发生变化的点——关键点 (critical point), 以下是两种关键点的定义:

Singularity: 满足 $f_y = f_x = 0$ 的点, 几何上任意多个曲线段 (arc) 在这里汇合, 即曲线具有自相交交点 (self-intersection)。

x-Extremality: 满足 $f_y = 0, f_x, f_{yy} \neq 0$ 的点, 几何上一个曲线的两个曲线段 (arc) 在这里汇合, 两个曲线段都在该点切线的一侧。

Vertical flex: 满足 $f_y = f_{yy} = 0, f_x \neq 0$, 一个曲线的两个曲线段在这里汇合, 两个曲线段在该点的两侧。

论文^[1] 给出了关于分割曲线的关键点 (critical point) 的引理:

引理: 令 $f \in K[x, y]$ 为一个 y 正规的代数曲线, 令 $(r, s) \in K^2$, 那么下列两者等价:

(i). (r, s) 是 f, f_y 的交点。

(ii). 多项式 $f(r, y) \in K[y]$ 在 $y = s$ 上有重根。

多项式 $f(r, y) \in K[y]$ 在 $y = s$ 上有重根的情况对应着关键点 (critical point) 中的 x-extremality 和 vertical flex 点, 因此在寻找关键点时, 可以利用上述求解曲线交点的算法, 将 f, f_y 的交点求出。

2.4 曲线扫描算法

曲线扫描算法是建立在 Bentley-Ottmann 的扫描算法上的。首先输入一个三次样条曲线集合 $S := \{f | f \in \mathbb{Q}[x, y], \deg(f) \leq 3\}$, 在预处理时将其依据单曲线事件点 (one-curve event point) 分为多段弧 (arc)。在将曲线拆分为多段弧的过程中, 算法同时也会检验曲线是否符合其输入条件, 如是否对 y 正规, 曲线是否有平方项等。如果存在不满足条件的曲线, 可以人工进行坐标变化或无平方分解后, 再将曲线输入。将曲线拆分完毕后, 用垂直于 x 轴的扫描线从左向右扫过平面。与扫描线在当下相交的所有曲线都按照纵坐标轴的顺序存储在 Y 结构中, 用 X 结构来存储所有已知的事件点 (字母顺序)。然后应用类似直线的扫描算法:

1. 从 X 结构中取出下一个事件点, 找到其在 Y 结构中相关联的曲线段。

这一步的操作需要比较相关点和线段的 y 坐标，可以通过比较不同曲线在横坐标等于相关点横坐标时，纵坐标的大小来决定。

2. 删掉结束扫描的曲线段。

这一步需要判断正在处理的点是否是曲线段的端点（target point），比较坐标关系即可。

3. 依据交点的重数改变尚未扫描的曲线在 Y 结构中的顺序。

之前计算结式矩阵的步骤保证了两条曲线是互质的。因此若两条曲线段重合，当且仅当它们所在的曲线相同，以及它们对应的 X 轴区间也相同。

4. 将即将开始扫描的新曲线按照顺序加入 Y 结构。

这里需要比较两条曲线段之间的位置关系，如果两条曲线段属于同一条曲线，那么比较两曲线的位置关系等同于比较其曲线段（或弧）的序号（在几何分析中就已经得出）；如果两条曲线段属于两条不同的曲线，那么可以比较交点右侧一个足够小的区间的纵坐标大小。

5. 将新发现的事件点加入 X 结构。

完成这一步需要比较事件点的坐标关系。先比较横坐标的大小，如果大小一样再比较纵坐标的大小。

论文^[1]第四节可以说明输入的曲线之间应该满足一些条件：如 f 和 g 的事件点不能处于同一竖直线， f 和 g 不能在上文所述的第二种关键点（ x -extremality）处相交等。这些条件是由算法本身决定的。在人工输入时，对于不满足条件的曲线，可以采取坐标旋转的方式解决问题。

3 算法实践与测试

3.1 CGAL 程序包

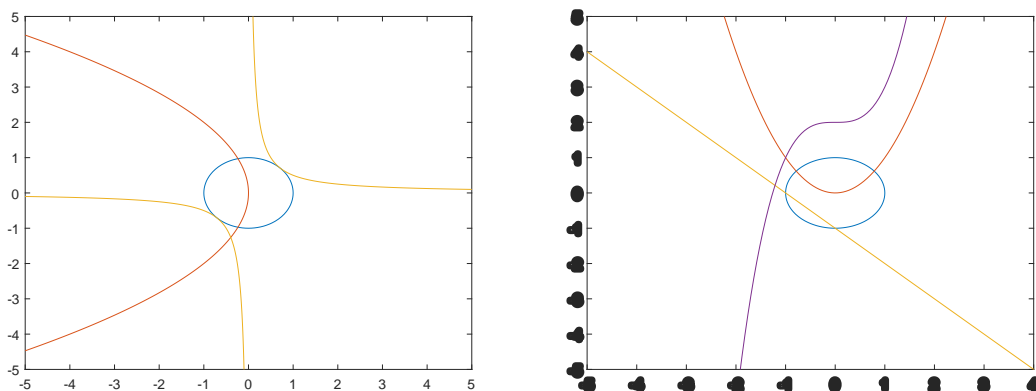
本次设计的代码建立在 CGAL 算法库（The Computational Geometry Algorithms Library）的库函数上。CGAL 的 Arithmetic and Algebra 程序包中的 Polynomial<Coeff> 类提供了构建多元多项式，求解多项式的 GCD 等成员函数。

2D Arrangement 程序包提供的平面代数曲线特征类 (*Arr_algebraic_segment_traits_2* \langle *Coefficient* \rangle) 可以将 *Polynomial* 类中的多项式转化为曲线形式, 进而表示平面上任意次数的隐函数形式的曲线, 并提供了与这种曲线相对应的点 *Point_2* 类型。多项式的系数由 *Coefficient* 输入, 目前 *Coefficient* 支持 CGAL 库中 *leda_integer*, *CORE :: BigInt* 和两种类型的输入。这两种类型均是 CGAL 库内定义的整数类型。因此输入的多项式系数必须是整数。由每一个有理数都可以写作两个整数的商这一事实, 可以人工地将研究曲线扩充到有理数为系数的多项式。

算法输出的交点即是 *Point_2* 类型的点。其中存储有交点坐标, 和所在曲线的曲线段 (arc) 的编号等信息。点的坐标是一个 *Algebraic_real_1* 的 *private* 变量, 但可以通过类内的成员函数转化为 *double* 类型。

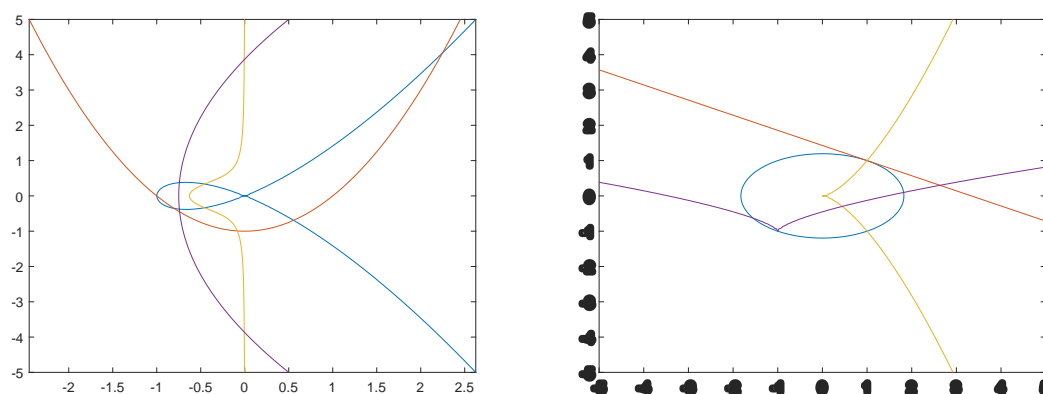
3.2 测试

本次毕业设计在 CGAL 算法库的基础上用 C++ 实现了计算平面上以有理数为系数的三次多项式曲线集合的交点的算法。程序构建了一个新的命名空间 *CurvesArrangement*, 在命名空间中定义了 *CurvesIntersector* class, 并进行了如下测试:



test1: $f_1 = x^2 + y^2 - 1$, $f_2 = 4 * x + y^2$, $f_3 =$ **test2:** $f_1 = x^2 + y^2 - 1$, $f_2 = y - x^2$, $f_3 = (x + y)^2 - (x - y)^2 - 2$, 得到 5 个交点 $x + y + 1$, $f_4 = y - x^3 - 2$, 共得到 6 个交点

图 3.1: 几种正常相交的处理



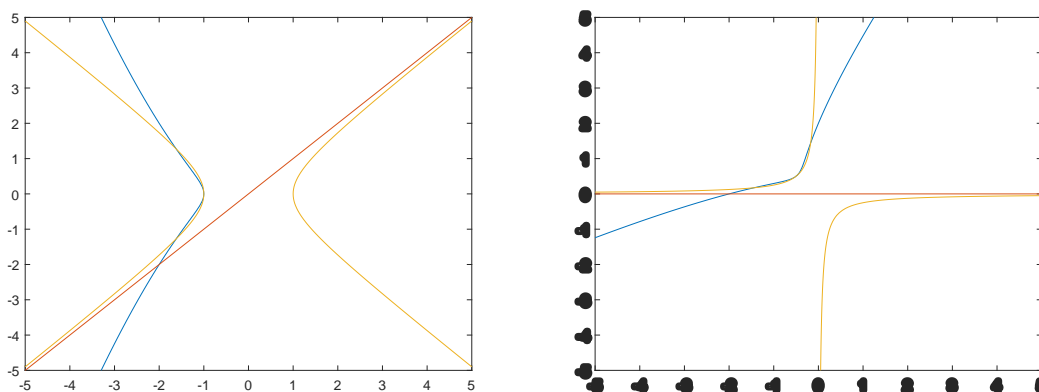
test3: $f_1 = y^2 - x^3 - x^2$, $f_2 = y - x^2 + 1$, $f_3 =$ **test4:** $f_1 = 3 * x^2 + 7 * y^2 - 10$, $f_2 = 14 * y + 6 * 2 * (x + y)^3 + 2 * (x - y)^3 + 1$, $f_4 = y^2 - 20 * x - 15$, $x - 20$, $f_3 = x^3 - y^2$, $f_4 = (x + 1)^2 - 6 * (y + 1)^3$,
共得到 15 个交点 共得到 7 个交点

图 3.2: 几种正常相交的处理

test5 和 test6 展示了曲线于第二种关键点处相交的情况，这时可以通过进行一个旋转变换求出交点。

4 不足与分析

本次毕业设计存在的不足在于，由于时间较为仓促，没有写出检测自相交情况（self-intersection）的程序。然而 CGAL 的 *Polynomial_d* 类中提供了对 *Polynomial₂* 类型的多项式进行求导的函数，以此同时加上求解多项式根的算法，可以说求解奇点（singularity）是可行的。



test5: $f_1 = y^2 + x^2 + x^3$, $f_2 = x - y$, $f_3 = x^2 - y^2 - 1$, 共得到 3 个交点
test6: $f_1 = (x+y)^2 + (x-y)^2 + (x-y)^3$, $f_2 = y$, $f_3 = (x-y)^2 - (x+y)^2 - 1$, 将 test5 进行旋转后, 检测出了全部交点

5 参考文献

- [1] EIGENWILLIG A, KETTNER L, SCHÖMER E, et al. Exact, efficient, and complete arrangement computation for cubic curves[J]. Comput. Geom., 2006, 35: 36–73.
- [2] BERBERICH E, FOGEL E, HALPERIN D, et al. Sweeping and maintaining two-dimensional arrangements on surfaces[J]. Aichholzer, Oswin; Hackl, Thomas: Collection of Abstracts of the 23rd European Workshop on Computational Geometry, Technische Universität Graz, 223-226 (2007), 2007.
- [3] ZHANG Q, LI Z. Fluid Modeling and Boolean Algebra for Arbitrarily Complex Topology in Two Dimensions[Z]. under review, <https://arxiv.org/abs/1902.06378>.
- [4] SHAPIRO V. Solid modeling[J/OL]. 2001. <https://pdfs.semanticscholar.org/f18f/58d3938542c37d865f8a09e1777beba02c29.pdf>.
- [5] Requicha, Voelcker. Solid Modeling: A Historical Summary and Contemporary Assessment[J]. IEEE Computer Graphics and Applications, 1982, 2: 9–24.
- [6] GIVANT S, HALMOS P. Introduction to Boolean Algebras[M]. [S.l.]: [s.n.], 2009. DOI: 10.1007/978-0-387-68436-9.

- [7] BERG M D, CHEONG O, KREVELD M V, et al. Computational Geometry: Algorithms and Applications[M]. 3rd ed. Santa Clara, CA, USA: Springer-Verlag TELOS, 2008. ISBN: 3540779736, 9783540779735.
- [8] LI Z. A Variant of the Line Sweep Algorithm with User-defined Uncertainty Quantification[C]//. [S.l.]: [s.n.], 2018.
- [9] BERG M D, CHEONG O, KREVELD M V, et al. Computational Geometry: Algorithms and Applications[M]. 3rd ed. Santa Clara, CA, USA: Springer-Verlag TELOS, 2008. ISBN: 3540779736, 9783540779735.
- [10] BERBERICHE, EMELIYANENKO P, SAGRALOFF M. An Elimination Method for Solving Bivariate Polynomial Systems: Eliminating the Usual Drawbacks[J/OL]. ALENEX '11 2011: 35–47. <http://dl.acm.org/citation.cfm?id=2790248.2790252>.
- [11] YUND Y. On Square-free Decomposition Algorithms[J/OL]. 1976: 26–35. <http://doi.acm.org/10.1145/800205.806320>.
- [12] COX D A, LITTLE J, O'SHEA D. Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3/e (Undergraduate Texts in Mathematics)[M]. Berlin, Heidelberg: Springer-Verlag, 2007. ISBN: 0387356509.

附录

作者简历

姓名：贺思齐性别：女民族：汉

出生年月：1997-03-25 籍贯：河南省三门峡市

2012.09-2015.07 郑州外国语学校

2015.09-2019.07 浙江大学攻读学士学位

获奖情况：学业三等奖学金

参加项目：UC Davis 暑期科研项目、浙江省省级大学生科研创新项目

本科生毕业论文（设计）任务书

一、题目：

二、指导教师对毕业论文（设计）的进度安排及任务要求：

起讫日期 20 年 月 日 至 20 年 月 日

指导教师（签名）_____ 职称 _____

三、系或研究所审核意见：

负责人（签名）_____

年 月 日

本科生毕业论文（设计）考核

一、指导教师对毕业论文（设计）的评语：

指导教师（签名）_____

年 月 日

二、答辩小组对毕业论文（设计）的答辩评语及总评成绩：

成绩 比例	中期报告 (10%)	开题报告 (15%)	外文翻译 (5%)	毕业设计质量 及答辩 (70%)	总评 成绩
分值	10	15	5	70	100

负责人（签名）_____

年 月 日

第二部分

毕业设计开题报告

浙 江 大 学

本 科 生 毕 业 设 计

开题报告



姓名与学号	贺思齐 3150105217
指导教师	张庆海
年级与专业	2015级数学与应用数学
所在学院	数学科学学院

一、题目：_____平面上物理区域的三次样条表示及布尔代数_____

二、指导教师对开题报告的具体要求：

指导教师（签名）_____

年 月 日

一、文献综述

1 文献综述

与本项目相关的文献主要有以下三个方面：

1. 殷空间上布尔代数的意义及基本理论；
2. 直线段交点的扫描算法；
3. 三次代数曲线求交点；

1.1 殷空间上布尔代数的意义及基本理论

布尔代数的概念首先由乔治·布尔（1847）在数理逻辑中提出。而后代数学的发展推动了人们对布尔代数的进一步研究。从格论的角度讲，布尔代数是一个带有补集运算的有界分配格。直观地说，布尔代数是指以下形式的一种代数：

$$B := (\mathbb{B}, \vee, \wedge, ', \hat{0}, \hat{1}), \quad (1.1.1)$$

其中二元运算 \vee, \wedge ，一元运算 $'$ (称为补运算) 和零元运算 $\hat{0}, \hat{1}$ 满足：

1. $x \wedge \hat{1} = x, x \vee \hat{0} = x,$
2. $x \wedge x' = \hat{0}, x \vee x' = \hat{1},$
3. $x \vee y = y \vee x, x \wedge y = y \wedge x,$
4. $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z), x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$

1.1.1 布尔代数在多边形运算和正则开集上的应用

布尔代数除了在工程领域的作用外还应用于数学本身。其中一个例子就是在正则开集上的应用。对正则开集定义 $\vee, \wedge, '$ 运算，可以证明这样的运算和正则开集构成的空间组成了布尔代数。

Steven Givant 和 Paul Halmos^[1] 在书中将正则开集意义下的布尔代数解释如下：

拓扑空间 X 中所有的正则开集可以被当做一个布尔代数，其中布尔代数的运算符为：

$$0 = \emptyset; \quad (1.1.2)$$

$$1 = X; \quad (1.1.3)$$

$$P \wedge Q = P \cap Q; \quad (1.1.4)$$

$$P \vee Q = (P \cup Q)^{\perp\perp}; \quad (1.1.5)$$

$$P' = P^{\perp}; \quad (1.1.6)$$

其中 \perp 指取外部运算，比如对于 P^{\perp} ， P^{\perp} 等于自己补集的内部。

布尔代数之所以会运用到正则开集上，一个很重要的原因就是正则开集具有一定的物理意义。^[1] 不论是正则开集还是正则闭集，都具有一个作为材料区域的很重要的特点：正则集的定义可以免去很多低维的元素，如二维空间中的孤立点和曲线，三维空间中孤立出来的面。

然而正则集在表示现实中的材料区域时还存在一定的局限性：有些正则集无法被有限数量的符号表示。比如说，如果一个集合有处处不可导的边界，那么它们往往由无限个连续函数组成。再比如说，两个正则集的交也可能产生一个只能用无限个连续函数表示的边界。如：

$$\mathcal{A}_p := \{(x, y) \in \mathcal{R}^2 : -2 < y < \frac{1}{\sin x}, 0 < x < 1\} \quad (1.1.7)$$

$$\mathcal{A}_s := \{(x, y) \in \mathcal{R}^2 : 0 < y < 1, -1 < x < 1\} \quad (1.1.8)$$

上述定义的两个集合的交是无限个互不相交的正则集。这可以说明，如果想用有限的连续函数来表示物理区域，对于集合应该加比正则集更强的条件。

1.1.2 布尔代数在殷空间上的应用

流体的拓扑变化在科学界是一个很重要的研究话题。而上文提到的用有限个连续函数来表示材料区域的数学问题满足了为流体建立模型的需求。^[2] 引入了

殷空间的概念，并给出了满足以上条件的集合的严格定义——半解析集。论文^[2]指出半解析集的条件满足了与多相流、GIS 以及实体建模相关的问题。与此同时，此模型使用解析函数这类可以进行计算的方式来表示材料区域，从而将多相流和几何、拓扑等领域建立起了桥梁。

根据^[2]的定义，对于一个集合 S ，如果存在有限个解析函数 $g_i : \mathcal{R}^D \rightarrow \mathcal{R}$ ，使得 S 在 $X_i = \{x \in \mathcal{R}^D : g_i(x) \geq 0\}$ 生成的有限布尔代数中，那么 S 是半解析集。

但是半解析闭集存在一处局限：正则半解析闭集会让边界表示失去唯一性。为了解决这一不足，论文定义殷集为一个边界有界的正则半解析开集。所有的殷集构成殷空间。而殷空间联合正则集上定义过的布尔运算，构成了一个布尔代数。

1.1.3 殷空间上布尔代数的基本理论

在介绍基本理论之前，应当先引入约当曲线偏序集空间的概念。引入约当曲线的目的在于将殷空间的表示实体化。论文^[2]中的定理证明建立了约当曲线和殷集之间的关系：不论殷集是连通的还是带有几个连通分支的，都可以用约当曲线唯一地表示出来。这样不难证明将约当曲线映射为殷集的映射是一个双射。

接下来论文定义了 **complementation** 和 **meet** 运算，将约当曲线空间变成一个有界分配格。这些运算和殷空间上的布尔运算是同构的。

对于 **realizable spadjor**，定义 **complementation** ' 运算：

$$\mathcal{J}' := \begin{cases} \hat{1} & \text{if } \mathcal{J} = \hat{0} \\ \hat{0} & \text{if } \mathcal{J} = \hat{1} \\ (S_V^{-1} \circ R \circ S_V)\mathcal{J} & \text{otherwise} \end{cases} \quad (1.1.9)$$

两个 **realizable spadjor** 之间的运算符 **meet** 的定义：

$$\mathcal{J} \wedge \mathcal{K} := \begin{cases} \hat{0} & \text{if } \mathcal{K} = \hat{0} \\ \mathcal{J} & \text{if } \mathcal{K} = \hat{1} \\ S_V^{-1}(E) & \text{otherwise} \end{cases} \quad (1.1.10)$$

上述两种运算的定义中， V 是约当曲线的非交叉交点。 R 是一个将约当曲线方向反过来的映射。 $S_V(\mathcal{J})$ 指 segmented realizable spadjor。 E 是一个有向多重图的边的集合。关于 E 的具体构建可以参考^[2]。

在这样的运算定义下，约当曲线偏序集空间和运算符共同构成了一个布尔代数。本次研究的主要内容就是建立在布尔代数的构造上的。

1.2 直线段交点的扫描算法

在计算几何领域中，用算法求直线交点是一个非常重要的基本问题。最直接的算法如下：

Algorithm 5 寻找直线的交点

Require: 平面上直线集合 S

Ensure: 直线的交点集合

```

for  $\forall e_i, e_j \in S$  do
    if  $e_i$  和  $e_j$  相交 then
        输出交点
    end if
end for
    
```

这样的算法会浪费很多不必要的时间。因此，求解直线段交点的另外一种高效的方法就是扫描算法。在^[3]介绍的扫描算法具体操作中，一条想象出来的线段从平面的一侧扫描到另一侧，这条扫描线段的状态定义为线段与考察直线段的交点。每次遇到线段的初始端点，就将该线段纳入考虑范围。每次遇到线段的末尾端点，就将线段删除。线段的初始端点、交点和末尾端点被称为线段的事件点。

扫描线段的过程不是连续的。在实际操作中，扫描线段是从一个事件点跳到另一个事件点。每一次到一个新的事件点，程序都会执行相应的操作（如遇到末尾端点就会删除相应的线段），并更新状态结构。初始状态下，所有直线的端点依照 y 坐标（如果扫描线段是水平线）的大小放入优先队列中。具体步骤如下^[3]：

优先队列可以用堆或者平衡二叉搜索树这样的数据结构来实现，从而将运算时间从 $O(n^2)$ 降低到 $O(n \log n)$ 。在一般操作中，往往使用平衡二叉搜索树。

Algorithm 6 寻找直线的交点

Require: 平面上直线集合 S

Ensure: 直线的交点集合

初始化事件队列 Q ，将线段端点依照 y 坐标顺序放入堆中

初始化状态结构 τ

while Q 非空 **do**

 寻找下一个事件点

 处理事件点

end while

Algorithm 7 处理事件点

令 $U(p)$ 为起始点是 p 的线段

找到存储在 τ 中包含 p 的线段。令 $L(p)$ 表示末端端点是 p 的线段，令 $C(p)$ 代表包含 p 的线段

if $L(p) \cup U(p) \cup C(p)$ 包含不止一个线段 **then**

 那么 p 是一个交点

end if

将 $L(p) \cup C(p)$ 中的线段从 τ 中删除

将 $U(p) \cup C(p)$ 中的线段插入到 τ 。

if $U(p) \cup C(p) = \emptyset$ **then**

 找到下一个事件点

end if

因为在执行扫描线算法时，需要找到某个已有事件进行修改。在平衡树中搜索节点的复杂度是 $O(\log n)$ ，而对于堆来说，查找非最小节点的复杂度是 $O(n)$ 。因此对于有寻找非最小节点需求的扫描线算法来说，平衡二叉搜索树更快。

普通的扫描算法通过 $O(n \log n)$ 的复杂度搜寻直线段的交点，这一点比大多数算法已经快很多了。但是普通的扫描线算法没有处理非交叉相交和线段重合的情况。^[4] 中提供了一种扫描算法的变体，解决了上述问题。该文献将部分重合的直线段用集合表示，并定义其为直线束 (bundle)，从而将研究直线交点的问题转化为研究直线束的交点的问题，使新算法对重合的线段也具有适用性。

1.3 三次代数曲线求交点

^[2] 证明了在定义 meet 和 complementation 运算后，约当曲线空间可以构成一个布尔代数。因此，如果用三次样条函数去模拟物理区域的边界，上述文献中的定理可以保证按照这种方式模拟的物理区域在进行运算时具有较好的性质。在对三次样条表示的区域进行布尔运算时，需要用到求解三次代数曲线交点的算法，这即是本次毕业设计的核心问题。我们希望能够找到一个算法，使得对于任意的输入集合 $S = \{f_1, f_2, \dots, f_n\}$ ，其中 $f_i \in \mathbb{Q}[x, y]$ ，算法都可以输出 S 集中所有曲线的交点，并对输出交点的类型（如奇点或普通交点）进行分类。

三次代数曲线求交点与直线求交点相比多了很多困难，如一条曲线可能会和一条扫描线有两个交点，因此需要进行严格的分类和拆分；再如有些曲线的交点无法显式表示；或无理数表示可能存在精度或者效率上的不足，等等。三次曲线求交点是本次研究的核心内容，具体问题会在开题报告中详述。

2 参考文献

- [1] GIVANT S, HALMOS P. Introduction to Boolean Algebras[M]. [S.l.]: [s.n.], 2009. DOI: 10.1007/978-0-387-68436-9.
- [2] ZHANG Q, LI Z. Fluid Modeling and Boolean Algebra for Arbitrarily Complex Topology in Two Dimensions[Z]. under review, <https://arxiv.org/abs/1902.06378>.
- [3] EIGENWILLIG A, KETTNER L, SCHÖMER E, et al. Exact, efficient, and complete arrangement computation for cubic curves[J]. Comput. Geom., 2006, 35: 36–73.
- [4] LI Z. A Variant of the Line Sweep Algorithm with User-defined Uncertainty Quantification[C]//. [S.l.]: [s.n.], 2018.
- [5] SUTHERLAND I E, HODGMAN G W. Reentrant Polygon Clipping[J]. Commun. ACM, 1974, 17(1): 32–42. DOI: 10.1145/360767.360802. ISSN: 0001-0782.
- [6] WEILER K, ATHERTON P. Hidden Surface Removal Using Polygon Area Sorting[J]. SIGGRAPH Comput. Graph., 1977, 11(2). DOI: 10.1145/965141.563896. ISSN: 0097-8930.
- [7] VATTI B R. A Generic Solution to Polygon Clipping[J]. Commun. ACM, 1992, 35(7). DOI: 10.1145/129902.129906. ISSN: 0001-0782.
- [8] GREINER G, HORMANN K. Efficient Clipping of Arbitrary Polygons[J]. ACM Trans. Graph., 1998, 17(2): 71–83. ISSN: 0730-0301.
- [9] CLARK G J, COOPER J N. LEADING COEFFICIENTS AND THE MULTIPLICITY OF KNOWN ROOTS[J]. Mathematics Subject Classification, 2018.
- [10] ZENG Z. A Method Computing Multiple Roots of Inexact Polynomials[C]// Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation. ISSAC '03. Philadelphia, PA, USA: ACM, 2003: 266–272. ISBN: 1-58113-641-2.

- [11] LANG S, MURROW G. Geometry (2. ed.)[M]. [S.l.]: Springer, 1997. ISBN: 978-3-540-96654-8.
- [12] GATHEN J V Z, GERHARD J. Modern Computer Algebra[M]. 3rd. New York, NY, USA: Cambridge University Press, 2013. ISBN: 1107039037, 9781107039032.
- [13] SAUGATA BASU M.-F R Richard Pollack. Algorithms in Real Algebraic Geometry[M]. Vol. 10. [S.l.]: Springer-Verlag Berlin Heidelberg, 2006.
- [14] GEDDES K O, CZAPOR S R, LABAHN G. Algorithms for Computer Algebra[M]. Norwell, MA, USA: Kluwer Academic Publishers, 1992. ISBN: 0-7923-9259-0.
- [15] YAP C K. Fundamental Problems of Algorithmic Algebra[M]. New York, NY, USA: Oxford University Press, Inc., 2000. ISBN: 0-19-512516-9.
- [16] COHEN H. A course in computational algebraic number theory[M]. Vol. 138. [S.l.]: Springer Science & Business Media, 2013.
- [17] COLLINS G E. Subresultants and Reduced Polynomial Remainder Sequences[J/OL]. J. ACM, 1967, 14(1): 128–142. <http://doi.acm.org/10.1145/321371.321381>. DOI: 10.1145/321371.321381. ISSN: 0004-5411.
- [18] BROWN W, TRAUB J. On Euclid's algorithm and the theory of subresultants[J]. J. Assoc. Comput. Mach, 1971, 18: 505–514.
- [19] LOOS R. Generalized Polynomial Remainder Sequences[J]. Comput. Algebr. Symb. Algebraic Comput., Comput. Supp, 1982, 4. DOI: 10.1007/978-3-7091-3406-1_9.
- [20] BROWN W. The subresultant PRS algorithm[J]. ACM Transactions on Mathematical Software (TOMS), 1978, 4(3): 237–249.
- [21] KNUTH D E. The Art of Computer Programming, Volume 1 (3rd Ed.): Fundamental Algorithms[M]. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 1997. ISBN: 0-201-89683-4.
- [22] T. LICKTEIG M.-F R. Cauchy index computation[M]. [S.l.]: Calcolo 33, 1996. ISBN: 337-351.

- [23] LICKTEIG T, ROY M.-F. Sylvester-Habicht Sequences and Fast Cauchy Index Computation[J/OL]. J. Symb. Comput., 2001, 31(3): 315–341. <http://dx.doi.org/10.1006/jSCO.2000.0427>. DOI: 10.1006/jSCO.2000.0427. ISSN: 0747-7171.
- [24] LOMBARDI H, ROY M.-F, SAFEY EL DIN M. New Structure Theorem for Subresultants[J]. J. Symb. Comput., 2000, 29: 663–689. DOI: 10.1006/jSCO.1999.0322.
- [25] EL KAHOUI M. An Elementary Approach to Subresultants Theory[J/OL]. J. Symb. Comput., 2003, 35(3): 281–292. [http://dx.doi.org/10.1016/S0747-7171\(02\)00135-9](http://dx.doi.org/10.1016/S0747-7171(02)00135-9). DOI: 10.1016/S0747-7171(02)00135-9. ISSN: 0747-7171.
- [26] LOOS R. Computing in Algebraic Extensions[J]. 1983. DOI: 10.1007/978-3-7091-7551-4_12.
- [27] COX D A, LITTLE J, O'SHEA D. Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3/e (Undergraduate Texts in Mathematics)[M]. Berlin, Heidelberg: Springer-Verlag, 2007. ISBN: 0387356509.
- [28] GATHEN J, LÜCKING T. Subresultants Revisited[J]. Theoretical Computer Science, 2003, 297: 199–239. DOI: 10.1016/S0304-3975(02)00639-4.
- [29] AKRITAS A G. Elements of computer algebra with applications[M]. Vol. 3. [S.l.]: Wiley New York, 1989.
- [30] ROTE G. Division-Free Algorithms for the Determinant and the Pfaffian: Algebraic and Combinatorial Approaches[M]//. Vol. 2122. [S.l.]: [s.n.], 2001: 119–135. DOI: 10.1007/3-540-45506-X_9.
- [31] ABDELJAOUED J, DIAZ-TOCA G M, GONZALEZ-VEGA L. Bezout Matrices, Subresultant Polynomials and Parameters[J/OL]. Appl. Math. Comput., 2009, 214(2): 588–594. <http://dx.doi.org/10.1016/j.amc.2009.04.032>. DOI: 10.1016/j.amc.2009.04.032. ISSN: 0096-3003.

- [32] DIAZ G M, GONZALEZ V, VEGA L. Various New Expressions for Subresultants and Their Applications[J/OL]. Appl. Algebra Eng., Commun. Comput., 2004, 15(3): 233–266. <https://doi.org/10.1007/s00200-004-0158-4>. DOI: 10.1007/s00200-004-0158-4. ISSN: 0938-1279.
- [33] BIKKER P, UTESHEV A Y. On the Bézout construction of the resultant[J]. Journal of Symbolic Computation, 1999, 28(1-2): 45–88.
- [34] BRÖNNIMANN H, EMIRIS I Z, PAN V Y, et al. Sign determination in residue number systems[J]. Theoretical Computer Science, 1999, 210(1): 173–197.
- [35] AKRITAS A G. There is No “Uspensky’s Method”[C/OL]// Proceedings of the Fifth ACM Symposium on Symbolic and Algebraic Computation. SYMSAC ’86. Waterloo, Ontario, Canada: ACM, 1986: 88–90. ISBN: 0-89791-199-7. <http://doi.acm.org/10.1145/32439.32457>. DOI: 10.1145/32439.32457.
- [36] COLLINS G E, AKRITAS A G. Polynomial Real Root Isolation Using Descarte’s Rule of Signs[C/OL]// Proceedings of the Third ACM Symposium on Symbolic and Algebraic Computation. SYMSAC ’76. Yorktown Heights, New York, US-A: ACM, 1976: 272–275. <http://doi.acm.org/10.1145/800205.806346>. DOI: 10.1145/800205.806346.
- [37] KRANDICK W. Isolierung reeller Nullstellen von Polynomen[M]. Wissenschaftliches Rechnen, Akademie-Verlag, Berlin: J. Herzberger (Ed.), 1995.
- [38] ROUILLIER F, ZIMMERMANN P. Efficient Isolation of Polynomial’s Real Roots[J/OL]. J. Comput. Appl. Math., 2004, 162(1): 33–50. <http://dx.doi.org/10.1016/j.cam.2003.08.015>. DOI: 10.1016/j.cam.2003.08.015. ISSN: 0377-0427.
- [39] MIGNOTTE M. Mathematics for Computer Algebra[M]. Berlin, Heidelberg: Springer-Verlag, 1992. ISBN: 0-387-97675-2.

- [40] KRANDICK W, MEHLHORN K. New Bounds for the Descartes Method[J/OL]. J. Symb. Comput., 2006, 41(1): 49–66. <http://dx.doi.org/10.1016/j.jsc.2005.02.004>. DOI: 10.1016/j.jsc.2005.02.004. ISSN: 0747-7171.
- [41] CAYLEY A. The collected mathematical papers[J]. The Collected Mathematical Papers, 2009, 4: 1–618. DOI: 10.1017/CB09780511703706.
- [42] JOHNSON J R. Algorithms for Polynomial Real Root Isolation[D]. UMI Order No. GAX92-01682. Columbus, OH, USA, 1992.
- [43] M. HEMMER E S L. Kettner. Effects of a modular filter on geometric applications,[M]. MaxB Planck Institut für Informatik, 66123 Saarbrücken, Germany: Technical Report ECG-TR-363111-01, 2004. ISBN: 0-521-64641-3.
- [44] KARAMCHETI V, LI C, PECHTCHANSKI I, et al. A Core Library for Robust Numeric and Geometric Computation[C/OL]// Proceedings of the Fifteenth Annual Symposium on Computational Geometry. SCG '99. Miami Beach, Florida, USA: ACM, 1999: 351–359. ISBN: 1-58113-068-6. <http://doi.acm.org/10.1145/304893.304989>. DOI: 10.1145/304893.304989.
- [45] BURNIKEL C, FLEISCHER R, MEHLHORN K, et al. A strong and easily computable separation bound for arithmetic expressions involving radicals[J]. Algorithmica, 2000, 27(1): 87–99.

二、开题报告

1 问题提出的背景

1.0.1 背景介绍

流体建模在计算流体力学中一直是一个极为重要的话题。近年来科技的发展更是加大了科学界工业界对流体建模的需求。流体的实体建模的一个主要特点是利用数学的语言将流相精确地表达出来,同时严格保持流体的物理性质。流体建模一个很重要的用途就是处理流体的拓扑相变。^[2]提出了一种用直线段来模拟物理区域的边界的方法,并在此基础上建立了严格的代数结构——布尔代数。

其相对应的布尔运算需要求解多边形的交集。这是多边形求交集算法的意义所在。历史上多边形求交集是计算几何中的一个较为基础的运算。最初的交集运算就像线段求交点的最初算法一样,将集合中的多边形进行两两比较,通过 $O(n^2)$ 的运行时间得到结果。

为了解决运行时间慢的问题,出现了很多多边形切割算法。Sutherland 和 Hodgman 提出了 Sutherland-Hodgman 多边形切割算法^[5]。其主要思想是依次将切割多边形的每一条边依延长为直线,判断被切割多边形的顶点分别在直线的“内部”还是“外部”,进而对多边形进行切割。该算法的不足之处在于仅限于凸多边形,对于凹多边形的情况, Sutherland-Hodgman 多边形切割算法可能会留下错误的边。因此, Weiler 和 Atherton 等人提出了 Weiler-Atherton 多边形切割算法^[6]。该算法解决了上一个算法无法求解凹多边形交集的问题。^[7]提出的 Vatti 多边形切割算法进一步解决了带有洞的多边形和自交的多边形的求交集问题。Greiner-Hormann 多边形切割算法^[8]在 Vatti 算法之上,还可以完成取并集等的布尔运算。

1.0.2 研究的目的和意义

本研究的应用目的在于用三次样条来表示流体的物理区域。^[2]中设计了在约当曲线偏序集空间中的 meet 运算和 complementation 运算,并保证了用约当曲线

表示出来的物理区域同样可以构成一个布尔代数。研究的意义在于用三次曲线表示物理区域后，精度会有大幅度提高。

2 项目的主要内容和路线

2.1 研究的主要内容和路线

2.1.1 研究的主要内容

本次毕业设计的主题是将用直线段表示物理区域的算法升级为用三次样条来表示的算法。即我们会用 $f_i \in \mathbb{Q}[x, y]$ 来表示物理边界并进行相应的布尔运算。本研究旨在编写算法和程序来支持物理区域上的布尔运算。目前已经有相关的布尔运算的 C++ 程序，因此主要侧重点在于求解三次样条曲线的交点。

求解三次代数曲线交点的大致思路是沿用求直线交点的扫描算法，但是求解曲线交点与文献综述中介绍的求解直线段交点相比，具有较大的挑战性。因此，本次毕业设计首先应该解决以下几类问题：

1. 三次曲线从几何和代数上讲都有什么样的分类？
2. 与直线不同的是，三次代数曲线与直线或曲线的交点可以不止一个。在这样的情况下，应该如何定义事件点才能准确拆分这些曲线？
3. 我们应该如何从计算机算法的角度用严格的数学语言去描述曲线上的事件点？
4. 对于直线段集合，可以通过观察直线段是否互换位置来判断两直线是否相交，对于曲线应该如何判断相交？
5. 在求解二元三次方程组时，如何进行消元同时又保证解集一定是零解集？
6. 以有理数为系数的多项式求解运算不是代数封闭的，那么求解过程中出现的无理数应该如何精确又高效地表达？
7. 对上述问题得出的算法是否适用于所有曲线，如果不是，可以通过怎么样的坐标变化使得算法更具有普遍性？

[3] 对上述问题做出了详细的讨论。其中 3.3 针对无平方因式的多项式 $f \in \mathbb{Q}[x, y]$ ，将其分为不可约多项式，直线和不可约二次曲线的乘积，以及三条直线的乘积三种情况，目的是讨论奇点数量，几何外观，以及其是否为有理数。这样可以为曲线的几何分析打下基础。

4.1 节从几何直观的角度定义了曲线几何中的事件点：想象一条垂直 x 轴的直线 $x = \vartheta$ 从左到右扫过目标曲线集，由于三次代数曲线与扫描线的交点可能不止一个，因此可能存在一些点集 S ，使得 $\vartheta = s_i \in S$ 时，扫描直线与曲线的交点数量发生了变化。这种点被定义为事件点。

对于第三个问题，可以证明几何直观上的事件点用数学语言可以写为 $v \in f \cap f_y$ 。具体的求解可以先通过 f, f_y 在 y 点的结式矩阵行列式 $R_f := \text{res}(f, f_y, y)$ 消去一个变量，再进行无平方分解 $R_f = \prod_{m=1}^M R_{fm}^m$ ([3] 的 2.2.3 提出可以通过 Yun 算法实现)。最后用隔离区间的方法求解每一个无平方因式的根，即得我们想要的事件点。

问题四中曲线是否相交可以通过 $R_{fg} := \text{res}(f, g, y)$ 在目标根的重数 m_i^{fg} 来判断，如果 $m_i^{fg} > 0$ ，说明 f 和 g 在该点有交点。具体的求根可以使用隔离算法：首先将 f 进行无平方分解以确保不会出现多重根的现象，然后利用二分法将每一个根隔离开来，使得每一个区间里要么没有根，要么只有一个根，判断根是否存在于目标区间的依据是 Descartes 符号法则。

对于问题六，可以引入分离界限数类型和符号表示来处理带根号的表达式，也可以两者结合使用。具体的使用会在后期写代码时探索。

文献中 2.3.1 节指出了当 f 是 y 正规的时，问题五的设想可以被实现。对于不具有 y 正规性质的 f 来说，文献中的 4.4 还提出了一种可能的解决办法。对于一些不满足算法的输入条件的多项式，可以通过转换坐标系来调整多项式，使得新的多项式满足条件。然而这样可能同时破坏掉之前满足的其他条件，因此存在一定风险。

2.1.2 技术路线

首先输入一个三次样条曲线集合 $S := \{f | f \in \mathbb{Q}[x, y], \deg(f) \leq 3\}$ ，在预处理时将其依据单曲线事件点 (one-curve event point) 分为多段弧 (arc)。在将曲线拆分为多段弧的过程中，算法同时也会检验曲线是否符合其输入条件，如是

否对 y 正规，是否有切线垂直 x 轴的弯曲点。如果存在不满足条件的曲线，^[3] 的 4.4 节讨论了重新选取坐标系的可能，但是这样是存在一定风险的，具体细节可以等到后期进行检验。将曲线拆分完毕后，用垂直于 x 轴的扫描线从左向右扫过平面。与扫描线在当下相交的所有曲线都按照纵坐标轴的顺序存储在 Y 结构中，用 X 结构来存储所有已知的事件点（字母顺序）。然后应用类似直线的扫描算法：

1. 从 X 结构中取出下一个事件点，找到其在 Y 结构中相关联的曲线段。

这一步的操作需要比较相关点和线段的 y 坐标，可以通过比较不同曲线在横坐标等于相关点横坐标时，纵坐标的大小来决定。

2. 删掉结束扫描的曲线段。

这一步需要判断正在处理的点是否是曲线段的端点（target point），比较坐标关系即可。

3. 依据交点的重数改变尚未扫描的曲线在 Y 结构中的顺序。

之前计算结式矩阵的步骤保证了两条曲线是互质的。因此若两条曲线段重合，当且仅当它们所在的曲线相同，以及它们对应的 X 轴区间也相同。

4. 将即将开始扫描的新曲线按照顺序加入 Y 结构。

这里需要比较两条曲线段之间的位置关系，如果两条曲线段属于同一条曲线，那么比较两曲线的位置关系等同于比较其曲线段（或弧）的序号（在几何分析中就已经得出）；如果两条曲线段属于两条不同的曲线，那么可以比较交点右侧一个足够小的区间的纵坐标大小。

5. 将新发现的事件点加入 X 结构。

完成这一步需要比较事件点的坐标关系。先比较横坐标的大小，如果大小一样再比较纵坐标的大小。

除了上述七个问题，曲线的扫描线算法和直线相比一个很重要的区别就在于，在检测出曲线的一个交点后， Y 结构中的曲线顺序未必会像直线一样发生交换（因为可能有相切的情况）。这时需要通过交点的重数来进行判断。^[3] 的 5.1 节重点讨论了这个问题。

2.1.3 可行性分析

由于有充足的理论做支撑,研究具有较高的可行性。从曲线的拆分,与扫描线求交,与曲线求交,以及完整的扫描线算法,上文七个问题都对其确定了解决方案。除此之外,^[3]还在第四章进行了大量的几何分析,以通过根的重数来判断曲线交点的类型。

技术路线中提到的很多问题都已经在计算几何算法库 (CGAL) 中得到了代码实现。CGAL 是一个提供高效准确的计算几何算法的 C++ 算法库,在工业界有着很广泛的应用。本次研究会借助计算几何算法库中的取模运算 (Modular Arithmetic) 程序包来完成基本的最大公因式运算,结式矩阵计算,以及过滤器,等等。多项式程序包 (Polynomial) 通过 C++ 引入了 d 元多项式 *Polynomial_d* 的概念。*Polynomial_d* 定义下的多项式集合是一个代数结构,这里多项式环的运算 (加减乘) 也被定义进程序包。同时, *Polynomial_d* 的定义还应该具备特征类 *Polynomial_traits_d* $< \text{Polynomial}_d >$ 。这样一个特征类里提供了进一步的多项式运算。

对于一个 R 环上的 d 元多项式,有两种可以用来分析的视角:

首先是从单变量多项式的角度来看,在这样的角度下,多项式可以被看作 $R[x_0, \dots, x_{d-2}][x_{d-1}]$ 中的元素,这样的多项式是环 $R[x_0, \dots, x_{d-2}]$ 上的单变量多项式。还有一种是多变量角度,即将所有的多项式看成 $R[x_0, \dots, x_{d-1}]$ 上的元素。

特征类也依据两种角度提供了两种系数类型, *Polynomial_traits_d* $:: \text{Coefficient_type}$ 代表属于 $R[x_0, \dots, x_{d-2}]$ 的系数。 *Polynomial_traits_d* $:: \text{Innermost_coefficient_type}$ 代表 R 上的系数类型。

本次研究可以通过 CGAL 算法库实现多项式上的无平方分解、二元多项式的消元等基本运算。除此之外,本次研究还需要计算已知根的重数以便对曲线进行相应的几何分析。目前关于已知根的重数计算存在着很多算法,如 GREGORY J. CLARK 和 JOSHUA N. COOPER^[9] 证明了首一多项式的根的重数可以由系数唯一确定,进而可以用算法将每个已知根的重数计算出来。ZHONGGANG ZENG 也在^[10]中提出了两种计算根的重数的算法。这些文献都有可能为本次研究提供参考。

3 研究计划进度安排及预期目标

3.0.1 进度安排

表 3.1: 未来计划表

时间	工作安排
2019 年 3 月 25 日——2019 年 3 月 31 日	修改开题报告
2019 年 4 月 1 日——2019 年 4 月 10 日	编写曲线求交点等基础子程序
2019 年 4 月 11 日——2019 年 4 月 20 日	编写扫描线算法
2019 年 4 月 20 日——2019 年 5 月 3 日	调试程序
2019 年 5 月 4 日——2019 年 5 月 13 日	撰写毕业论文并修改

3.0.2 预期成果

预计在毕业设计结题答辩之前完成 Matlab 或 C++ 代码。实现输入任意三次曲线的集合都可以输出所有的交点的算法。并完成学位论文。

4 参考文献

- [1] GIVANT S, HALMOS P. Introduction to Boolean Algebras[M]. [S.l.]: [s.n.], 2009. DOI: 10.1007/978-0-387-68436-9.
- [2] ZHANG Q, LI Z. Fluid Modeling and Boolean Algebra for Arbitrarily Complex Topology in Two Dimensions[Z]. under review, <https://arxiv.org/abs/1902.06378>.
- [3] EIGENWILLIG A, KETTNER L, SCHÖMER E, et al. Exact, efficient, and complete arrangement computation for cubic curves[J]. Comput. Geom., 2006, 35: 36–73.
- [4] LI Z. A Variant of the Line Sweep Algorithm with User-defined Uncertainty Quantification[C]//. [S.l.]: [s.n.], 2018.
- [5] SUTHERLAND I E, HODGMAN G W. Reentrant Polygon Clipping[J]. Commun. ACM, 1974, 17(1): 32–42. DOI: 10.1145/360767.360802. ISSN: 0001-0782.
- [6] WEILER K, ATHERTON P. Hidden Surface Removal Using Polygon Area Sorting[J]. SIGGRAPH Comput. Graph., 1977, 11(2). DOI: 10.1145/965141.563896. ISSN: 0097-8930.
- [7] VATTI B R. A Generic Solution to Polygon Clipping[J]. Commun. ACM, 1992, 35(7). DOI: 10.1145/129902.129906. ISSN: 0001-0782.
- [8] GREINER G, HORMANN K. Efficient Clipping of Arbitrary Polygons[J]. ACM Trans. Graph., 1998, 17(2): 71–83. ISSN: 0730-0301.
- [9] CLARK G J, COOPER J N. LEADING COEFFICIENTS AND THE MULTIPLICITY OF KNOWN ROOTS[J]. Mathematics Subject Classification, 2018.
- [10] ZENG Z. A Method Computing Multiple Roots of Inexact Polynomials[C]// Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation. ISSAC '03. Philadelphia, PA, USA: ACM, 2003: 266–272. ISBN: 1-58113-641-2.

- [11] LANG S, MURROW G. Geometry (2. ed.)[M]. [S.l.]: Springer, 1997. ISBN: 978-3-540-96654-8.
- [12] GATHEN J V Z, GERHARD J. Modern Computer Algebra[M]. 3rd. New York, NY, USA: Cambridge University Press, 2013. ISBN: 1107039037, 9781107039032.
- [13] SAUGATA BASU M.-F R Richard Pollack. Algorithms in Real Algebraic Geometry[M]. Vol. 10. [S.l.]: Springer-Verlag Berlin Heidelberg, 2006.
- [14] GEDDES K O, CZAPOR S R, LABAHN G. Algorithms for Computer Algebra[M]. Norwell, MA, USA: Kluwer Academic Publishers, 1992. ISBN: 0-7923-9259-0.
- [15] YAP C K. Fundamental Problems of Algorithmic Algebra[M]. New York, NY, USA: Oxford University Press, Inc., 2000. ISBN: 0-19-512516-9.
- [16] COHEN H. A course in computational algebraic number theory[M]. Vol. 138. [S.l.]: Springer Science & Business Media, 2013.
- [17] COLLINS G E. Subresultants and Reduced Polynomial Remainder Sequences[J/OL]. J. ACM, 1967, 14(1): 128–142. <http://doi.acm.org/10.1145/321371.321381>. DOI: 10.1145/321371.321381. ISSN: 0004-5411.
- [18] BROWN W, TRAUB J. On Euclid's algorithm and the theory of subresultants[J]. J. Assoc. Comput. Mach, 1971, 18: 505–514.
- [19] LOOS R. Generalized Polynomial Remainder Sequences[J]. Comput. Algebr. Symb. Algebraic Comput., Comput. Supp, 1982, 4. DOI: 10.1007/978-3-7091-3406-1_9.
- [20] BROWN W. The subresultant PRS algorithm[J]. ACM Transactions on Mathematical Software (TOMS), 1978, 4(3): 237–249.
- [21] KNUTH D E. The Art of Computer Programming, Volume 1 (3rd Ed.): Fundamental Algorithms[M]. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 1997. ISBN: 0-201-89683-4.
- [22] T. LICKTEIG M.-F R. Cauchy index computation[M]. [S.l.]: Calcolo 33, 1996. ISBN: 337-351.

- [23] LICKTEIG T, ROY M.-F. Sylvester-Habicht Sequences and Fast Cauchy Index Computation[J/OL]. J. Symb. Comput., 2001, 31(3): 315–341. <http://dx.doi.org/10.1006/jscs.2000.0427>. DOI: 10.1006/jscs.2000.0427. ISSN: 0747-7171.
- [24] LOMBARDI H, ROY M.-F, SAFEY EL DIN M. New Structure Theorem for Subresultants[J]. J. Symb. Comput., 2000, 29: 663–689. DOI: 10.1006/jscs.1999.0322.
- [25] EL KAHOUI M. An Elementary Approach to Subresultants Theory[J/OL]. J. Symb. Comput., 2003, 35(3): 281–292. [http://dx.doi.org/10.1016/S0747-7171\(02\)00135-9](http://dx.doi.org/10.1016/S0747-7171(02)00135-9). DOI: 10.1016/S0747-7171(02)00135-9. ISSN: 0747-7171.
- [26] LOOS R. Computing in Algebraic Extensions[J]. 1983. DOI: 10.1007/978-3-7091-7551-4_12.
- [27] COX D A, LITTLE J, O'SHEA D. Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3/e (Undergraduate Texts in Mathematics)[M]. Berlin, Heidelberg: Springer-Verlag, 2007. ISBN: 0387356509.
- [28] GATHEN J, LÜCKING T. Subresultants Revisited[J]. Theoretical Computer Science, 2003, 297: 199–239. DOI: 10.1016/S0304-3975(02)00639-4.
- [29] AKRITAS A G. Elements of computer algebra with applications[M]. Vol. 3. [S.l.]: Wiley New York, 1989.
- [30] ROTE G. Division-Free Algorithms for the Determinant and the Pfaffian: Algebraic and Combinatorial Approaches[M]//. Vol. 2122. [S.l.]: [s.n.], 2001: 119–135. DOI: 10.1007/3-540-45506-X_9.
- [31] ABDELJAOUED J, DIAZ-TOCA G M, GONZALEZ-VEGA L. Bezout Matrices, Subresultant Polynomials and Parameters[J/OL]. Appl. Math. Comput., 2009, 214(2): 588–594. <http://dx.doi.org/10.1016/j.amc.2009.04.032>. DOI: 10.1016/j.amc.2009.04.032. ISSN: 0096-3003.

- [32] DIAZ G M, GONZALEZ V, VEGA L. Various New Expressions for Subresultants and Their Applications[J/OL]. Appl. Algebra Eng., Commun. Comput., 2004, 15(3): 233–266. <https://doi.org/10.1007/s00200-004-0158-4>. DOI: 10.1007/s00200-004-0158-4. ISSN: 0938-1279.
- [33] BIKKER P, UTESHEV A Y. On the Bézout construction of the resultant[J]. Journal of Symbolic Computation, 1999, 28(1-2): 45–88.
- [34] BRÖNNIMANN H, EMIRIS I Z, PAN V Y, et al. Sign determination in residue number systems[J]. Theoretical Computer Science, 1999, 210(1): 173–197.
- [35] AKRITAS A G. There is No “Uspensky’s Method”[C/OL]// Proceedings of the Fifth ACM Symposium on Symbolic and Algebraic Computation. SYMSAC ’86. Waterloo, Ontario, Canada: ACM, 1986: 88–90. ISBN: 0-89791-199-7. <http://doi.acm.org/10.1145/32439.32457>. DOI: 10.1145/32439.32457.
- [36] COLLINS G E, AKRITAS A G. Polynomial Real Root Isolation Using Descarte’s Rule of Signs[C/OL]// Proceedings of the Third ACM Symposium on Symbolic and Algebraic Computation. SYMSAC ’76. Yorktown Heights, New York, US-A: ACM, 1976: 272–275. <http://doi.acm.org/10.1145/800205.806346>. DOI: 10.1145/800205.806346.
- [37] KRANDICK W. Isolierung reeller Nullstellen von Polynomen[M]. Wissenschaftliches Rechnen, Akademie-Verlag, Berlin: J. Herzberger (Ed.), 1995.
- [38] ROUILLIER F, ZIMMERMANN P. Efficient Isolation of Polynomial’s Real Roots[J/OL]. J. Comput. Appl. Math., 2004, 162(1): 33–50. <http://dx.doi.org/10.1016/j.cam.2003.08.015>. DOI: 10.1016/j.cam.2003.08.015. ISSN: 0377-0427.
- [39] MIGNOTTE M. Mathematics for Computer Algebra[M]. Berlin, Heidelberg: Springer-Verlag, 1992. ISBN: 0-387-97675-2.

- [40] KRANDICK W, MEHLHORN K. New Bounds for the Descartes Method[J/OL]. J. Symb. Comput., 2006, 41(1): 49–66. <http://dx.doi.org/10.1016/j.jsc.2005.02.004>. DOI: 10.1016/j.jsc.2005.02.004. ISSN: 0747-7171.
- [41] CAYLEY A. The collected mathematical papers[J]. The Collected Mathematical Papers, 2009, 4: 1–618. DOI: 10.1017/CB09780511703706.
- [42] JOHNSON J R. Algorithms for Polynomial Real Root Isolation[D]. UMI Order No. GAX92-01682. Columbus, OH, USA, 1992.
- [43] M. HEMMER E S L. Kettner. Effects of a modular filter on geometric applications,[M]. MaxB Planck Institut für Informatik, 66123 Saarbrücken, Germany: Technical Report ECG-TR-363111-01, 2004. ISBN: 0-521-64641-3.
- [44] KARAMCHETI V, LI C, PECHTCHANSKI I, et al. A Core Library for Robust Numeric and Geometric Computation[C/OL]// Proceedings of the Fifteenth Annual Symposium on Computational Geometry. SCG '99. Miami Beach, Florida, USA: ACM, 1999: 351–359. ISBN: 1-58113-068-6. <http://doi.acm.org/10.1145/304893.304989>. DOI: 10.1145/304893.304989.
- [45] BURNIKEL C, FLEISCHER R, MEHLHORN K, et al. A strong and easily computable separation bound for arithmetic expressions involving radicals[J]. Algorithmica, 2000, 27(1): 87–99.

三、外文翻译

1 外文翻译——准确、有效、完备的曲线计算

1.2 代数基础

¹代数曲线的算法处理很大程度上取决于代数运算。我们将使用过的算法简洁精炼地总结了下来，使之构成一个完整的逻辑体系，并且让非符号计算专业的人也可以轻易看懂。本章还在算法的选择上提供了必要的背景知识。

本章节中与抽象代数有关的知识是以 Lang 写的教科书^[11] 为参考的。与算法有关的详细知识可参见 Geddes, Czapor 以及 Labahn, Akritas, Cohen 以及 von zur Gathen 和 Gerhard 编写的《计算机代数》(Computer Algebra)^[12]。最近由 Basu, Pollack 和 Roy^[13] 共同编写出版的书将实代数几何和计算机代数结合了起来。

1.2.1 环和域

回忆一下和环、域相关的代数知识。本章节中所有的域都默认拥有特征零，即其包含有理数域 \mathbb{Q} 作为子域。在现代术语学中，环指带单位元的可交换环。我们额外要求它包含整数 \mathbb{Z} ，并且除非特别说明，这样的环没有零因子。在我们使用的术语中，每一个环 \mathcal{R} 都有一个唯一的商域 $Q(\mathcal{R})$ ，即，一个在“属于” \subset 意义下以 \mathcal{R} 作为子环的最小域。比如， $Q(\mathbb{Z}) = \mathbb{Q}$ 。

如果存在 $c \in \mathcal{R}$ 使得 $r = cd$ ，我们称 d 除尽 r ，写作 $d|r$ 。如果 $u|1$ ，那么 u 称为 \mathcal{R} 的单位。一个 $r \neq 0$ 且不能被写作另外两个非单位元素乘积的非单位元素被称为不可约元。如果环 \mathcal{R} 中任何一个非零非单位的元素都具有如下形式的分解

$$r = \prod_{i=1}^k p_i \quad (1.2.1)$$

¹本文翻译自 Arno Eigenwillig, Lutz Kettner, Elmar Schömer, Nicola Wolpert 撰写的“Exact, efficient, and complete arrangement computation for cubic curves”第二章, Computational Geometry, vol. 35, pp. 36–73, 08 2006

那么 R 被称为唯一分解环 (UFD)。其中 p_i 是不可约元素, p_i 的选择在不考虑次序时是唯一的。在任意一个唯一分解环 R 中, 不可约元素 p 都是质数, 即对于所有 $r, s \in R$ 可以得到 $p|rs \Rightarrow p|r \vee p|s$ (证明见^[1]II.4)。给定 $r \in R$ 和某个不可约元素 p , 使得 $p^e|r$ 的最大 $e \in \mathcal{N}_0$ 被称为 p 在 r 中的重数。

1.2.2 多项式

多项式基础 将一个不定元加入到环 R 中, 就得到了单变量多项式环 $R[x]$ 。我们称首位系数 $l(f)$ 是 1 的多项式 $f \in R[x]$ 为首一多项式。递归性地添加不定元可以得到多变量多项式 $R[x_1] \dots [x_n] = R[x_1, \dots, x_n]$ 。 $R[x_1, \dots, x_n]$ 的单位就是 R 的单位。

多变量多项式可以看成是一个以 x_n 为变量, $R[x_1, \dots, x_{n-1}]$ 为系数的多项式。其次数 $\deg(f)$ 定义为 $\max\{i_1 + \dots + i_n | a_{i_1 \dots i_n} \neq 0\}$ 。

若多项式中的每一项的次数都相同, 这样的多项式成为齐次多项式。否则, 非齐次多项式可以唯一地分解为几个齐次项相加 $f = f_0 + f_1 + \dots + f_{\deg(f)}$, 这里 f_d 是 d 次齐次项。第 d 次依照 $d = 0, 1, 2, 3, \dots$ 的顺序被称为常数项、线性项、二次项等等。

令 $k \leq n, \epsilon \in R^k$ 。用 $\epsilon_1, \dots, \epsilon_k$ 代替 $x_1 \dots x_k$, 我们得到了一个映射: $f \mapsto f|_\epsilon = f(\epsilon_1, \dots, \epsilon_k, x_{k+1}, \dots, x_n) \in R[x_{k+1}, \dots, x_n]$ 。

如果多项式 $f \in R[x_1, \dots, x_n]$ 含有 $cx_n^{\deg(f)}, 0 \neq c \in R$ 的项, 那么其被称为 x_n 正规。在这样的情况下, 当 x_1, \dots, x_k 被替代后, 多项式的次数保持不变。两个 x_n 正规的多项式的乘积也是 x_n 正规的。 x_n 正规的多项式的任何因式也是 x_n 正规的。

唯一分解环和 GCD 令 K 为一个域。单变量多项式环 $K[x]$ 可以通过整除和余数来证明其为一个欧几里得环。对于任意两个多项式 $f, g \in K[x], g \neq 0, K[x]$ 中存在一个唯一的除数 q 和余数 r , 使得

$$f = qg + r, \deg(r) < \deg(g) \quad (1.2.2)$$

此定理的证明过程本身就提供了一个简单有效的算法。当且仅当 $g|f$ 时, 余数为 0, 除数为 $\frac{f}{g}$ 。

回忆在环论中，两个元素 $r, s \in R$ 的最大公因数 $d \in R$ ，使得 $d|r \wedge d|s$ ，对任意 d' ， $d'|r \wedge d'|s \Rightarrow d'|d$ 。欧几里得算法用来计算 $K[x]$ 中的最大公因数：给定两个多项式 $f, g \in K[x]$ ，将次数较大的多项式的余数与另一个多项式的模替换，重复此步骤，直到余数为 0。最后一个非 0 余数就是 f 和 g 的最大公因数。此外，升级版的欧几里得算法 (EEA) 还额外计算了满足 $d = pf + qg$ 的 Bezout 因子。详情可见^[14]2.4,^[12]3.2,^[15]2.2,^[16]3.2.1。

EEA 证明了 $K[x]$ 是一个主理想整环。因此是一个 UFD^[11]II.4, V.4。因此任意一个非零多项式 $f \in K[x]$ 都有唯一的分解：

$$f = l(f) \prod_{j=1}^k q_j^{e_j} \quad (1.2.3)$$

该分解将多项式 f 分解成了系数和几个不同、不可约、带有幂次 $e_j > 0$ 首一多项式的乘积。这个结论是由 (1) 得出来的。

我们现在考虑唯一分解环 R 上多项式的可除性质。在这种情况下，非单位常量是相关的。对于非零多项式 $f \in R[x]$ ，定义其容度 $\text{cont}(f)$ 为其系数的最大公因数。如果 $\text{cont}(f) = 1$ ，那么称 f 为本原多项式。

命题 1: (高斯引理)。令 R 为一个唯一分解环。令 $f, g \in R[x]$ 为非零元。那么

$$\text{cont}(fg) = \text{cont}(f)\text{cont}(g) \quad (1.2.4)$$

证明详见^[11]V.6。接下来的命题将 $R[x]$ 和 $Q(R)[x]$ 的能否被除尽的问题联系起来。

推论 2: 令 R 为一个唯一分解环，令 $f, g \in R[x]$ 为非零多项式，那么接下来的条件相互等价。

1. 在 $R[x]$ 中 $g|f$
2. $Q(R)[x]$ 中 $g|f$ ， R 中 $\text{cont}(g)|\text{cont}(f)$

特别地，如果 g 是本原多项式，那么以上两个条件等价

接着^[11]V.6 的思路，我们可以得到：

定理 3: (高斯定理) 如果 R 是 UFD，那么 $R[x_1, \dots, x_n]$ 是 UFD。如果 K 是一个域，那么 $K[x_1, \dots, x_n]$ 也是一个 UFD。

因此，任意非零元素 $f \in K[x_1, \dots, x_n]$ 都可以分解成唯一的互质的不可约的带有幂次的因式。(我们会在 $n = 2$ 曲线下详细了解这类不可约因子) 由此得到对于 $f, g \in K[x_1, \dots, x_n]$ ，存在 f, g 的最大公因数。下一个推论对于命题 1 的计算起到了很大的作用。

推论 4: 令 R 为 UFD，令 $f, g \in R[x]$ 为一个非零多项式，那么：

$$\text{cont}(\gcd(f, g)) = \gcd(\text{cont}(f), \text{cont}(g)) \quad (1.2.5)$$

假设我们可以在 UFD R 中计算最大公因数。那么我们就可以计算出非零多项式 $f, g \in R[x]$ 的最大公因数 $\gcd(f, g)$ ：因为依据推论 2， $\gcd(f, g)$ 可以被看做 $Q(R)[x]$ 的元素，在这样的意义下， $\gcd(f, g)$ 是 f, g 的最大公因数。我们可以用欧几里得算法将其计算出来。通过利用推论 4，常量因式中的误差可以通过将容度调整为 $\gcd(\text{cont}(f), \text{cont}(g))$ 。这样可以帮助我们计算最大公因数，比如 $\mathbb{Z}[x], R[x_1, \dots, x_n]$ 或 $K[x_1, \dots, x_n]$ 。参见^[14]。

而有效的算法只是在概念上应用 $Q(R)$ 。实际算法存储了 R 中的所有系数，从而避免分数运算带来的误差损失。这时为了计算 $R[x]$ 中的余数，我们需要把欧几里得算法中的除法修改为形式上的除法。具体操作为将 (2) 中的 f 替换为 $l(g)^{\deg(f) - \deg(g) + 1} f$ 。这样 R 中的任何除法操作都具有可行性了。为了防止系数增长，我们需要预测出每一个余数的常量因子，并将它除去。我们这里利用 Collins^[17]，Brown 和 Traub^[18] 的子结式方法。此方法在 Loos^[19]4.5，Brown 的^[20] 以及 Knuth 的^[21] 有进一步的介绍。(此方法在 Lickteig 和 Roy 的^[22],^[23] Theorem 4.1，以及^[24],^[13]8.3,^[25] 中有进一步的改进)

重数和导数 只要我们理解了将多项式分解成唯一的不可约因式这一概念，我们就可以导出一个较弱的概念：无平方因式分解，或依重数分解。以 (3) 为例子，将 (3) 中具有相同重数的因子 q_i 分成一组组成因式 s_m ：

$$f = l(f) \prod_{m=1}^{\max_i e_i} s_m^{e_i}, s_m = \prod_{e_i=m} p_i \quad (1.2.6)$$

这里的因式 s_m 是无平方因式，也就是说它们的所有因子重数均为 1。并且任意两个 s_m 都是互质的，即，两者之间不含有共同的非单位因式。我们将 f 中的无平方成分定义为 $\prod_m s_m$ 。

无平方分解的意义在于便于通过求导而进行计算。对于一个多项式 $f = \sum_{i=0}^n a_i x^i$ 我们将其导数定义为 $f' := \sum_{i=1}^n i a_i x^{i-1}$ 。对于多变量多项式 $f \in R[x_1, \dots, x_n]$ ，

偏导数的定义方法也是同理的，写作 $f_{x_1 x_2}$ 。关于不同变量的偏导数运算是可以交换的。对 f 进行 $r, r \geq 0$ 次求导可以的结果称为 f 的 r 次导数。 f 的一阶偏导的列向量是梯度 $\nabla f := (f_{x_1}, \dots, f_{x_m})^T$

导数和重数的关系见下列命题：

命题 5 令 K 为一个域。令 $f \in K[x]$ 为一个非零多项式，令 $f = l(f) \prod_{m=1}^M s_m^m$ 为其无平方分解。那么对于任意的 $k \geq 0$ ：

$$\gcd(f, f', \dots, f^{(k)}) = \prod_{m=k+1}^M s_m^{m-k} \quad (1.2.7)$$

特别地， f 中的非平方元素是 $\frac{f}{\gcd(f, f')}$

证明：通过对 k 进行归纳。 $k = 0$ 的情况下结论显然成立。归纳部分的证明可以通过一下来实现：令 $f = g^r h, r \geq 1$ 这里 g 是不可约的，且无法除尽 h 。那么 $f' = r g' g^{r-1} h + g^r h' = (r g' h + g h') g^{r-1}$ 。这里 g 无法除尽 $r g' h + g h'$ ，因为 g 可以除尽 $g h'$ 但是无法除尽 g' (次数较小) 或 h (根据定义)。

无平方分解，就像最大公因数计算一样，不取决于 K 的选择，且对所有包含 f 系数的域都适用。此外，该命题对 x_n 正规多项式 $f \in K[x_1, \dots, x_n]$ 以及其对 x_n 求导的情况都适用。

(5) 中重数和导数的关系帮助我们通过重复的求导和最大公因数计算来求解 $f \in K[x]$ 或者 $f \in R[x]$ (R 为 UFD) 的无平方分解。Yun 算法^[14] 的 8.2 采用了一种巧妙的算法来进行计算，并将中间多项式的系数控制到最小。

这个算法提出了一个不涉及最大公因数的算法，即，在 $d = \gcd(f, g)$ 的时候，将一对多项式 (f, g) 换成三个多项式 $(f/d, g/d, d)$ 。我们采用最直观的方法，计算 d 并用它除尽其他多项式。另一种执行欧几里得算法并计算不涉及最大公因数部分的算法相比之下，并不能更有效地解决我们研究中遇到的单一变量问题。

多项式的根 令 K 为一个域，令 $f \in K[x]$ 。我们可以知道在某一点 $\epsilon \in K$ 处 $f(\epsilon) = 0$ 当且仅当 $x - \epsilon$ 是 f 的一个不可约因式，这是带余除法的一个结论。这样的结论有利于我们将 ϵ 作为多项式的根或 0 的重数定义为 $x - \epsilon$ 作为 f 因式的重数。另外，这样的定义在某种程度上帮助我们理解，一个 n 次的多项式 f 至多有 n 个根（重数算在内），以及对于 f 的一个不可约因式 p ，其要么是线性的（即和 f 的一个根对应），要么次数大于 1，在 K 中无解。

如果上文中的第二种情况不出现，那么域 K 被称为代数封闭的。对于每一个

域 K 都存在唯一一个在 \mathbb{C} 意义下包含 K 的最小代数封闭域 \bar{K} ，称为 \mathbb{C} 意义下 K 的闭包。 \bar{K} 中的每一个元素都是以 K 元素为系数的多项式的根。详情见^[11]VII.2。实数域的代数封闭域是复数域。有理数 \mathbb{Q} 的代数闭包 $\bar{\mathbb{Q}}$ 是一个真子集。

令 ϑ 为一个代数数。 $\mathbb{Q}(\vartheta)$ 代表 $\bar{\mathbb{Q}}$ 包含 ϑ 的最小 \mathbb{C} 子域。在 $\mathbb{Q}[x]$ 所有以 ϑ 为根的首一多项式中，存在一个次数最小的多项式。这个多项式是唯一存在的，被称为 f 的极小多项式。 ϑ 的次数被定义为 f 的次数 $\deg(f)$ 。求值同态 $\mathbb{Q}[x] \rightarrow \mathbb{Q}(\vartheta), x \mapsto \vartheta$ 的核包括 f 的倍数因式。因此 $\mathbb{Q}(\vartheta)$ 域与商环 $\mathbb{Q}[x]/(f)$ 属于同构关系。详情见^[11]II.1, VII.1。现考虑 f 的所有根 $\vartheta_1 := \vartheta, \vartheta_2, \dots, \vartheta_d$ 。它们在代数的意义下是等价的，因为对于任何 $1 \leq i, j \leq d$ ， $\mathbb{Q}(\vartheta_i)$ 和 $\mathbb{Q}(\vartheta_j)$ 之间可以建立起一个同构。因此一个系数为实数的多项式的方程的解要么包含所有 $\vartheta_1 := \vartheta, \vartheta_2, \dots, \vartheta_d$ 要么哪个都不包含。我们称 ϑ_i 为 ϑ 的代数共轭。我们可以观察，将 i 用 $-i$ 来代替时，得到了极小多项式 $x^2 + 1$ 的另外一个根，也就是其共轭复数。

代数共轭的概念可以延伸到代数数组成的 n 元组 ϵ ，具体操作是选择一个代数数 ϑ ，使得所有的 ϵ_i 都是 ϑ 中的有理式。而 ϑ 相应地也是 $\epsilon_1, \dots, \epsilon_n$ 的有理式。（这样的 ϑ 是存在的，被称为 $\mathbb{Q}(\epsilon_1) \dots (\epsilon_n)$ 的素元，详见^[11]VII.6 和^[26]Theorem 11 的证明）。 ϑ 的两个不一样的共轭对应不同的 ϵ 。如果 ϵ 是一个以有理数为系数的多项式方程组的解，那么其任何共轭也都是多项式方程组的解。

1.2.3 结式矩阵

将两个代数曲线 f 和 g 相交即意味着解方程组 $f(x, y) = g(x, y) = 0$ 。为了达到这样一个目的，我们需要一些与消元相关的定理。

Sylvester 结式矩阵 我们首先来看单变量的两个方程式的可解性。

命题 6. 令 K 为一个域。令 $f = \sum_{i=0}^m a_i x^i, a_m \neq 0$ 和 $g = \sum_{i=0}^n b_i x^i, b_n \neq 0$ 为 $K[x]$ 中的两个多项式，那么下列两个条件等价：

1. f 和 g 在代数闭包 \bar{K} 中有共同的根
2. $\deg(\gcd(f, g)) > 0$
3. 存在非零 $u, v \in K[x]$ ，其中 $\deg(u) < \deg(g)$ ， $\deg(v) < \deg(f)$ ，使得 $uf + vg = 0$

4. Sylvester 矩阵的行列式为 0 (写作 $\text{Syl}(f, g)$):

$$\begin{bmatrix} a_m & \dots & \dots & a_0 \\ & & & & & \\ & & a_m & \dots & \dots & a_0 \\ b_n & \dots & \dots & b_0 \\ & & & & & \\ & & b_n & \dots & \dots & b_0 \end{bmatrix}$$

证明: (i)-(iii) 的等价性证明比较显而易见。(iii) 和 (iv) 的等价性可以用线性代数来证明。注意 $\text{Syl}(f, g)^T$ 是一个将代表 u, v 的 $K^n \times K^m$ 映射到代表 $uf + vg$ 的 K^{n+m} 的线性映射。如果有一个非零向量被映射为 0, 那么行列式为 0。详见^[27]3.5。

行列式 $\det(\text{Syl}(f, g))$ 被称为 f, g 关于 x 的结式 $\text{res}(f, g, x)$ 。下列命题在 (i) \Leftrightarrow (iv) 下不难证明。

命题 7. 令 K 为一个域。考虑 $K[x]$ 中的两个非零多项式 $f = a_m \sum_{i=1}^m (x - \alpha_i)$ 和 $g = b_n \prod_{j=1}^n (x - \beta_j)$, 那么下列等式成立:

$$\text{res}(f, g, x) = \alpha_m^n b_n^m \prod_{i=1}^m \prod_{j=1}^n (\alpha_i - \beta_j) \quad (1.2.8)$$

定理的证明可以见^[11]V.10, ^[26]Theorem 1, ^[16]3.3.2。

定理 6 可以看成欧几里得算法计算 f 和 g 的执行过程: 当且仅当 $\det(\text{Syl}(f, g))$ 为 0 时, f 和 g 在求出最大公因数之前就可以遇到余数为 0 的情况。事实上, Sylvester 矩阵的子行列式和多项式的余数序列之间有很深的关系。详见^[19], ^[15]3.6-3.9 或^[13]8。。我们这里只强调最重要的部分:

令 K 为一个域, 令 $f, g \in K[x]$ 为两个次数高于 $k \geq 0$ 的非零多项式。那么 f, g 的第 k 个子结式² $\text{sres}_k(f, g, x)$ 可以被定义为删除后 $2k$ 列矩阵和 α, β 的后 k 行的矩阵的行列式。

命题 8. f, g, k 的定义与上相同。那么, 用欧几里得算法计算出来的 f 和 g 的余数序列包含这一个 k 阶的余数, 当且仅当 f 和 g 的子行列式不为 0。特别地,

$$\text{sres}_i(f, g, x) = \forall 0 \leq i < k \Leftrightarrow \deg(\gcd(f, g)) \geq k \quad (1.2.9)$$

²子结式 (subresultant) 有两种不同的定义, 本文中的子结式是指标量子结式 (scalar subresultant), 在讨论多项式子结式这种定义也被称为主子结式系数 (principal subresultant coefficient)。^[28] 提供了较为系统的比较,^[19], ^[15], ^[13], ^[25] 详细介绍了多项式子结式的理论。

具体证明可以参考^[12]6.10。本文中，我们只需要将 (8) 限定到 $k = 2$ 的情况即可。直接的证明可见^[29]Theorem 5.2.5。结式矩阵的消去 现考虑双变量多项式 $f, g \in K[x, y]$ 。如果我们将 f 和 g 看做 $Q(K[x])[y]$ 中的多项式，命题 6 可以帮助我们求解 $f = g = 0$ 。求解过程需要两步来完成^[27]3.1:

1. 消元：在这一步我们确定一个解 ϵ ，即，允许方程 $f = g = 0$ 有解 (ϵ, η) 的 x 的值。
2. 求另外一根：这里我们通过所有可能的 y 值 η 来延伸解 $x = \epsilon$ 使得 $f(\epsilon, \eta) = g(\epsilon, \eta) = 0$

在这样的情境下， $\text{res}(f, g, y)$ 是 $K[x]$ 中的一个多项式，我们希望它的解集是 ϵ 集，因为一个结式矩阵的行列式为零意味着 $f|_\epsilon = g|_\epsilon = 0$ 。但是，结式矩阵的定义是建立在 f, g 的次数上。而当 ϵ 是 $l(f)$ 和 $l(g)$ 的解时，次数就会降低。这时 Sylvester 矩阵 $\text{Syl}(f, g)|_\epsilon$ 最左边的列是零，这时 $\text{res}(f, g, y)(\epsilon) = 0$ ，这种情况与 ϵ 是不是部分解没有关系。因为这些零向量的存在，建立结式矩阵求解出的结果不一定就是要求的零解。但是，如果其中一个多项式是 y 正规的，就可以得到一个理想情况：

命题 9. 令 K 为一个域，令 $f, g \in K[x, y]$ 为非零多项式。另外地，令 f 为 y 正规的，那么对于任何 $\epsilon \in K$ ，以下两种情况成立：

1. $\text{res}(f, g, y)(\epsilon) = 0$
2. 存在 $\eta \in \bar{K}$ ，使得 $f(\epsilon, \eta) = g(\epsilon, \eta) = 0$

证明. 我们证明 $\text{res}(f, g, y)(\epsilon) = 0$ 等价于 $\text{res}(f|_\epsilon, g|_\epsilon, y) = 0$ ：首先 $\text{Syl}(f, g)|_\epsilon$ 包含了 $\text{Syl}(f|_\epsilon, g|_\epsilon)$ 作为一个矩阵块 S 。那么 S 留下的 $d := \deg(g) - \deg(g|_\epsilon) \geq 0$ 列在对角线下都是 0，并且在对角线处 $l(f) \neq 0$ 。因此 $\det(\text{Syl}(f, g)|_\epsilon) = l(f)^d \det(\text{Syl}(f|_\epsilon, g|_\epsilon))$ ，得证。

在相交两个曲线时，我们意在寻找有限多个相交点，因此我们希望得到一个非零的结式矩阵。

命题 10. 令 K 为一个域，令 $f, g \in K[x, y]$ 为非零多项式，那么 $\text{res}(f, g, y) \neq 0$ 当且仅当 f 和 g 在 y 中没有幂数为正的公因式。

证明. 回忆 2.2.2 节中, $K[x]$ 是一个 UFD。那么推论 2 说明了作为 $K[x][y]$ 的元素, f 和 g 没有幂数为正的单变量公因式, 当且仅当他们作为 $Q(K[x])[y]$ 的元素, 没有正次数单变量的公因子。由命题 6, 可以导出此情况与 $\text{res}(f, g, y) \neq 0$ 等价。

另外一个麻烦在于计算的域不是代数封闭的 \mathcal{R} 。 $\text{res}(f, g, y)$ 的实数零解也许是由一个 x 坐标恰好为实数的多项式的复数解产生。然而, 复数解往往以共轭的形式成对出现。因此我们会在接下来保证解的唯一性, 这样 $\text{res}(f, g, y)$ 的实数零解就一定是来自多项式的实数解了。计算结式矩阵 我们已经考虑了计算两个多项式 f, g 的结式矩阵的三种选项:

1. 计算上文定义的 **Sylvester** 矩阵 $Syl(f, g)$ 的行列式, 这一步是最直观的
2. 计算上文定义的 **Bezout** 矩阵 $Bez(f, g)$ 的行列式, 这个行列式较小, 计算起来也更快
3. 计算 f, g 的余数多项式序列, 利用 2.2.2 节提到的子结式方法。该方法可以计算子结式行列式、余数多项式序列, 以及控制系数增长。特别地, 它还可以计算第零阶子结式, 即结式行列式。

在计算由环 (整数环或多项式环) 构成的矩阵的行列式时, 出于效率的考虑往往要省去分数的计算。我们为这种不需要分数的行列式计算提供了两种选项: **Gauss-Bareiss** 算法和 **Berkowitz**³ 算法。**Gauss-Bareiss** 算法需要 $O(n^3)$ 复杂度, 运算包括在环的容许范围内进行的除法操作。而 **Berkowitz** 需要 $O(n^4)$ 的运算, 不需要进行除法操作。

我们结式行列式消元法主要应用于 y 正规的三次双变量多项式 $f, g \in \mathbb{Z}[x][y]$ 。在这样一个特定的设置下, 我们通过使用检验数据, 可以观测到 **Berkowitz** 计算行列式的方法更快。而利用 **Berkowitz** 来衡量 **Bezout** 行列式比使用 **Sylvester** 行列式要快。这一点与最近 **Abdeljaoued** 等人的一项研究相一致。^[31] 参考文献中讲述了更高次数的多项式处理方法。

所以让我们看一看有着相同次数 m 的 $f, g \in K[x]$ 组成的 **Bezout** 矩阵。考虑加入了新变量 z 的多项式 $f(x)g(z) - f(z)g(x)$ 。它在 $x = z$ 处取值 0, 因此可

³这个算法命名自 **Rote**^[30]。也有一些人将这一算法的提出归功于 **Samuelson**

以被 $x - z$ 除尽。其结果是

$$\Lambda(x, z) := \frac{f(x)g(z) - f(z)g(x)}{x - z} = \sum_{i,j=0}^{m-1} c_{ij}x^i z^j \quad (1.2.10)$$

f 和 g 的 Bezout 矩阵是 Λ 的 $m \times m$ 的系数矩阵 $Bez(f, g) := (c_{ij})_{i,j=0}^{m-1}$ 。由 Gelfand, Kapranov 和 Zelevinsky 记录了下文的结论：

命题 11

令 K 为一个域。令 $f = \sum_{i=0}^m a_i x^i$ 以及 $g = \sum_{i=0}^n b_i x^i$ 为 $K[x]$ 中 m 次和 n 次的多项式，其中 $m \leq n$ 。

Bezout 矩阵的元素 $Bez(f, g) = (c_{ij})_{i,j=0}^{m-1}$ 写作

$$c_{ij} = \sum_{k=0}^{\min(i,j)} [k, i+j+1-k], [i, j] := a_i b_j - a_j b_i \quad (1.2.11)$$

这里约定对于 $i > m$ 有 $a_i = 0$ ，对 $i > n$ 有 $b_i = 0$

Bezout 矩阵的行列式是

$$\det(Bez(f, g)) = \pm a_m^{m-n} \text{res}(f, g, x) \quad (1.2.12)$$

$m > n$ 时，(11) 中 a_m 的出现高幂次的情况可以通过生成一个包含 n 个相似 Bezout 矩阵的行和 $m - n$ 个相似 Sylvester 的行的 $m \times m$ Bezout 混合矩阵来避免^[32]。我们利用 Berkowitz 方法，通过计算混合 Bezout 矩阵的行列式来计算 f 和 g 的结式⁴。

第 k 个子结式可以被表示成 (混合)Bezout 矩阵的 $(m - k) \times (m - k)$ 子行列式^[32]。这就是我们计算子结式的方法。

我们这里提供两个更深入的参考文献：Bezout 矩阵的 (子) 行列式的 (子) 结式的性质可以由 Goldman 等人提出的第一原则导出 (不需要通过 Sylvester 矩阵)。而通过构建 Bezout 矩阵来一次性消除好几个变量的问题得到了 Bikker 和 Uteshev 的研究^[33]，其研究还同样包括了本文中的单变量情况的结式性质⁵。

⁴最近的一篇文献^[31]介绍了如何通过执行 Berkowitz 方法来获得所有的子结式。虽然这篇文章很有趣，但是不会对我们在本文的讨论有太多贡献

⁵注意^[33]中考虑的矩阵 B 和我们讨论的 Bezoutiant $Bez(f, g)$ 不同。其相差了一组基的坐标变换

1.2.4 寻找和处理多项式的根

多项式根的隔离 从算法的角度来讲, 我们在求解方程的根时, 我们主要关注利用有理数区间来孤立这些根, 从而确定多项式 $f \in \mathbb{Q}[x]$ 的实数解。区间 $[l, r] \subset \mathbb{R}$ 如果满足 $\{x \in [l, r] | f(x) = 0\} = \{\xi\}$ 那么被称为根 $\xi \in \mathbb{R}$ 的隔离区间。

在开始前, 我们将 f 通过重数分成多个因式 (参见 2.2.3 章节)。然后我们分别确定每一个无平方因式的根。

如果 $\text{res}(f, f', x) \neq 0 \pmod{p}$ (这里 p 是某个质数) 我们就将多项式按重数分解为无平方因式。因为 $\text{res}(f, f', x) \neq 0$ 等价于 f 是无平方因式 (见章节 2.2.3 和 2.3.1)。检查因式以一个质数为模的结果, 这个质数应该足够小, 以便保证计算的精度, 详见^[34], 这种方法比计算 $\text{gcd}(f, f') = 1$ 要快很多。与浮点数过滤类似, 我们将此方法称为模过滤。无平方数的性质是我们计算多项式根的前提。模过滤让我们以高效率和准确率来处理相应问题, 而且比传统上的计算 gcd 的算法要简单。

在分解之后, 我们分别处理 $\sum_{i=0}^n a_i x^i$ 的每一个无平方因式。我们计算出如下的一个包含所有实根的区间 $[-B, B]$ 。通过不断尝试 $k = 0, 1, 2, 3, \dots$, 然后取 $B = 2^{k_0}$, 我们确定如下的式子:

$$k_0 := \min\{k \geq 0 | |a_n| 2^{nk} > \sum_{i=0}^{n-1} |a_i| 2^{ik}\} \quad (1.2.13)$$

在实际操作中, 我们使用 Descartes 方法⁶。该方法是建立在 Descartes 符号法则上的分治法。详细的证明可以参考 Collins 和 Akritas 的文献^[36], Krandick 的文献^[37] 以及 Rouillier 和 Zimmermann 的研究成果^[38]。

命题 12(Descartes 符号法则) 令 $f \in \mathbb{R}[x]$ 为一个非零多项式, 其中将重数计入在内共有 p 个根, 其系数序列有 v 次符号变换。那么 $v - p$ 非负且是偶数。

证明详见^[29]Theorem 7.2.6,^[39]Theorem 5.5 和^[40]。序列 a_0, \dots, a_n 的符号变换是一组满足 $\text{sign}(a_i)\text{sign}(a_j) = -1$ 和 $a_{i+1} = \dots = a_{j-1} = 0$ 的指数 $0 \leq i < j \leq n$ 。这里符号变换的次数是不考虑零系数的符号的。

⁶Uspensky 算法的使用很普遍但不正确^[35]

如果 $v = 0$ 或 $v = 1$ ，那么 v 就是正实根的个数。如果 $v > 1$ ，那么计数的时候可能把靠近正实轴的复数也计算进内了。

Descartes 法则对于任意一个开区间 $]l, r[\neq \mathbb{R}$, $l, r \in \mathbb{R} \cup \{\pm\infty\}$ 上的 f 的实根个数同样适用。具体的方法是采用一个函数复合 $f \circ T$ 。这里 T 是一个莫比乌斯变换 $T(x) = \frac{ax+b}{cx+d}$ ，将区间 $]0, +\infty[$ 放缩到 $]l, r[$ ，详见^[29]7.3.2。（这里反方向的括号代表区间不包括边界点）

Descartes 方法原理如下：最初的区间包含所有根，随后持续地分割区间，直到区间要么为空集要么有且仅有一个根，即 $v = 0$ 或 $v = 1$ 。关键是证明最后一定会出现 $v = 1$ 或 $v = 0$ 的情况。A.J.H. Vincent 于 1836 年最先证明出了这个事实。这里我们推荐读者参考^[40] 和^[41]。

另外一种可以履行 **Descartes** 方法的是 **Sturm** 序列。该序列是 f 和 f' 经过符号处理的多项式余数序列。详见^[29]7.2.2,^[39]5.4.1,^[15]Chapter 7，因此，除非在进行模过滤的时候跳过了这一步，否则在计算 f 无平方分解时会自然而然计算出 **Sturm** 序列。不过我们不采用这种方法来隔离方程的根。**Johnson**^[42] 实践证明了 **Sturm** 序列隔离方程的根通常不如 **Descartes** 法有效。

实代数数的表示和排序 令多项式 $f_0 \in \mathbb{Q}[x]$ 存在实根 $\vartheta \in \mathbb{R}$ 。章节 2.4.1 找到了根的一个表达方法 $\vartheta \sim (f, [l, r])$, $l, r \in \mathbb{Q}$ ，因为 $f \in \mathbb{Z}[x]$ 是 f_0 的一个无平方因式。在符号运算的领域，这是一种标准的表达方式^[26]。我们保证 $\vartheta \in]l, r[$ ，除去 $l = r = \vartheta$ 这种情况。

我们需要比较这种形式的数字。比较手段是使用二分法，在一个隔离区间 $]l, r[$ 选取点 $t \in]l, r[$ 。因为 f 是无平方的，因此 $f'(\vartheta) \neq 0$ ，因此连续可微的函数 $f: \mathbb{R} \rightarrow \mathbb{R}$ 在 ϑ 处没有极值，因此在这一点变号。又因为 $[l, r]$ 中没有其他零根，我们可以得到 $\pm 1 = \text{sign}(f(l)) \neq \text{sign}(f(r)) = \mp 1$ 。通过比较 $\text{sign}(f(t))$ 和 $\text{sign}(f(r))$ ， 0 ， $\text{sign}(f(l))$ ，我们可以确定 $\vartheta \in]l, t[$, $\vartheta = t$ 还是 $\vartheta \in]t, r[$ ，然后相应地把 $[l, r]$ 改成 $[l, t]$, $[t, t]$ 或 $[t, r]$ 。

令 $\vartheta_1 \sim (f_1, [l_1, r_1])$ 和 $\vartheta_2 \sim (f_2, [l_2, r_2])$ 为两个用来比较的代数数。如果它们的隔离区间交集非空，我们就以另外的隔离区间为边界，在该隔离区间上使用二分法，以得到两个不相交的区间。这时代数数的排序问题就可以由边界的大小给出。

现在我们假设 $[l, r] := [l_1, r_1] = [l_2, r_2]$ ，这里 $l < r$ 。多项式 f_1, f_2 是无平方

的, 且在区间 $[l, r]$ 中都有根。考虑无平方项的 $d := \gcd(f, g)$ 。如果 $\text{sign}(d(l)) \neq \text{sign}(d(r))$, 那么 $[l, r]$ 中 f, g 有共同根 $\vartheta_1 = \vartheta_2$ 。否则 d 在 $[l, r]$ 中没有根, 即 $\vartheta_1 \neq \vartheta_2$ 。我们就可以通过在 $\frac{1}{2}(l_i + r_i)$ 上使用二分法来对根进行区间隔离。这样的迭代是可以终止的, 因为区间的长度会最终收敛到 0, 所以最终可以达到区间隔离的目标。

在比较 ϑ_1, ϑ_2 之后, 就可以用改进过的隔离区间代表它们。 $\gcd d$ 计算出来后, 我们可以用 d 或者 f_i/d 替代在 ϑ_i 取 0 的 f_i 。

就像在 2.4.1 节所讲, 用模过滤的操作是可行的。在比较两个代数数 ϑ_1 和 ϑ_2 , 和它们对应的函数应该是互质的。因此在计算 $d := \gcd(f_1, f_2)$ 之前, 我们可以通过检查 $\text{res}(f_1, f_2, x)$ 是否恒等于 $0 \pmod{p}$ 。如果成立, 那么 $d = 1$ 。更多的方法可以参考^[43]。

上文方法的一个变形可以用来判断对于 $\vartheta \sim (f, [l, r])$, $g(\vartheta)$ 是否为 0。计算 $d := \gcd(f, g)$, 通过观察 l 和 r 处的符号来判断 d 在 ϑ 处是否为 0。

这种表示实代数数的方法可以通过一定数量的符号运算, 精确地寻找多项式的根。代数数的大小比较可以在区间得到修改后完成。其大小不等式关系以互不相交的区间的形式表示。在区间修改的过程中计算出的数均是有理数。

除此之外我们还需要对代数数执行一步操作: 即确保 ϑ 的隔离区间内也不会有其他具有无平方因式性质的 $g \in \mathbb{Z}[x]$ 的根。为了达到这一步, 我们要对区间 $[l, r]$ 不断使用二分法, 直到满足以下两个条件之一: 要么研究区间中不存在 g 的根, 要么 g 在区间中有且仅有一个根, $\gcd(f, g)$ 在区间中如果有符号变换, 说明 f 和 g 的根是相同的。这些条件都可以用 **Descartes** 方法来判断。

对于这种运算, 我们希望根 ϑ 在区间的内部。所以如果二分法中某一点恰好为方程的根, 我们就需要另外找一个点来代替。上文提到的用模余数来检验互质性也可以应用于此。

带平方根的计算 在下一章几何分析的几个特殊例子中, 我们需要计算带有平方根 $\sqrt{D}, D > 0$ 的式子, 不过大多数分析运算都是只涉及整数和有理数的。目前有两种计算带有平方根式子的方法:

第一种是分离界限数类型, 如 $\text{CORE} :: \text{Expr}^{[44]}$ 和 $\text{leda} :: \text{real}^{[45]}$ 。这些类型支持代数表达式, 包括对整数运用的开根号操作。它们通过在数值上逼近和分离边界来比较, 以确定在哪个点的精度足以检测出等式。比较最适用于嵌套深度

较小和值差异较大的表达式。

第二种是符号表示。在这里，计算代数数的最小多项式的方法被简化成了“铅笔和纸”的表达 $\mathbb{Q}(\sqrt{D}) = \{a + b\sqrt{D} | a, b \in \mathbb{Q}\}$ 。在检验 D 不是一个平方数后，我们就可以得出 $a + b\sqrt{D} = a' + b'\sqrt{D} \Leftrightarrow (a, b) = (a', b')$ 。这种方法对于 $\mathbb{Q}(\sqrt{D})[x]$ 上的欧几里得算法较为适用，因为这种表达不再有通过建立树来表达的需要。

我们对第四节几何分析的算法采用了符号表示。我们用 $CORE :: Expr$ 或 $leda :: real$ 来比较根的表达式，但是在这些情况下这些表达式往往都是不相等的。这里我们只采用它的数值逼近的功能，和不依赖分离界限。关于两种表达方式如何共同使用，详情可以见 4.2.4 “m=4” 的情况。

四、外文原文



Available online at www.sciencedirect.com



Computational Geometry 35 (2006) 36–73

Computational
Geometry

Theory and Applications

www.elsevier.com/locate/comgeo

Exact, efficient, and complete arrangement computation for cubic curves[☆]

Arno Eigenwillig^{a,*}, Lutz Kettner^a, Elmar Schömer^b, Nicola Wolpert^a

^a Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany

^b Johannes-Gutenberg-Universität Mainz, Institut für Informatik, Staudingerweg 9, 55099 Mainz, Germany

Received 6 August 2004; accepted 5 August 2005

Available online 28 November 2005

Communicated by J.-D. Boissonnat and J. Snoeyink

Abstract

The Bentley–Ottmann sweep-line method can compute the arrangement of planar curves, provided a number of geometric primitives operating on the curves are available. We discuss the reduction of the primitives to the analysis of curves and curve pairs, and describe efficient realizations of these analyses for planar algebraic curves of degree three or less. We obtain a *complete*, *exact*, and *efficient* algorithm for computing arrangements of cubic curves. Special cases of cubic curves are conics as well as implicitized cubic splines and Bézier curves.

The algorithm is *complete* in that it handles all possible degeneracies such as tangential intersections and singularities. It is *exact* in that it provides the mathematically correct result. It is *efficient* in that it can handle hundreds of curves with a quarter million of segments in the final arrangement. The algorithm has been implemented in C++ as an EXACUS library called CUBIX.

© 2005 Elsevier B.V. All rights reserved.

Keywords: Arrangements; Algebraic curves; Sweep-line algorithm; Robustness; Exact geometric computation

1. Introduction

1.1. Problem statement

The Bentley–Ottmann sweep-line method [9] can be used to compute the arrangement of planar curves. One *only* has to provide a number of geometric primitives (break a curve into x -monotone pieces, given two curves compute their intersections, compare two intersections or endpoints lexicographically, etc.). The “only” is the crux of the matter; not in principle, but in terms of efficient realization.

We discuss the mathematics of the primitives for cubic curves, i.e., planar algebraic curves of degree three (or less), and derive efficient realizations. Conics as well as implicitized cubic splines and Bézier curves are special cases

[☆] Partially supported by the IST Programme of the European Union as a Shared-cost RTD (FET Open) Project under Contract No. IST-2000-26473 (ECG—Effective Computational Geometry for Curves and Surfaces).

* Corresponding author.

E-mail addresses: arno@mpi-inf.mpg.de (A. Eigenwillig), kettner@mpi-inf.mpg.de (L. Kettner), schoemer@uni-mainz.de (E. Schömer), wolpert@mpi-inf.mpg.de (N. Wolpert).

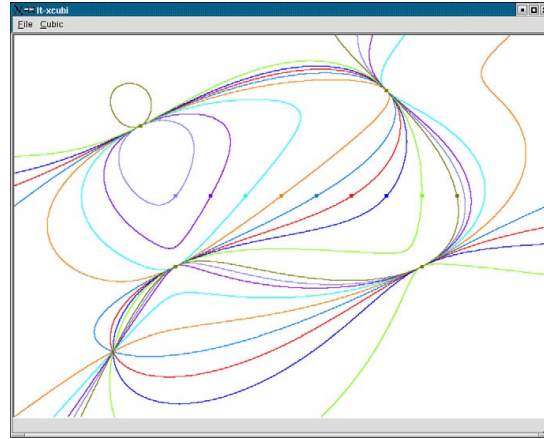


Fig. 1. A screenshot of our implementation showing a pencil of nine cubics with four common tangential intersections.

of cubic curves. We obtain a *complete* (it handles all possible degeneracies), *exact* (it provides the mathematically correct result), and *efficient* (it can handle hundreds of curves with a quarter million of resulting segments) sweep-line algorithm for computing planar arrangements of cubic curves.

With only minor modifications our geometric primitives can also be used to realize incremental approaches. Our implementation can be easily extended to compute arrangements of cubic segments and to perform regularized boolean operations on polygons bounded by them.

1.2. Related work

Complete, exact, and efficient implementations for planar arrangements of straight line segments exist, e.g., in LEDA [62, Chapter 10.7], and in the planar map [34] and planar Nef-polyhedron [74] classes of CGAL. However, existing implementations for curved objects are either incomplete, inexact, or not aimed at efficiency except for some recent work on circle and conic arcs (see below). For cubic curves we are not aware of any complete, exact, and efficient implementation.

Our work is influenced by results from three different communities: computer aided geometric design (CAGD), computational geometry, and computer algebra. The problem of computing intersections of curves and surfaces has a long history in CAGD. The CAGD community concentrates on approximate solutions by numerical methods which cannot distinguish, e.g., between a tangential intersection and two intersections lying very close together. Complete and exact implementations have been addressed recently: MAPC [50] is a library for exact computation and manipulation of algebraic curves. It offers arrangements of planar curves but does not handle all degenerate situations. Yap [82] uses separation bounds to make the traditional subdivision algorithm for the intersection of Bézier curves exact and complete in the presence of tangential and almost-tangential intersections.

Arrangements, mostly of linear objects, are also a major focus in computational geometry; see the survey articles of Halperin [43] and Agarwal and Sharir [2]. Many exact methods for curved objects have been formulated for the Real RAM model of computation [67], which allows unit-time operations on arbitrary real numbers and conceals the high cost of exact arithmetic with algebraic numbers.

Sakkalis [70,71], Hong [45], and Gonzalez-Vega and Necula [42] analyze the topology of a single real algebraic curve. Their approaches, like ours, can be seen as special forms of Cylindrical Algebraic Decomposition [5,23]. However, they do not consider the interaction between pairs of curves, and the full algebraic machinery they deploy is not necessary for cubic curves. The recent textbook by Basu, Pollack, and Roy [8] is a very useful reference for the algebraic background of those methods and also contains a curve topology algorithm. Aspects of the crucial problem to capture behavior at irrational points by rational arithmetic were treated by Canny [20] (Gap Theorem) and Pedersen [66] (multivariate Sturm sequences).

Predicates for arrangements of circular arcs that reduce all computations to sign determination of polynomial expressions in the input data are treated by Devillers et al. [26]. Recent work by Emiris et al. [31,32] discusses some predicates on conics in this style. However, these approaches do not extend easily to more complicated curves.

Exact, efficient, and complete algorithms for planar arrangements have been published by Wein [77] and Berberich et al. [11] for conic segments, and by Wolpert [73,78] for special quartic curves as part of a surface intersection algorithm. A generalization of Jacobi curves (used for locating tangential intersections) is described by Wolpert [79].

The work presented here follows the Master's thesis of the first author [28] and has appeared as an extended abstract in [29].

1.3. Our results

What are the difficulties in going from straight lines and straight line segments to conics and further on to cubic curves? The defining equations and thus the geometry of the basic objects and the coordinates of their intersections become more complicated. For straight line segments with rational endpoints, all vertices in their arrangement have rational coordinates. In the case of curves, the intersections are solutions to systems of non-linear polynomial equations and thus, in general, irrational algebraic numbers. We review polynomials, algebraic numbers, and elimination of variables among polynomial equations in Section 2.

The sweep-line algorithm works on x -monotone segments. Lines and line segments are x -monotone, conics need to be split at points of vertical tangent, and cubic curves need to be split at points of vertical tangent and at singularities. These notions and other required elements of the geometry of algebraic curves are introduced in Section 3. Cubic curves have a more diverse geometry than conics. Its analysis is a separate step in our algorithm, explained in Section 4.1.

We turn to pairs of curves. Two lines intersect in a single point with rational coordinates. Two conics intersect in up to four points. Each of the intersecting conic arcs can be parametrized in the form $y(x)$ using a single square root. In the case of a tangential intersection, the x -coordinate of the intersection can also be written as an expression involving a single square root. Therefore, previous work on conics [11,77] could make heavy use of the existing efficient methods for arithmetic within FRE [18,48], the field of real root expressions. FRE is the closure of the integers under the operations $+$, $-$, $*$, $/$, and $\sqrt[k]{}$ for arbitrary but fixed k ; it is a subfield of the real algebraic numbers. However, the $\sqrt[k]{}$ operation is restricted to extracting real-valued roots, thus general polynomial equations of degree ≥ 3 are not solvable in FRE, severely limiting its applicability to cubics. (The usability of recent additions for arithmetic with all real algebraic numbers [19,55,72] remains to be investigated.) Hence we need more powerful techniques. We discuss them and the geometric analysis of curve pairs in Section 4.2.

The central idea of our approach to curve and curve pair analyses is to exploit geometric properties of cubic curves in order to avoid arithmetic with irrational numbers as far as possible.

In Section 5 we put everything together and discuss high-level issues of the Bentley–Ottmann sweep for algebraic curves and how the required predicates reduce to curve and curve pair analyses. This part applies to algebraic curves of any degree.

We conclude with a discussion of running time from a theoretical (Section 6) and especially an experimental (Section 7) point of view. We point out that there is a full implementation of our algorithm.

2. Algebraic foundations

The algorithmic handling of algebraic curves rests to a large extent on algebraic operations. We give a concise summary of the existing results we use, just enough to make our presentation self-contained and accessible to non-experts in symbolic computation. This summary also provides the necessary context for describing the implementation choices we made.

As reference for the abstract algebra involved here we mention the textbook by Lang [54]. The algorithmic aspects are covered in the computer algebra books by Geddes, Czapor and Labahn [38], Akritas [4], Cohen [21], and von zur Gathen and Gerhard [36]. The recent book by Basu, Pollack and Roy [8] combines the viewpoints of real algebraic geometry and computer algebra.

五、中期检查

- 1 项目概况
- 2 工作进展情况
- 3 问题与建议
- 4 其他

毕业设计开题报告、外文翻译的考核

导师对开题报告、外文翻译的评语及成绩评定：

成绩比例	开题报告 占（15%）	外文翻译 占（5%）
分值	15	5

导师签名 _____

年 月 日

学院盲审专家对开题报告、外文翻译的评语及成绩评定：

成绩比例	开题报告 占（15%）	外文翻译 占（5%）
分值		

开题报告审核负责人（签名/签章） _____

年 月 日

毕业设计中期报告考核

导师对中期报告的评语及成绩评定：

成绩比例	中期报告 (10%)
分值	15

指导教师（签名）_____

年 月 日