

**UNIVERSITY
OF THE NATIONAL
EDUCATION COMMISSION, KRAKOW**

DOCTORAL SCHOOL

DISCIPLINE: MATHEMATICS

Maciej Zięba

DOCTORAL DISSERTATION

**Geometry of Symmetric Configurations
of Points in Projective Spaces**

Supervisor: prof. dr hab. Tomasz Szemberg

Krakow, 2026

**UNIWERSYTET KOMISJI EDUKACJI NARODOWEJ
W KRAKOWIE**

SZKOŁA DOKTORSKA

DYSCYPLINA: MATEMATYKA

Maciej Zięba

ROZPRAWA DOKTORSKA

**Geometria symetrycznych konfiguracji
punktów w przestrzeniach rzutowych**

Promotor: prof. dr hab. Tomasz Szemberg

Kraków, 2026

Streszczenie

Rozprawa bada szczególne konfiguracje punktów, prostych i krzywych w geometrii rzutowej poprzez algebrę ich ideałów jednorodnych. Jej motywem przewodnim jest obserwacja, że klasyczne zjawiska incydencyjne często mają precyzyjne odpowiedniki algebraiczne: potęgi symboliczne opisują zadane krotności znikania, potęgi zwykłe kodują ograniczenia faktoryzacyjne, funkcje Hilberta mierzą zjawiska interpolacyjne, a projekcje pozwalają przenosić konfiguracje przestrzenne do płaskich przecięć zupełnych.

Pierwsza część pracy zawiera ogólne tło matematyczne potrzebne w dalszych rozdziałach. Przypomniane są skończone podschematy przestrzeni rzutowej, ideały jednorodne, funkcje Hilberta, przecięcia zupełne, potęgi symboliczne, problemy zawierania, testy przynależności za pomocą baz Gröbnera, projekcje skończonych zbiorów punktów, problemy interpolacyjne typu Weddle’a oraz prawo grupowe na gładkiej krzywej kubicznej. Celem tego wprowadzenia jest przedstawienie późniejszych przykładów jako konstrukcji geometrycznych, a nie tylko jako izolowanych obliczeń.

Kolejne rozdziały dotyczą czterech rodzin konfiguracji. Układ Yoshinagi służy do badania niepowodzenia zawierania $I^{(3)} \subseteq I^2$ dla ideałów punktów na płaszczyźnie. Konfiguracja $\mathbf{H}_4$ daje wyjątkowy przykład zbioru geproci w $\mathbb{P}^3$, którego ogólna projekcja na płaszczyznę jest przecięciem zupełnym, ale nie wynika z elementarnej konstrukcji kratowej. Loci typu Weddle’a są omawiane jako zbiory wierzchołków stożków przechodzących przez zadane punkty bazowe, ze szczególnym uwzględnieniem różnicy między rachunkiem interpolacyjnym a udowodnionymi stopniami. Ostatnia część pracy opisuje punkty fleksyjne, punkty sekstaktyczne oraz punkty typu 9 na kubice Fermata za pomocą odwzorowania stycznia–residuum i torsji w prawie grupowym krzywej eliptycznej.

Słowa kluczowe: potęgi symboliczne, problem zawierania, zawieranie Harbourne’a–Huneke, układy prostych, przecięcia zupełne, zbiory geproci, powierzchnie Weddle’a, kubika Fermata, punkty sekstaktyczne, punkty torsyjne.

Abstract

This dissertation studies special configurations of points, lines, and curves in projective geometry through the algebra of their homogeneous ideals. The guiding theme is that classical incidence phenomena often have precise algebraic counterparts: symbolic powers detect assigned multiplicities, ordinary powers encode factorization constraints, Hilbert functions measure interpolation, and projections translate spatial configurations into plane complete intersections.

The first part of the dissertation gives the general mathematical background needed for the later chapters. It reviews finite subschemes of projective space, homogeneous ideals, Hilbert functions, complete intersections, symbolic powers, containment questions, Gröbner-basis membership tests, projections of finite sets, Weddle-type interpolation problems, and the group law on a smooth plane cubic. The discussion is written so that the later examples can be read as geometric constructions rather than as isolated computations.

The subsequent chapters treat four families of configurations. Yoshinaga's arrangement is used to study the failure of the containment $I^{(3)} \subseteq I^2$ for ideals of points in the plane. The $\mathbf{H}_4$ configuration gives an exceptional geproci set in $\mathbb{P}^3$, whose general projection to the plane is a complete intersection but which is not explained by the elementary grid construction. Weddle-type loci are discussed as vertex loci of cones through fixed base points, with careful attention to the distinction between interpolation counts and proved degree statements. Finally, flexes, sextactic points, and type-9 points on the Fermat cubic are described using the tangent-residual map and torsion in the elliptic-curve group law.

Keywords: symbolic powers, containment problem, Harbourne–Huneke containment, line arrangements, complete intersections, geproci sets, Weddle surfaces, Fermat cubic, sextactic points, torsion points.

Acknowledgments

I am deeply grateful to my supervisor, Professor Tomasz Szemberg, for his guidance, patience, and trust. He believed in my abilities, challenged me to grow, and continually encouraged me to take the next step.

My deepest thanks go to my wife, whose patience, understanding, and unwavering belief in me carried me through the most difficult moments. She supported me when I struggled and helped me move forward whenever I doubted I could continue.

This dissertation would not have been completed without them.

Contents

Streszczenie	iii
Abstract	v
Acknowledgments	vii
1 Introduction	1
1.1 Special configurations in projective geometry	1
1.2 From configurations to ideals	2
1.3 Symbolic-power containments	2
1.4 Projections and geproci configurations	3
1.5 Weddle loci and interpolation by cones	3
1.6 Higher contact on the Fermat cubic	4
1.7 Organization of the dissertation	4
2 Preliminaries	5
2.1 Projective spaces and homogeneous coordinate rings	5
2.2 Hilbert functions	6
2.3 Complete intersections	7
2.4 Intersection multiplicity and Bézout’s theorem	8
2.5 Symbolic powers	9
2.6 Containment theorems and counterexamples	9
2.7 Initial degrees and resurgence	10
2.8 Cayley–Bacharach principle	11
2.9 Minimal free resolutions	12
2.10 Normal forms and membership tests	13
2.11 Linear systems and projections	14
2.12 Grids and geproci sets	14
2.13 Weddle loci	15
2.14 The group law on a smooth plane cubic	16

3	Yoshinaga's Arrangement and Containment	19
3.1	Background and Motivation	19
3.2	Yoshinaga's Arrangement	20
3.2.1	The explicit line equations	22
3.3	The Non-Containment Statement	23
3.4	Computational Verification	25
3.4.1	Completeness of the singular locus	26
3.4.2	Point ideals and symbolic powers	26
3.4.3	Normal-form tests	27
3.4.4	Exact incidence and containment certificate driver	27
3.5	Conclusion	28
4	The H_4 Configuration and General Projections	29
4.1	The H_4 Configuration	29
4.2	The Geproci Property	30
4.3	Incidence Structure	33
4.4	Computational Checks	37
4.4.1	SINGULAR certificate drivers	38
4.5	Conclusion	39
5	The Weddle Surface and Analogous Loci	41
5.1	The Classical Weddle Surface	42
5.2	Cubic Cones and the Surface W_{10}	44
5.2.1	Determinantal description	45
5.3	The Curve of Vertices for Eleven Points	46
5.3.1	Exact rational SINGULAR specialization driver	47
5.4	Weddle Surfaces for Cones of Degree n	48
5.5	Conclusion	50
6	Sextactic and Type-9 Points on the Fermat Cubic	51
6.1	Introduction	51
6.2	Sextactic Points on the Fermat Cubic	51
6.2.1	Tangent lines and incidence	53
6.2.2	Shadow points of the sextactic conic arrangement	53
6.3	Points of Exact Order 9 on the Fermat Cubic	54
6.3.1	The tangent-residual map and its period-3 orbit structure	55
6.4	Conics through Sextactic and Type-9 Points	56
6.4.1	Intersection structure of conics through a fixed point	57
6.5	Conclusion	57

<i>CONTENTS</i>	xi
7 Concluding remarks	59
A Computational Sources	61
A.1 Reusable libraries	61
A.1.1 Shared certificate utilities	61
A.1.2 Symbolic powers library	62
A.1.3 Yoshinaga arrangement library	65
A.1.4 Fermat auxiliary checks	69
A.2 Yoshinaga certificate implementation	71
A.3 $\mathbf{H}_4$ certificates	78
A.4 Cubic Weddle specialization	87
Bibliography	93

1 Introduction

1.1 Special configurations in projective geometry

Projective geometry is rich in configurations whose definition is elementary but whose behavior is unexpectedly rigid; the general background used here is classical and can be found, for example, in Hartshorne [21] and Dolgachev [9]. A line arrangement may be described by writing down a product of linear forms, but the distribution of its double, triple, and higher multiple points can force delicate algebraic consequences. A finite set of points in three-dimensional projective space may look like a discrete object, but its general plane projection can remember hidden structure and become a complete intersection. A point on a smooth cubic may be studied by drawing tangent lines and osculating conics, but the true organizing principle is the group law on the cubic.

This dissertation is devoted to such examples. The emphasis is not on projective configurations as pictures alone, and not on homogeneous ideals as formal algebra alone, but on the translation between the two. The same object is viewed simultaneously as a configuration of points, lines, or curves and as an ideal in a graded polynomial ring. Incidence becomes vanishing. Multiplicity becomes symbolic membership. A complete intersection becomes a numerical and homological constraint. A tangent construction becomes a statement about torsion on an elliptic curve.

The configurations studied here are classical or classically inspired. Yoshinaga's arrangement is a deformation of a Fermat–Ceva arrangement of lines and is discussed in relation to the containment problem in [29, 10]. The configuration denoted by $\mathbf{H}_4$ comes from the root system H_4 and the geometry of the 600-cell; its geproci interpretation follows the work of Chiantini–Migliore [5] and Wiśniewska–Zięba [31]. Weddle surfaces arise from the nineteenth-century problem of finding vertices of cones through fixed points in space [30, 1, 23]. The Fermat cubic supplies a highly symmetric elliptic curve on which flexes, sextactic points, and higher contact points can be described explicitly [4, 16, 27, 24].

The common feature is that each construction produces a finite set of points with more structure than a general set of the same cardinality. The main task is to say exactly what this extra structure implies, and equally importantly, what it does not imply.

1.2 From configurations to ideals

Let $\mathbb{K}$ be an algebraically closed field and let $S = \mathbb{K}[x_0, \dots, x_N]$ be the homogeneous coordinate ring of $\mathbb{P}_{\mathbb{K}}^N$. A finite set of points $Z \subset \mathbb{P}_{\mathbb{K}}^N$ is encoded by its saturated homogeneous ideal, in the sense recalled in Definition 2.1; standard references for this language are [21, 13, 25]:

$$I(Z) = \bigcap_{p \in Z} I(p).$$

The Hilbert function of $S/I(Z)$ measures the number of independent conditions imposed by Z on hypersurfaces of each degree. If the points are general, this function is usually governed by dimension counts. For special configurations, however, incidence and symmetry can force unexpected dependencies.

One recurrent phenomenon is the tension between symbolic and ordinary powers; see [7, 2, 3] for broad introductions to this comparison. The symbolic power $I(Z)^{(m)}$ consists of forms vanishing to order at least m at every point of Z , as recalled in Definition 2.8 and used concretely in Lemma 2.9. It is therefore geometric in nature. The ordinary power $I(Z)^r$, on the other hand, consists of sums of products of r forms from $I(Z)$ and is sensitive to the way the ideal is generated. The inclusion $I(Z)^r \subseteq I(Z)^{(r)}$ is automatic, but the reverse direction is more subtle.

This distinction becomes concrete for line arrangements. If $L = L_1 \cdots L_s$ is a product of line equations, then the vanishing order of L at a point is exactly the number of arrangement lines passing through that point. Thus a product of lines gives an immediate symbolic-power candidate. The difficult part is to decide whether the same product lies in an ordinary power. For Yoshinaga's arrangement, this distinction produces a failure of the $I^{(3)} \subseteq I^2$ containment.

1.3 Symbolic-power containments

Containment problems for symbolic powers form one of the central algebraic threads of the dissertation. General theorems of Ein–Lazarsfeld–Smith and Hochster–Huneke, recalled precisely in Theorem 2.11, imply uniform containments of symbolic powers inside ordinary powers in regular rings [12, 22]. In the case of finite point sets in the plane, these results guarantee

$$I^{(2r)} \subseteq I^r.$$

It is natural to ask whether the exponent $2r$ can be improved. A particularly tempting strengthening for points in the plane is

$$I^{(2r-1)} \subseteq I^r.$$

For $r = 2$ this becomes the containment $I^{(3)} \subseteq I^2$.

Special point sets arising from line arrangements show that this proposed improvement fails. The failure is not merely numerical. One must produce a concrete form that vanishes to the required symbolic order and then prove that it is not contained in the ordinary power. In this thesis this is done by combining elementary multiplicity arguments with exact normal-form computations over explicitly defined fields.

1.4 Projections and geproci configurations

A second guiding problem concerns finite subsets of $\mathbb{P}^3$ and their general projections to $\mathbb{P}^2$. Following Chiantini and Migliore [5], a finite set $Z \subset \mathbb{P}^3$ is called geproci if a general projection of Z to the plane is a complete intersection; the formal definition used in the thesis is Definition 2.17, and complete intersections are recalled in Definition 2.3. Grids provide the basic examples: if one takes points on two skew lines and joins them in the natural way, a general projection can produce the intersection of two plane curves.

The interest of the subject comes from non-grid examples. The $\mathbf{H}_4$ configuration is a distinguished example: it comes from a highly symmetric root system and has a general projection that is a complete intersection of type $(6, 10)$. Its behavior is not explained by the elementary grid construction. This makes it a useful test case for understanding which spatial configurations can be recognized by their projected complete-intersection structure.

The thesis treats the $\mathbf{H}_4$ example as a bridge between finite geometry, root systems, and commutative algebra. The complete-intersection property is a geometric statement about general projections, but it is reflected algebraically in the Hilbert function and the ideal of the projected point set.

1.5 Weddle loci and interpolation by cones

The Weddle part of the dissertation concerns interpolation problems in $\mathbb{P}^3$, following the classical tradition of Weddle, Baker, and Hudson [30, 1, 23]. Fix points $p_1, \dots, p_s \in \mathbb{P}^3$. For a point $v \in \mathbb{P}^3$, one can ask whether there exists a cone with vertex v and prescribed degree passing through all the p_i . As v varies, the set of such vertices is a Weddle-type locus.

The classical Weddle surface appears for quadric cones through six general points in $\mathbb{P}^3$ and is a singular quartic surface; this is the object formalized in Definition 5.1. Its construction is elementary in spirit: one writes down a linear system of quadrics constrained to be singular at a chosen vertex and to pass through fixed base points. The resulting determinant condition defines the vertex locus.

For higher-degree analogs, the thesis uses the Macaulay-duality multiplication map in the quotient by powers of the dual linear forms. Its determinantal ideals define the Weddle schemes directly, while their expected codimensions determine the relevant degrees and Cohen–Macaulay properties.

1.6 Higher contact on the Fermat cubic

The final geometric theme is the study of higher-order contact on the Fermat cubic

$$x^3 + y^3 + z^3 = 0.$$

A smooth plane cubic becomes an elliptic curve after choosing a flex as the origin [26, 15], and the notions of flex, sextactic point, and type-9 point are fixed in Definition 2.19. The tangent line at a point p meets the cubic again at a residual point $\nu(p)$, and in the group law this map is

$$\nu(p) = -2p.$$

This simple formula governs much of the special contact geometry of the curve.

Flexes correspond to 3-torsion. Sextactic points are points at which an irreducible conic has contact order six with the cubic. Type-9 points are non-flex points at which an irreducible cubic distinct from the original cubic has contact order nine. These notions are geometric, but on a smooth cubic they are controlled by torsion conditions. The Fermat cubic is especially suitable for an explicit discussion because its symmetry makes the relevant torsion sets accessible.

1.7 Organization of the dissertation

Chapter 2 fixes the preliminaries and notation for projective point schemes, complete intersections, symbolic powers and containments, projections and geproci sets, Weddle loci, and the group law on a smooth cubic.

Chapter 3 discusses Yoshinaga’s arrangement and the containment failure attached to its singular points. Chapter 4 treats the $\mathbf{H}_4$ configuration and the geproci property. Chapter 5 reviews Weddle surfaces and related vertex loci. Chapter 6 studies flexes, sextactic points, and type-9 points on the Fermat cubic. Chapters 3–5 include theorem-specific SINGULAR certificates, while Chapter 6 contains exact auxiliary checks of the flex counts and tangent–residual identities used there. The computational appendix contains the corresponding implementations and reusable libraries.

2 Preliminaries

This chapter collects the basic notions and tools that will be used throughout the thesis. We recall the standard language of projective point schemes, saturated ideals, Hilbert functions, complete intersections, symbolic powers, projections, Weddle loci, and minimal free resolutions. The aim is not to give a complete survey, but rather to fix notation and to isolate the algebraic and combinatorial ingredients needed in the later chapters. Standard background references for this chapter are Hartshorne [21], Eisenbud [13], Fulton [15], Migliore [25], and the survey of symbolic powers by Dao–De Stefani–Grifo–Huneke–Núñez-Betancourt [7].

2.1 Projective spaces and homogeneous coordinate rings

The conventions in this section are standard; see, for instance, Hartshorne [21, Chapter I] and Eisenbud [13, Chapters 1–2].

Let $\mathbb{K}$ be an algebraically closed field unless explicitly stated otherwise. The projective space $\mathbb{P}^N = \mathbb{P}_{\mathbb{K}}^N$ is the set of one-dimensional linear subspaces of $\mathbb{K}^{N+1}$. Its homogeneous coordinate ring is

$$S = \mathbb{K}[x_0, \dots, x_N],$$

graded by the total degree. A homogeneous ideal $I \subseteq S$ defines the projective variety

$$V(I) = \{p \in \mathbb{P}^N : f(p) = 0 \text{ for all homogeneous } f \in I\}.$$

Conversely, a projective subscheme is described by a saturated homogeneous ideal.

For a point $p = [a_0 : \dots : a_N] \in \mathbb{P}^N$, the ideal $I(p)$ is generated by the linear forms vanishing at p . If $Z = \{p_1, \dots, p_s\}$ is a reduced finite set of points, then

$$I(Z) = \bigcap_{i=1}^s I(p_i).$$

More generally, a zero-dimensional subscheme is described by a saturated homogeneous ideal whose quotient ring has Krull dimension one.

Definition 2.1 (Saturation)

Let $\mathfrak{m} = (x_0, \dots, x_N)$ be the irrelevant ideal. The saturation of a homogeneous ideal $I \subseteq S$ is defined as

$$I^{\text{sat}} = \{f \in S : \mathfrak{m}^t f \subseteq I \text{ for some } t \gg 0\}.$$

The ideal I is saturated if $I = I^{\text{sat}}$.

2.2 Hilbert functions

For Hilbert functions of graded modules and finite projective schemes we follow the standard treatments in [13, 25].

Let $M = \bigoplus_{d \geq 0} M_d$ be a finitely generated graded S -module. Its Hilbert function is

$$H_M(d) = \dim_{\mathbb{K}} M_d.$$

For a finite scheme $Z \subset \mathbb{P}^N$, the Hilbert function of Z is

$$H_Z(d) = \dim_{\mathbb{K}}(S/I(Z))_d.$$

If Z is reduced and consists of s points, then $H_Z(d) = |Z| = s$ for all sufficiently large d .

The value of $H_Z(d)$ can be interpreted as the number of independent conditions imposed by Z on hypersurfaces of degree d . Indeed,

$$\dim_{\mathbb{K}} S_d = \binom{N+d}{N},$$

and the degree- d piece $I(Z)_d$ is the vector space of degree- d forms vanishing on Z . Hence

$$H_Z(d) = \dim S_d - \dim I(Z)_d.$$

A special configuration is often detected by the failure of Z to impose independent conditions in some degree.

Definition 2.2 (Initial degree and regularity index)

For a homogeneous ideal $I \subseteq S$, its initial degree is

$$\alpha(I) = \min\{d : I_d \neq 0\}.$$

For a finite set of points Z , the regularity index is the least degree from which the Hilbert function of $S/I(Z)$ equals $|Z|$.

2.3 Complete intersections

A complete intersection is one of the most rigid kinds of projective schemes; for background see [13, 21, 25].

Definition 2.3 (Complete intersection)

Let $X \subset \mathbb{P}^N$ be a subscheme of codimension c with saturated homogeneous ideal I_X . We say that X is a complete intersection of type $(d_1, \dots, d_c)$ if

$$I_X = (F_1, \dots, F_c)$$

where $F_1, \dots, F_c$ is a homogeneous regular sequence and $\deg F_i = d_i$.

In the plane, a reduced complete intersection of type (a, b) is the intersection of two curves of degrees a and b having no common component. It consists of ab points counted with multiplicity.

Proposition 2.4 (Plane complete intersections)

Let $F, G \in \mathbb{K}[x, y, z]$ be homogeneous forms of degrees a and b with no common component. Then the scheme

$$Z = V(F, G) \subset \mathbb{P}^2$$

is zero-dimensional of degree ab . If all intersection points are reduced, then Z is a reduced set of ab points and $I(Z) = (F, G)$.

PROOF. Since F and G have no common component, the ideal (F, G) has height two in the polynomial ring $\mathbb{K}[x, y, z]$. Thus F, G form a homogeneous regular sequence and define a zero-dimensional complete intersection in $\mathbb{P}^2$. The degree of a complete intersection is the product of the degrees of the defining equations, hence $\deg Z = ab$. Equivalently, this is Bézout's theorem with multiplicities. If the intersection points are reduced, the complete-intersection ideal is radical and equals the ideal of the reduced point set. $\square$

Remark 2.5

A finite set of ab points lying on curves of degrees a and b is not automatically a complete intersection of type (a, b) . One needs the scheme-theoretic ideal of the point set to be generated by a regular sequence of those degrees; set-theoretic containment alone may leave residual or embedded structure.

2.4 Intersection multiplicity and Bézout's theorem

We use the classical intersection-theoretic form of Bézout's theorem; see Fulton [15] and Hartshorne [21, Chapter I].

Let $C = V(f)$ and $D = V(g)$ be plane curves in $\mathbb{P}^2$ with no common component. At a point $p \in C \cap D$, the local intersection multiplicity $I_p(C, D)$ measures how strongly the two curves meet at p . It is equal to 1 when the two curves meet transversely, and it is larger when the curves are tangent or when the intersection is otherwise non-transversal.

More formally, provided the intersection is isolated at p , one defines

$$I_p(C, D) = \dim_{\mathbb{K}} \mathcal{O}_{\mathbb{P}^2, p} / (f, g).$$

Theorem 2.6 (Bézout's theorem)

Let $C, D \subset \mathbb{P}^2$ be plane curves over an algebraically closed field $\mathbb{K}$. Assume that C and D have no common component. Then

$$\sum_{p \in C \cap D} I_p(C, D) = \deg(C) \deg(D).$$

In other words, two plane curves of degrees m and n intersect in mn points when the points are counted with their intersection multiplicities. This form of Bézout's theorem will be used repeatedly for intersections involving tangent lines, conics, and cubics.

Lemma 2.7 (Multiplicity of a product)

Let $f = f_1 \cdots f_s$ be a product of local equations at a smooth surface point p . Then

$$\text{mult}_p(f) = \sum_{i=1}^s \text{mult}_p(f_i).$$

In particular, if exactly r distinct lines of an arrangement pass through p , then the product of all line equations has multiplicity r at p .

PROOF. In the regular local ring $\mathcal{O}_{\mathbb{P}^2, p}$, the order of vanishing satisfies

$$\text{ord}_p(fg) = \text{ord}_p(f) + \text{ord}_p(g).$$

A line through p has order one at p , and a line not passing through p has order zero. Summing over the factors gives the claim. $\square$

2.5 Symbolic powers

Symbolic powers are discussed in many sources; throughout this thesis we use the standard commutative-algebra convention from [13, 7].

Definition 2.8 (Symbolic power)

Let $I \subseteq S$ be a homogeneous ideal. The m -th symbolic power is

$$I^{(m)} = S \cap \bigcap_{\mathfrak{p} \in \text{Ass}(S/I)} I^m S_{\mathfrak{p}},$$

where the intersection is taken inside the total quotient ring. If $I = I(Z)$ is the radical ideal of a finite set of points, then

$$I(Z)^{(m)} = \bigcap_{p \in Z} I(p)^m.$$

Thus $F \in I(Z)^{(m)}$ precisely when F vanishes to order at least m at every point of Z .

Lemma 2.9 (Symbolic membership for products of lines)

Let $\mathcal{A}$ be a line arrangement in $\mathbb{P}^2$, let L be the product of all line equations, and let Z be a set of singular points of the arrangement. If at least m lines of $\mathcal{A}$ pass through every point of Z , then

$$L \in I(Z)^{(m)}.$$

PROOF. At a point $p \in Z$, the vanishing order of L equals the number of lines of the arrangement passing through p . By assumption this number is at least m . Hence L belongs to $I(p)^m$ for every $p \in Z$, and therefore to the intersection $\bigcap_{p \in Z} I(p)^m = I(Z)^{(m)}$. $\square$

Remark 2.10

For the full singular locus of an arrangement containing double points, the product of the line equations has multiplicity only two at a double point. Therefore it cannot by itself prove membership in the third symbolic power of the full singular-locus ideal.

2.6 Containment theorems and counterexamples

For every radical ideal one has

$$I^r \subseteq I^{(r)}.$$

The reverse containment is much more difficult. The fundamental uniform theorem is the following; it is one of the main reference points for the containment problem [12, 22, 20].

Theorem 2.11 (Ein–Lazarsfeld–Smith; Hochster–Huneke, [12, 22])

Let R be a regular ring containing a field, and let $I \subset R$ be a radical ideal of big height at most e . Then

$$I^{(er)} \subseteq I^r$$

for every $r \geq 1$.

Remark 2.12

The characteristic-zero proof uses multiplier ideals, while the positive-characteristic proof uses tight closure. The consequence needed in this dissertation is that for a radical ideal of points in $\mathbb{P}^N$, one has

$$I^{(Nr)} \subseteq I^r.$$

The theorem is quoted here as a standard uniform containment theorem.

For points in the plane, the theorem gives $I^{(2r)} \subseteq I^r$. The proposed containment

$$I^{(2r-1)} \subseteq I^r$$

is stronger. When $r = 2$, it becomes $I^{(3)} \subseteq I^2$. The dual Hesse configuration and Yoshinaga’s arrangement show that this strengthening fails in general [10, 29].

2.7 Initial degrees and resurgence

The invariants recalled here are standard in the literature on symbolic powers and point configurations; see [2, 3, 19, 20].

The initial degree of a symbolic power records the least degree of a form with a prescribed uniform multiplicity along the point set:

$$\alpha(I^{(m)}) = \min\{d : I_d^{(m)} \neq 0\}.$$

The Waldschmidt constant is the asymptotic invariant

$$\widehat{\alpha}(I) = \lim_{m \rightarrow \infty} \frac{\alpha(I^{(m)})}{m}.$$

The limit exists because the sequence $\alpha(I^{(m)})$ is subadditive.

Another invariant measuring symbolic-ordinary containment is the resurgence

$$\rho(I) = \sup \left\{ \frac{m}{r} : I^{(m)} \not\subseteq I^r \right\}.$$

The failure of $I^{(3)} \subseteq I^2$ implies $\rho(I) \geq 3/2$ for the relevant ideal. In this thesis, these invariants serve mainly as background for understanding why explicit containment failures are significant.

2.8 Cayley–Bacharach principle

The Cayley–Bacharach theorem is a useful tool for complete intersections of points in the plane; the elementary form used below follows the standard treatment of plane curves and complete intersections in [15, 13, 25].

Theorem 2.13 (Cayley–Bacharach, plane form)

Let $Z = C_a \cap C_b \subset \mathbb{P}^2$ be a reduced complete intersection of curves of degrees a and b . If a curve of degree $a + b - 3$ passes through all but one point of Z , then it passes through the remaining point as well.

PROOF. Write

$$C_a = V(A), \quad C_b = V(B),$$

where A and B are homogeneous forms of degrees a and b . Let

$$Z' = Z \setminus \{p\}$$

and suppose, for contradiction, that there exists a form F of degree $a + b - 3$ such that

$$F|_{Z'} = 0 \quad \text{but} \quad F(p) \neq 0.$$

Choose a line $\ell = V(L)$ through p which contains no other point of Z and which is not tangent to either C_a or C_b at p . Then the form LF vanishes at every point of Z . Since Z is the complete intersection defined by A and B , its ideal is

$$I(Z) = (A, B).$$

Hence

$$LF = AU + BV$$

for some homogeneous forms

$$\deg U = b - 2, \quad \deg V = a - 2.$$

Now restrict this identity to the line ℓ . Since $L = 0$ on ℓ , we obtain

$$0 = \overline{A}\overline{U} + \overline{B}\overline{V}$$

on $\ell \simeq \mathbb{P}^1$, where bars denote restrictions to ℓ .

By the choice of ℓ , the curves C_a and C_b meet ℓ in a common point only at p , and the intersection there is simple. Thus

$$\overline{A} = \lambda A_1, \quad \overline{B} = \lambda B_1,$$

where λ is a linear form on ℓ vanishing at p , and A_1 and B_1 have no common factor. Therefore

$$A_1 \overline{U} + B_1 \overline{V} = 0.$$

Since A_1 and B_1 are coprime, A_1 divides $\overline{V}$. But

$$\deg A_1 = a - 1 \quad \text{and} \quad \deg \overline{V} \leq a - 2,$$

so $\overline{V} = 0$. Similarly, $\overline{U} = 0$.

Thus both U and V vanish identically on the line ℓ , so both are divisible by L . Write

$$U = LU', \quad V = LV'.$$

Then

$$LF = A(LU') + B(LV') = L(AU' + BV').$$

Canceling L , we get

$$F = AU' + BV'.$$

Hence $F \in (A, B) = I(Z)$, so F vanishes at every point of Z , including p . This contradicts $F(p) \neq 0$.

Therefore every curve of degree $a + b - 3$ passing through all but one point of Z must also pass through the remaining point. $\square$

2.9 Minimal free resolutions

For minimal graded free resolutions and Betti numbers we use the conventions of Eisenbud [13].

Let M be a finitely generated graded S -module. A minimal graded free resolution of M has the form

$$0 \leftarrow M \leftarrow \bigoplus_j S(-j)^{\beta_{0,j}} \leftarrow \bigoplus_j S(-j)^{\beta_{1,j}} \leftarrow \cdots.$$

The integers $\beta_{i,j}$ are the graded Betti numbers. They record the degrees of generators and syzygies.

For a finite point set Z , the resolution of $S/I(Z)$ or of $I(Z)$ gives more refined information than the Hilbert function alone. In computations involving special configurations, a change in Betti numbers often signals a special incidence or an unexpected relation among generators.

Definition 2.14 (Arithmetically Cohen–Macaulay scheme)

Let $X \subset \mathbb{P}^N$ be a projective scheme. Write $S = k[x_0, \dots, x_N]$, and let $I_X \subset S$ be its saturated homogeneous ideal. We say that X is *arithmetically Cohen–Macaulay* (ACM) if its homogeneous coordinate ring S/I_X is Cohen–Macaulay, that is, $\text{depth}(S/I_X) = \dim(S/I_X)$.

The determinantal ACM curves used in the Weddle construction below are treated in detail by Chiantini, Farnik, Favacchio, Harbourne, Migliore, Szemberg, and Szpond in [6, Proposition 5.2].

We shall use the following linear case of the Hilbert–Burch theorem [13, 25].

Theorem 2.15 (Hilbert–Burch)

Let M be an $r \times (r+1)$ matrix of linear forms in $S = k[x_0, \dots, x_N]$, and let $I = I_r(M)$ be its ideal of maximal minors. If $\text{ht } I = 2$, then I is perfect and has the graded free resolution

$$0 \longrightarrow S(-r-1)^r \xrightarrow{M^T} S(-r)^{r+1} \longrightarrow I \longrightarrow 0.$$

Consequently S/I is Cohen–Macaulay. In particular, when $N = 3$, the corresponding projective curve is ACM.

2.10 Normal forms and membership tests

The computational membership tests in the thesis are based on the standard Gröbner-basis normal-form criterion, implemented in SINGULAR [8] and described, for example, in [13, 17].

Let G be a Gröbner basis of an ideal $J \subseteq S$ with respect to a fixed monomial order. For a polynomial $f \in S$, the normal form $\text{NF}(f; G)$ is the remainder after division by G .

Lemma 2.16 (Normal-form criterion)

With notation as above,

$$f \in J \iff \text{NF}(f; G) = 0.$$

If $I = (f_1, \dots, f_t)$, then $I \subseteq J$ if and only if $\text{NF}(f_i; G) = 0$ for every generator f_i .

PROOF. The first equivalence is the standard membership criterion for Gröbner bases. For the second statement, the ideal I is contained in J exactly when each generator of I is contained in J . Applying the first criterion to each generator gives the result. $\square$

2.11 Linear systems and projections

The language of linear systems and projections is standard in projective geometry; see [21, 9].

A linear system of degree- d hypersurfaces in $\mathbb{P}^N$ is a projective linear subspace of $\mathbb{P}(S_d)$. Imposing that a hypersurface pass through a point is a linear condition. Imposing a singularity or a prescribed multiplicity at a point gives several linear conditions, described locally by the vanishing of partial derivatives up to the appropriate order.

Let $Z \subset \mathbb{P}^3$ be a finite set and let $p \in \mathbb{P}^3 \setminus Z$. Projection from p is the rational map

$$\pi_p : \mathbb{P}^3 \dashrightarrow \mathbb{P}^2$$

defined by evaluating linear generators of $I(p)$ at points of $\mathbb{P}^3$. Intuitively, one can think of $\mathbb{P}^2$ as a hyperplane H in $\mathbb{P}^3$. Then the image of $q \in \mathbb{P}^3$ under π_p is the intersection of the line pq with H . For p outside a proper closed subset, the projection is injective on Z and preserves the cardinality of Z .

2.12 Grids and geproci sets

Geproci sets were introduced by Chiantini and Migliore [5]; the exceptional $\mathbf{H}_4$ example used later is due to Wiśniewska and Zięba [31].

Definition 2.17 (Geproci set)

A finite set $Z \subset \mathbb{P}^3$ is called geproci if, for a general point $p \in \mathbb{P}^3$, the projection

$$\pi_p : \mathbb{P}^3 \dashrightarrow \mathbb{P}^2$$

maps Z onto a complete intersection in $\mathbb{P}^2$. More precisely, if $\pi_p(Z)$ is a complete intersection of plane curves of degrees a and b , then Z is called an (a, b) -geproci set.

The simplest examples of geproci sets are grids. Let

$$\mathcal{A} = \{L_1, \dots, L_a\}, \quad \mathcal{B} = \{M_1, \dots, M_b\}$$

be two families of lines in $\mathbb{P}^3$ such that the lines in each family are pairwise skew, and such that every line L_i meets every line M_j . Then the set

$$Z = \{L_i \cap M_j : 1 \leq i \leq a, 1 \leq j \leq b\}$$

consists of ab points and is called an (a, b) -grid.

For a general projection $\pi_p : \mathbb{P}^3 \dashrightarrow \mathbb{P}^2$, the lines $L_1, \dots, L_a$ project to a lines in $\mathbb{P}^2$, and the lines $M_1, \dots, M_b$ project to b lines in $\mathbb{P}^2$. Hence the projected set $\pi_p(Z)$ is the intersection of two plane curves,

$$\pi_p(L_1 \cup \dots \cup L_a) \quad \text{and} \quad \pi_p(M_1 \cup \dots \cup M_b),$$

of degrees a and b , respectively. Thus $\pi_p(Z)$ is a complete intersection in $\mathbb{P}^2$, and every grid is geproci.

Following Wiśniewska and Zięba [31], let a and b be positive integers. A set $Z \subset \mathbb{P}^3$ of ab points is called an (a, b) -half-grid if it is covered by a mutually skew lines $L_1, \dots, L_a$ and, under a general projection π_p , the set $\pi_p(Z)$ is the complete intersection of the degree- a plane curve $\pi_p(L_1 \cup \dots \cup L_a)$ with a possibly reducible plane curve of degree b . Thus one defining curve splits into the projected lines, whereas the other need not split into lines.

Geproci sets which are neither grids nor half-grids are of particular interest, because in their case the complete-intersection property of the general projection is not explained by an evident line configuration in $\mathbb{P}^3$. The $\mathbf{H}_4$ configuration is one such example. Its general projection is a complete intersection of type $(6, 10)$, but the configuration is neither a grid nor a half-grid.

2.13 Weddle loci

The classical Weddle surface and related loci go back to Weddle [30]; classical references include Baker [1] and Hudson [23], and a modern perspective can be found in Dolgachev [9].

Let $p_1, \dots, p_s \in \mathbb{P}^3$ be fixed. A surface $\Sigma \subset \mathbb{P}^3$ is called a cone of degree d with vertex v if Σ has degree d and is swept out by lines passing through v . Equivalently, for every point $q \in \Sigma$, $q \neq v$, the line $\overline{vq}$ is contained in Σ . After a projective change of coordinates sending v to $[1 : 0 : 0 : 0]$, such a cone is defined by a homogeneous equation

$$F(x_1, x_2, x_3) = 0$$

of degree d , independent of the coordinate x_0 .

The Weddle problem asks for the locus of points $v \in \mathbb{P}^3$ for which there exists a cone of prescribed degree d with vertex v passing through the given points $p_1, \dots, p_s$. In the classical case of six general points and quadric cones, this locus is the Weddle quartic surface.

Remark 2.18

In the characteristic-zero setting used in Chapter 5, the Weddle scheme is defined by the determinantal ideal of the relevant Macaulay-duality multiplication map. Its expected codimension must be verified before the determinantal degree formula is applied; saturation or radicalization is needed only when one passes from that scheme structure to a different support or residual construction. No positive-characteristic version of the results cited there is asserted in this thesis.

2.14 The group law on a smooth plane cubic

For the group law on smooth plane cubics and its relation to divisors, see Silverman [26] and Fulton [15].

Let $C \subset \mathbb{P}^2$ be a smooth plane cubic and choose a flex point $o \in C$ as the origin. The chord-and-tangent construction gives C the structure of an elliptic curve. If a line meets C at p, q, r , counted with multiplicities, then

$$p + q + r = o.$$

In particular, if the tangent line at p meets C with residual point $\nu(p)$, then

$$2p + \nu(p) = o, \quad \text{hence} \quad \nu(p) = -2p.$$

A plane curve of degree d cuts on C a divisor linearly equivalent to $3d o$. Thus if a degree- d curve has contact divisor $3d p$ with C , then

$$3d(p - o) = o$$

in the group law. This is the basic mechanism behind the torsion descriptions used later.

Definition 2.19 (Flex, sextactic point, and type-9 point)

A point $p \in C$ is a flex if the tangent line has contact order 3 with C at p . It is a sextactic point if there is an irreducible conic having contact order 6 with C at p . A non-flex point $p \in C$ is a type-9 point if there is an irreducible cubic different from C having contact order 9 with C at p .

Proposition 2.20 (Necessary torsion condition for sextactic points)

Let C be a smooth complex plane cubic with origin a flex point. If a conic has intersection divisor $6p$ with C , then $6(p - o) = o$ in the group law.

PROOF. A conic cuts on C a divisor linearly equivalent to $6o$. If its intersection divisor is $6p$, then $6p \sim 6o$ on the elliptic curve, hence $6(p - o) = o$ in the group law. $\square$

Proposition 2.21 (Torsion counts)

For a smooth complex cubic C , write $C[n]$ for its subgroup of n -torsion points. Then

$$C[n] \cong (\mathbb{Z}/n\mathbb{Z})^2.$$

Hence $|C[3]| = 9$, $|C[6]| = 36$, and $|C[9]| = 81$.

PROOF. Analytically, a smooth complex cubic is isomorphic to $\mathbb{C}/\Lambda$ for a lattice Λ . Its n -torsion subgroup is $(\frac{1}{n}\Lambda)/\Lambda$, which is isomorphic to $(\mathbb{Z}/n\mathbb{Z})^2$. $\square$

3 Yoshinaga's Arrangement and the Containment Problem

This chapter is devoted to one of the most instructive examples in the study of containments between symbolic and ordinary powers of ideals of points in the projective plane. We use the notation and containment terminology fixed in Definition 2.1, Definition 2.8, Lemma 2.9, and Theorem 2.11. The example is Yoshinaga's arrangement of 18 lines, treated in detail by Tombarkiewicz and Zięba [29]. Its importance comes from the fact that its singular locus gives explicit counterexamples to the containment

$$I^{(3)} \subseteq I^2,$$

both for the set of triple points and for the full singular locus of the arrangement.

The construction may be viewed as a deformation of the sixth Fermat arrangement. This point of view is useful because the Fermat arrangement has many highly symmetric multiple points, while Yoshinaga's deformation preserves enough incidence structure to produce many triple points but changes the geometry in a way that leads to non-containment. We describe the arrangement explicitly, identify the relevant point configurations, and then verify the failure of containment by exhibiting concrete elements in symbolic cubes which do not belong to the corresponding ordinary squares.

3.1 Background and Motivation

Let $I \subset \mathbb{K}[x, y, z]$ be the radical ideal of a finite set of points in $\mathbb{P}^2$. As recalled in the preliminaries, the ordinary power I^m is generated by products of m elements of I , whereas the symbolic power $I^{(m)}$ consists of forms which vanish to order at least m at every point of the set. Hence one always has

$$I^m \subseteq I^{(m)}.$$

The reverse inclusion is much more delicate and is rarely true in general.

The Ein–Lazarsfeld–Smith and Hochster–Huneke containment theorem implies, in the plane case, that

$$I^{(2r)} \subseteq I^r$$

for all $r \geq 1$. For $r = 2$, this gives $I^{(4)} \subseteq I^2$, but it says nothing about the intermediate containment

$$I^{(3)} \subseteq I^2.$$

The first known counterexample to this containment for points in $\mathbb{P}^2$ is obtained from the dual Hesse arrangement [10]. Yoshinaga's arrangement gives another fundamental counterexample, with different combinatorics and with a very explicit geometric element [29].

The aim of this chapter is therefore twofold. First, we present the arrangement in explicit coordinates. Second, we explain why the relevant forms belong to the symbolic cubes and why they fail to belong to the ordinary squares. The symbolic membership is visible from multiplicities along the arrangement, while the failure of ordinary membership is checked by Gröbner basis computations.

3.2 Yoshinaga's Arrangement

Throughout this chapter we work over $\mathbb{C}$. Let

$$a = e^{2\pi i/6}.$$

Following the standard construction of Fermat-type arrangements used in the containment literature [11, 10, 28, 29], we start from the sixth Fermat, or sixth Ceva, arrangement

$$Q_6(x, y, z) = (x^6 - y^6)(y^6 - z^6)(z^6 - x^6).$$

This is an arrangement of 18 lines in $\mathbb{P}^2$. It has 36 triple points and 3 points of multiplicity 6. Its strong symmetry makes it a natural starting point for constructing line arrangements with many multiple points. Figure 3.1 records this Fermat starting arrangement.

Yoshinaga's arrangement is obtained by deforming a six-line subpencil of Q_6 together with its two cyclic images. Fix $c = 15$, and define

$$P'_1(x, y, z) = (x^3 - y^3)(x + y - cz)(ax + a^5y + cz)(a^5x + ay + cz).$$

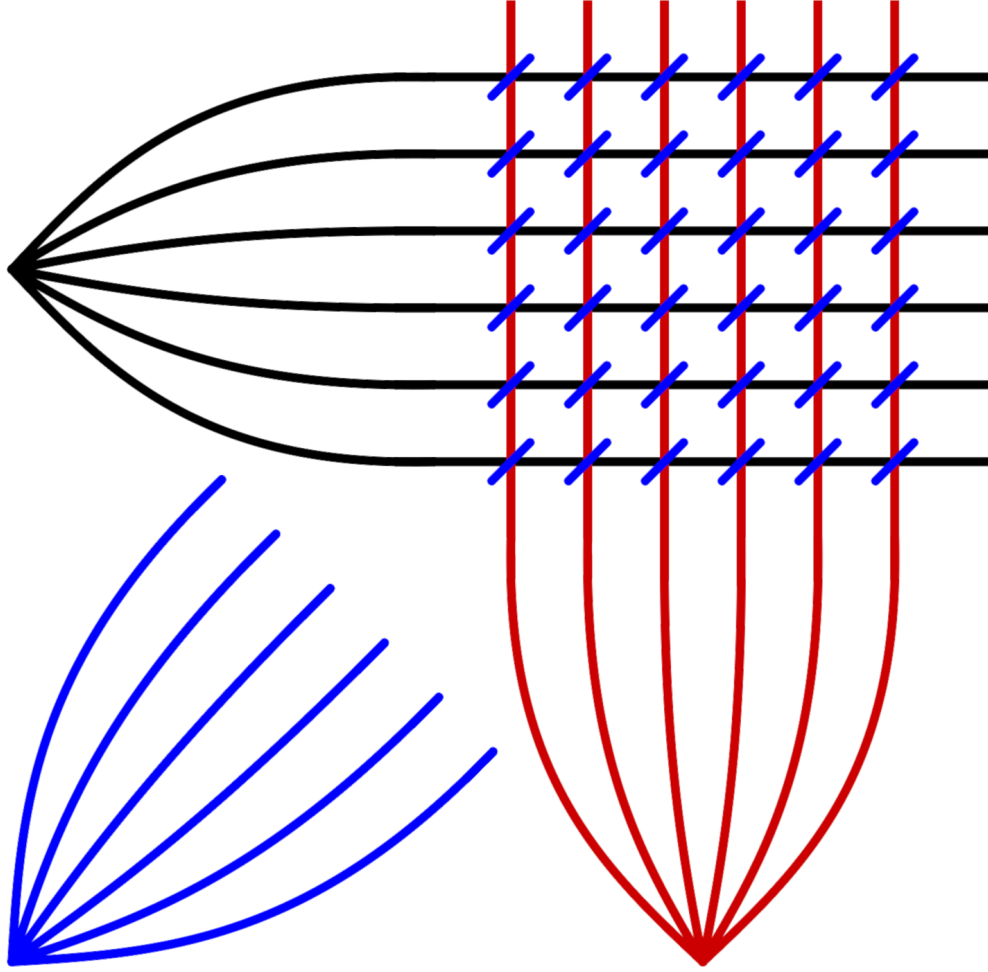


Figure 3.1: Schematic incidence diagram of the Fermat arrangement Q_6 , used as the starting configuration for Yoshinaga's deformation. The curved arcs represent projective lines only to display the incidences clearly; the drawing is not an affine realization of the arrangement.

This polynomial is a product of six linear forms. It is the simplicial arrangement $A_1(6)$, which has 4 triple points and 3 double points. Expanding the product gives

$$P'_1 = x^6 - y^6 + 3cx^4yz - 3cxy^4z - c^3x^3z^3 + c^3y^3z^3.$$

Let

$$\tau(x, y, z) = (y, z, x)$$

be the cyclic permutation of the variables, and put

$$P'_2 = \tau(P'_1), \quad P'_3 = \tau^2(P'_1).$$

The defining equation of Yoshinaga's arrangement is then

$$F = P'_1 P'_2 P'_3.$$

Thus F splits into 18 linear factors. The resulting arrangement has 57 singular points: 48 triple points and 9 double points. Figure 3.2 shows the deformed arrangement used in the theorem below.

3.2.1 The explicit line equations

In the normalization above, the 18 lines are given by the following equations:

$$\begin{array}{ll}
 \ell_1 : x - y = 0 & \ell_{10} : x + 15ay + (a - 1)z = 0 \\
 \ell_2 : x + ay = 0 & \ell_{11} : x + (a - 1)y + 15az = 0 \\
 \ell_3 : x + (-a + 1)y = 0 & \ell_{12} : x - ay + (-15a + 15)z = 0 \\
 \ell_4 : x - 15y + z = 0 & \ell_{13} : x - z = 0 \\
 \ell_5 : x - \frac{1}{15}y - \frac{1}{15}z = 0 & \ell_{14} : x + (-a + 1)z = 0 \\
 \ell_6 : x + y - 15z = 0 & \ell_{15} : x + az = 0 \\
 \ell_7 : x + \frac{a}{15}y + \left(-\frac{a}{15} + \frac{1}{15}\right)z = 0 & \ell_{16} : y - z = 0 \\
 \ell_8 : x + \left(-\frac{a}{15} + \frac{1}{15}\right)y + \frac{a}{15}z = 0 & \ell_{17} : y + (-a + 1)z = 0 \\
 \ell_9 : x + (-15a + 15)y - az = 0 & \ell_{18} : y + az = 0.
 \end{array}$$

We denote by

$$L = \prod_{i=1}^{18} \ell_i$$

the defining equation of the arrangement. Let $\mathcal{P}_3$ be the set of its 48 triple points, and let $\mathcal{P}_2$ be the set of its 9 double points. The full singular locus is

$$\mathcal{P} = \mathcal{P}_3 \cup \mathcal{P}_2.$$

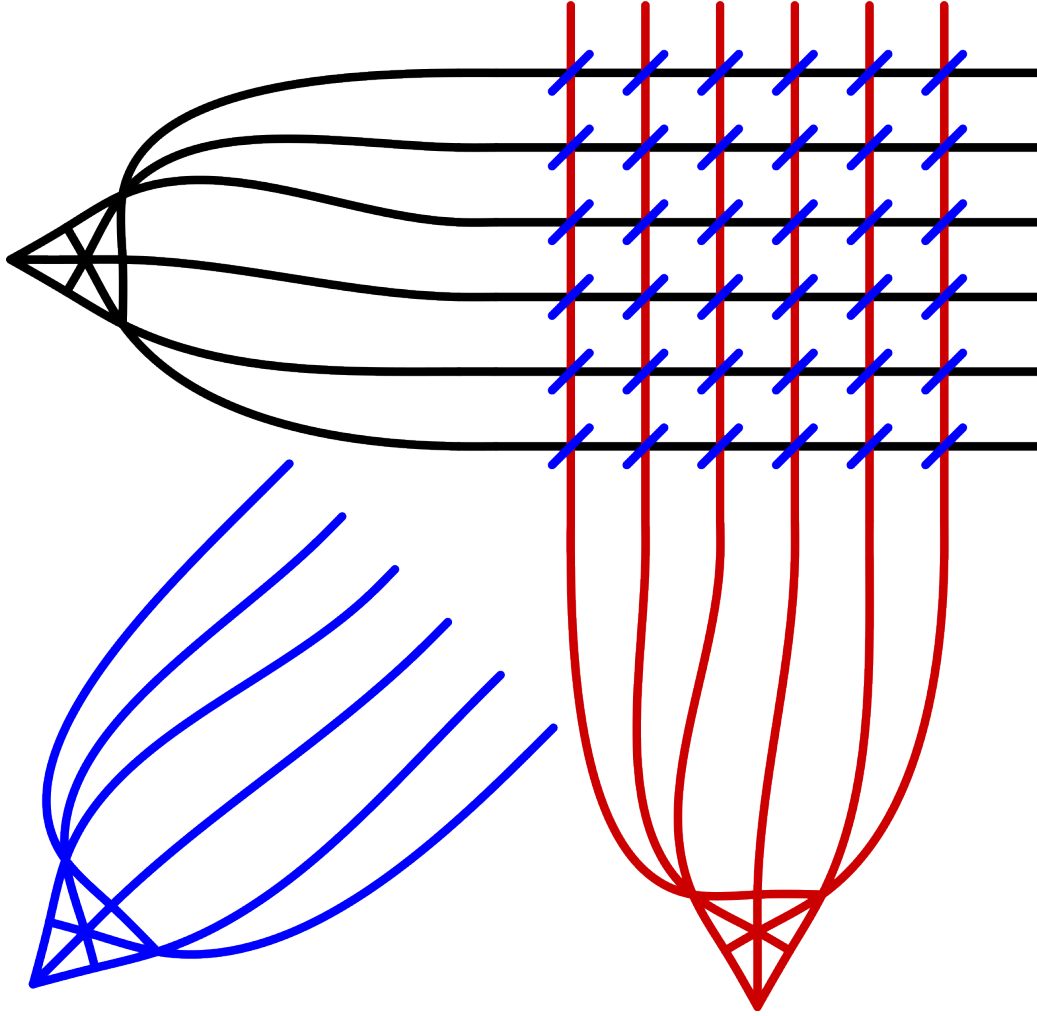


Figure 3.2: Schematic incidence diagram of Yoshinaga's arrangement obtained from a deformation of the sixth Fermat arrangement. The curved arcs represent projective lines only to display the incidences clearly; the drawing is not an affine realization of the arrangement.

3.3 The Non-Containment Statement

We now state the main result of this chapter. The result concerns two ideals: the radical ideal of the triple points alone, and the radical ideal of all singular points.

Theorem 3.1 (Non-containment for Yoshinaga's arrangement, [29])

Let $\mathcal{P}$ be the set of all 57 singular points of Yoshinaga's arrangement, and let $I = I(\mathcal{P})$ be its radical ideal. Then

$$I^{(3)} \not\subseteq I^2.$$

Let $\mathcal{P}_3$ be the set of 48 triple points, and let $I_3 = I(\mathcal{P}_3)$. Then

$$I_3^{(3)} \not\subseteq I_3^2.$$

PROOF. We first consider the set $\mathcal{P}_3$ of triple points. Since

$$L = \prod_{i=1}^{18} \ell_i$$

is the product of all defining linear forms of the arrangement, its multiplicity at a singular point is equal to the number of arrangement lines passing through that point. Hence, for every $p \in \mathcal{P}_3$,

$$\text{mult}_p(L) = 3.$$

Therefore

$$L \in I_3^{(3)}.$$

A direct Gröbner basis computation, described below, gives

$$\text{NF}(L; \text{std}(I_3^2)) \neq 0.$$

Thus $L \notin I_3^2$, and consequently

$$I_3^{(3)} \not\subseteq I_3^2.$$

We now consider the full singular locus $\mathcal{P} = \mathcal{P}_3 \cup \mathcal{P}_2$. At the double points the form L has multiplicity only 2, so L need not belong to $I^{(3)}$. We therefore multiply it by the cubic

$$h = x^3 + y^3 + z^3 - \frac{3379}{225}xyz.$$

This cubic vanishes at the nine double points of the arrangement. Hence, at every double point $q \in \mathcal{P}_2$,

$$\text{mult}_q(Lh) \geq 2 + 1 = 3.$$

At every triple point $p \in \mathcal{P}_3$, the form L already has multiplicity 3, and so

$$\text{mult}_p(Lh) \geq 3.$$

It follows that

$$Lh \in I^{(3)}.$$

A second Gröbner basis computation gives

$$\text{NF}(Lh; \text{std}(I^2)) \neq 0.$$

Therefore $Lh \notin I^2$, and hence

$$I^{(3)} \not\subseteq I^2.$$

□

Figure 3.3 summarizes the two witness elements used in the proof.

$$\begin{array}{ccc}
 \mathcal{P}_3 & \longrightarrow & L = \prod_{i=1}^{18} \ell_i \\
 |\mathcal{P}_3| = 48 & & \\
 & & \xrightarrow{\quad} \begin{array}{l} L \in I(\mathcal{P}_3)^{(3)} \\ L \notin I(\mathcal{P}_3)^2 \end{array}
 \end{array}$$

$$\begin{array}{ccc}
 \mathcal{P} = \mathcal{P}_3 \cup \mathcal{P}_2 & \longrightarrow & Lh \\
 |\mathcal{P}| = 57 & & \\
 & & \xrightarrow{\quad} \begin{array}{l} Lh \in I(\mathcal{P})^{(3)} \\ Lh \notin I(\mathcal{P})^2 \end{array}
 \end{array}$$

Figure 3.3: The two elements used to check non-containment. The line product L proves the failure for the set of triple points, while Lh proves the failure for the full singular locus.

Remark 3.2

The same construction also suggests studying intermediate configurations

$$\mathcal{P}_3 \subseteq Z \subseteq \mathcal{P}.$$

For every such Z , the form Lh belongs to $I(Z)^{(3)}$, since it vanishes to order at least 3 at all triple points and at all double points. However, the failure of containment

$$I(Z)^{(3)} \not\subseteq I(Z)^2$$

does not follow formally from the two extreme cases $Z = \mathcal{P}_3$ and $Z = \mathcal{P}$. It has to be checked separately for each intermediate configuration, or proved by an additional argument. The point-list certificate invoked by Listing 3.1 records only the two extreme configurations used in Theorem 3.1.

3.4 Computational Verification

The exact certificate below has two logically separate tasks. First it verifies that the stated $48 + 9$ points form the complete singular locus of the 18-line arrangement. It then checks the symbolic-power memberships and the failures of membership in the ordinary squares. The latter step is a direct Gröbner basis computation.

We briefly recall the computational principle. Let $J \subset \mathbb{K}[x, y, z]$ be an ideal, and let G be a Gröbner basis of J with respect to a fixed monomial order. For a polynomial f , the normal form

$$\text{NF}(f; G)$$

is the remainder obtained by division of f by G . Then

$$f \in J \iff \text{NF}(f; G) = 0.$$

Thus, to prove that an element f does not belong to J , it is enough to compute a nonzero normal form modulo a Gröbner basis of J .

3.4.1 Completeness of the singular locus

The incidence part of the script first checks that all 57 listed homogeneous triples are nonzero and pairwise distinct as projective points, and that the 18 linear forms define pairwise distinct lines. It then verifies that precisely three lines pass through each of the 48 listed triple points and precisely two lines pass through each of the 9 listed double points.

Finally, the script examines all

$$\binom{18}{2} = 153$$

pairs of lines. Each pair meets in exactly one point of the list, and the number of line pairs represented by each listed point is 3 for a triple point and 1 for a double point. The resulting exact ledger is

$$48 \binom{3}{2} + 9 \binom{2}{2} = 144 + 9 = 153.$$

Every singular point of a line arrangement is the intersection of at least two of its lines. Consequently, exhausting all 153 line pairs proves that no additional singular point is missing from the list.

3.4.2 Point ideals and symbolic powers

The computation starts from explicit homogeneous coordinates of the singular points. For a point

$$p = [p_1 : p_2 : p_3] \in \mathbb{P}^2,$$

we use the ideal

$$I(p) = \langle p_1y - p_2x, p_1z - p_3x, p_2z - p_3y \rangle.$$

This ideal is generated by the 2×2 minors of the matrix

$$\begin{pmatrix} p_1 & p_2 & p_3 \\ x & y & z \end{pmatrix},$$

and hence defines the point $p = [p_1 : p_2 : p_3]$.

Then

$$I_3 = \bigcap_{p \in \mathcal{P}_3} I(p), \quad I = \bigcap_{p \in \mathcal{P}} I(p).$$

Since the ideals are radical ideals of reduced points, their third symbolic powers are computed as

$$I_3^{(3)} = \bigcap_{p \in \mathcal{P}_3} I(p)^3, \quad I^{(3)} = \bigcap_{p \in \mathcal{P}} I(p)^3.$$

This is precisely the form of symbolic powers recalled in the preliminaries.

3.4.3 Normal-form tests

The short, executable driver is reproduced below in Listing 3.1. It loads the exact input and implementation printed in Section A.2.

The composed executable source carries out the following checks:

$$\text{NF}(L; \text{std}(I_3^{(3)})) = 0, \quad \text{NF}(L; \text{std}(I_3^2)) \neq 0,$$

and

$$\text{NF}(Lh; \text{std}(I^{(3)})) = 0, \quad \text{NF}(Lh; \text{std}(I^2)) \neq 0.$$

It also verifies the complete incidence data:

$$|\mathcal{P}_3| = 48, \quad |\mathcal{P}_2| = 9, \quad \binom{18}{2} = 153.$$

Thus the computation proves that the coordinate lists give the entire singular locus and confirms the two containment claims proved above: L is a non-containment witness for the triple-point configuration, and Lh is a non-containment witness for the full singular locus.

3.4.4 Exact incidence and containment certificate driver

Printed dependencies: Listings A.5–A.6.

Listing 3.1: Certificate driver for Yoshinaga's exact incidence and containment checks

```
// Yoshinaga certificate over  $\mathbb{Q}(e)$ .

option(redSB);
ring R=(0,e),(x,y,z),dp;
minpoly=e^2-e+1;

LIB "yoshinaga/yoshinaga_certificate.lib";
execute(read("yoshinaga/data/yoshinaga_input.inc"));

int yoshinaga_ok=0;
yoshinaga_ok=verifyYoshinagaCertificate();
if (yoshinaga_ok!=1) { exit; }

"PASS: exact Yoshinaga incidence and containment certificate";
```

3.5 Conclusion

Yoshinaga's arrangement gives a particularly transparent counterexample to the containment

$$I^{(3)} \subseteq I^2.$$

For the 48 triple points, the non-containment witness is simply the product L of all 18 defining lines. For the full singular locus, one must also account for the 9 double points; this is achieved by multiplying L by the cubic

$$h = x^3 + y^3 + z^3 - \frac{3379}{225}xyz.$$

The symbolic membership follows directly from the vanishing multiplicities at the singular points. The failure of ordinary membership is verified by Gröbner basis normal forms.

This example illustrates the central theme of the containment problem: symbolic powers are governed by local multiplicity conditions, while ordinary powers encode a more rigid global algebraic structure. Yoshinaga's arrangement shows that even for highly structured configurations coming from line arrangements, these two conditions may differ in a substantial way.

4 The H_4 Configuration and General Projections

The purpose of this chapter is to present one of the most remarkable sporadic examples in the theory of geproci point sets. In Chapter 2 we recalled the basic language of finite point configurations in projective space, linear projections, complete intersections, and Bézout’s theorem. We now apply these notions to a highly symmetric configuration of 60 points in $\mathbb{P}^3$, known as the H_4 configuration.

The importance of this example comes from the fact that it is geproci but does not arise from the standard line-based constructions. The known infinite families of geproci sets, such as grids and half-grids, are built from configurations of lines in $\mathbb{P}^3$. The H_4 configuration behaves differently: it is a set of points coming from the H_4 root system, and under a general projection to the plane it becomes a complete intersection of type $(6, 10)$. Thus it shows that the geproci property is not exhausted by grids and half-grids.

The theorem discussed in this chapter was proved by Wiśniewska and the author in [31], building on the framework introduced by Chiantini and Migliore [5]. We present the construction, explain the geometric mechanism behind the geproci property, and give the complete computational checks later in this chapter.

4.1 The H_4 Configuration

Let H_4 denote the non-crystallographic root system of type H_4 . Geometrically, this root system is closely related to the regular 600-cell. The 600-cell has 120 vertices, and after identifying antipodal vertices one obtains a set of 60 points in projective three-space. This is the point configuration denoted by

$$Z_{H_4} \subset \mathbb{P}^3.$$

Equivalently, Z_{H_4} is the projectivization of the H_4 root system. Throughout this chapter we use the numbering from [31]: the projective points are $p_1, \dots, p_{60}$, and the 72 lines containing exactly five configuration points (the 5-reach lines defined below)

are $\ell_1, \dots, \ell_{72}$. Their incidence structure is recalled in Section 4.3. Figure 4.1 gives a schematic view of this passage from roots to projective points.

This configuration has a very large symmetry group. This symmetry is the reason why it is possible to find special surfaces through the points and, after projection, special curves through their image. The resulting behavior is not accidental. It is a manifestation of the strong incidence structure carried by the H_4 root system.

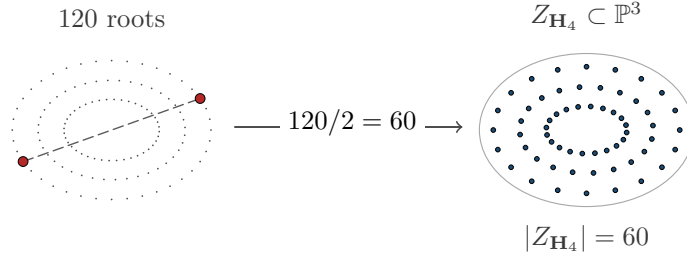


Figure 4.1: Antipodal identification of the 120 H_4 roots, yielding 60 projective points in $\mathbb{P}^3$.

4.2 The Geproci Property

We now recall the main result concerning the configuration $Z_{\mathbf{H}_4}$. In the terminology introduced in Definition 2.17, a finite set $Z \subset \mathbb{P}^3$ is geproci if a general projection of Z to $\mathbb{P}^2$ is a complete intersection.

For the $\mathbf{H}_4$ configuration, the complete intersection type is $(6, 10)$. Since $6 \cdot 10 = 60$, this is numerically compatible with the number of points in $Z_{\mathbf{H}_4}$. The content of the theorem is that for a general projection the projected points are exactly the intersection points of a plane sextic and a plane curve of degree ten.

Theorem 4.1 ($\mathbf{H}_4$ geproci theorem, [31])

Let $Z_{\mathbf{H}_4} \subset \mathbb{P}^3$ be the 60-point configuration obtained from the H_4 root system by projectivization. For a general point $q \in \mathbb{P}^3$, let

$$\pi_q : \mathbb{P}^3 \dashrightarrow \mathbb{P}^2$$

be the projection from q . Then the image $\pi_q(Z_{\mathbf{H}_4})$ is a complete intersection of type $(6, 10)$. In particular,

$$|\pi_q(Z_{\mathbf{H}_4})| = |Z_{\mathbf{H}_4}| = 60 = 6 \cdot 10.$$

Moreover, $Z_{\mathbf{H}_4}$ is neither a grid nor a half-grid.

PROOF. We follow [31, Theorem 3.1]. Let q and π_q be as in the statement. The existence and uniqueness of a degree 6 cone $\mathcal{C}_6$ with vertex q through $Z_{\mathbf{H}_4}$ is proved in [18, Section 3.7]. Wiśniewska and Zięba report a SINGULAR verification that $\mathcal{C}_6$ is smooth away from its vertex. Hence its projection is a smooth plane sextic C_6 containing $\pi_q(Z_{\mathbf{H}_4})$.

For the first half of the configuration, take the $(5, 5)$ -grid defined by

$$(\ell_1, \ell_{25}, \ell_{32}, \ell_{37}, \ell_{44}) \quad \text{and} \quad (\ell_2, \ell_{26}, \ell_{31}, \ell_{38}, \ell_{43}).$$

If g_i , respectively h_i , is the equation of the plane spanned by q and the corresponding line, then $g_1 \cdots g_5$ and $h_1 \cdots h_5$ generate a pencil of degree 5 cones through the 25 grid points. The five further points are $p_4, p_{39}, p_{40}, p_{47}, p_{48}$, all lying on ℓ_{24} . The condition of vanishing at p_4 selects a member of the pencil. The coordinate computation reported in [31, proof of Theorem 3.1] verifies that this member also vanishes at $p_{39}, p_{40}, p_{47}, p_{48}$. Denote the resulting cone by $\mathcal{C}_5$, and the resulting 30-point set by Z_1 . The cited proof records that $\mathcal{C}_5$ is smooth away from q , so its projection C_5 is a smooth, hence irreducible, plane quintic. Moreover,

$$Z_1 \subset \ell_1 \cup \ell_{24} \cup \ell_{25} \cup \ell_{32} \cup \ell_{37} \cup \ell_{44}.$$

Thus $\pi_q(Z_1)$ lies on both C_5 and the union of the six projected lines. The six covering lines are mutually skew, as verified in [31]. These curves have no common component, and Bézout's theorem gives $5 \cdot 6 = 30$; hence they meet exactly in $\pi_q(Z_1)$. In particular, Z_1 is a $(6, 5)$ -half-grid.

For $Z_2 = Z_{\mathbf{H}_4} \setminus Z_1$, the same construction in [31, Theorem 3.1] uses the grid defined by

$$(\ell_7, \ell_{51}, \ell_{60}, \ell_{65}, \ell_{70}) \quad \text{and} \quad (\ell_8, \ell_{54}, \ell_{58}, \ell_{63}, \ell_{71}),$$

together with ℓ_{17} . It yields a second cone $\mathcal{C}'_5$ whose projection is a smooth quintic C'_5 , and

$$Z_2 \subset \ell_7 \cup \ell_{17} \cup \ell_{51} \cup \ell_{60} \cup \ell_{65} \cup \ell_{70}.$$

The six covering lines are again mutually skew [31]. The same Bézout argument shows that Z_2 is also a $(6, 5)$ -half-grid.

The product of the equations of C_5 and C'_5 defines the degree 10 curve

$$C_{10} = C_5 \cup C'_5$$

containing $\pi_q(Z_{\mathbf{H}_4})$. Since the smooth sextic C_6 is irreducible, it has no common component with C_{10} . Therefore, by Theorem 2.6,

$$\text{length}(C_6 \cap C_{10}) = 6 \cdot 10 = 60.$$

For general q , the projection is injective on $Z_{\mathbf{H}_4}$, so $\pi_q(Z_{\mathbf{H}_4})$ consists of 60 distinct points. Consequently,

$$\pi_q(Z_{\mathbf{H}_4}) = C_6 \cap C_{10},$$

and the complete-intersection assertion follows.

Finally, no curve of degree below 6 can contain $\pi_q(Z_{\mathbf{H}_4})$: otherwise its intersection with the irreducible sextic C_6 would contain 60 points, contradicting Bézout. Thus any complete-intersection presentation of these 60 points has both degrees at least 6, and its type must be $(6, 10)$ up to order. If $Z_{\mathbf{H}_4}$ were the corresponding half-grid, it would be covered by mutually skew lines, each containing at least six configuration points. Section 4.3 shows that no line contains more than five. Hence $Z_{\mathbf{H}_4}$ is not a half-grid, and in particular it is not a grid. $\square$

The projection mechanism in the proof is summarized schematically in Figure 4.2.

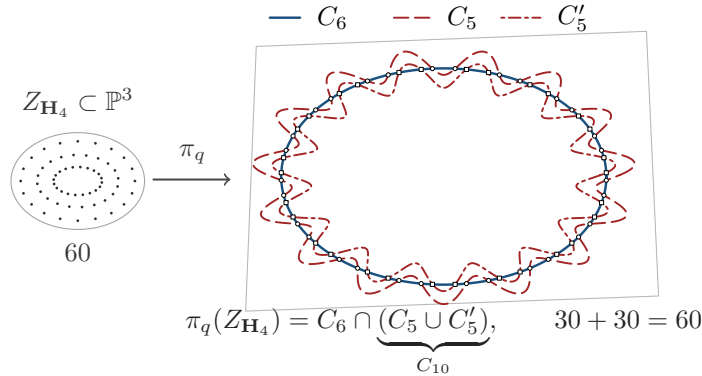


Figure 4.2: The geproci mechanism for the $\mathbf{H}_4$ configuration. A general projection sends the 60 points in $\mathbb{P}^3$ to a plane complete intersection of type $(6, 10)$.

Remark 4.2

Although $Z_{\mathbf{H}_4}$ is not itself a half-grid, the proof shows that it is very close to half-grid geometry. Indeed, it can be written as a union

$$Z_{\mathbf{H}_4} = Z_1 \cup Z_2$$

of two disjoint 30-point subsets, each of which is a half-grid. Each subset is obtained from a $(5, 5)$ -grid by adding five further points lying on one additional 5-reach line. Thus each Z_i is a $(6, 5)$ -half-grid: it is covered by six mutually skew 5-reach lines, and its general projection is a complete intersection of type $(6, 5)$.

Consequently, the full configuration is not line-free. Rather, it is built from many highly organized collinear subsets, but in a way that does not allow the whole set of 60 points to be covered by lines with six or more points. This is precisely what makes the example more subtle than grids and half-grids.

4.3 Incidence Structure

The $\mathbf{H}_4$ configuration also has a natural incidence structure coming from the bilinear form associated with the H_4 root system. This structure is useful both conceptually and computationally, and it is also one of the main ingredients in the proof of Theorem 4.1.

Let

$$\varphi = \frac{1 + \sqrt{5}}{2}$$

be the golden ratio, so that

$$\varphi^2 - \varphi - 1 = 0.$$

The coordinates of the H_4 roots may be written over the field $\mathbb{Q}(\varphi)$. If

$$p_i = [p_{i0} : p_{i1} : p_{i2} : p_{i3}] \in Z_{\mathbf{H}_4},$$

we associate to p_i the plane

$$V_i = \{[X_0 : X_1 : X_2 : X_3] \in \mathbb{P}^3 \mid \langle p_i, X \rangle = 0\},$$

where $\langle \cdot, \cdot \rangle$ denotes the standard bilinear form in the chosen coordinates. Thus V_i is the projective plane orthogonal to p_i .

The incidence relation is encoded by the simple rule

$$p_j \in V_i \iff \langle p_i, p_j \rangle = 0.$$

For the full $\mathbf{H}_4$ configuration this gives a highly symmetric incidence pattern: each point is incident with 15 of the corresponding planes, and each plane contains 15 of the points. In this sense the configuration has a symmetric (60_{15}) -structure. Figure 4.3 records this point-plane incidence as a compact ledger.

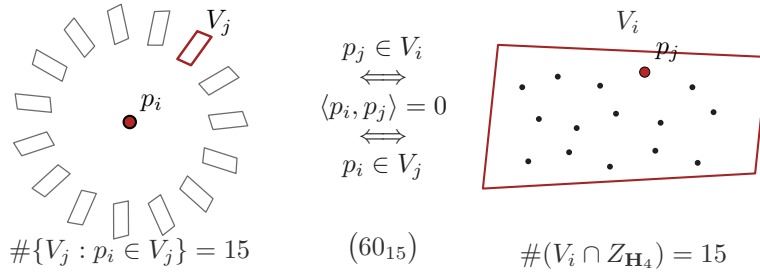


Figure 4.3: Incidence schematic for the $\mathbf{H}_4$ configuration. Orthogonality gives the point-plane incidence relation and leads to the symmetric (60_{15}) structure.

In addition to these point-plane incidences, the configuration determines many special lines. Following [31], we call a line a 5-reach line if it contains exactly five points of $Z_{\mathbf{H}_4}$. The configuration contains 72 such lines. Moreover, no line contains more than five configuration points, and through each point of $Z_{\mathbf{H}_4}$ there pass exactly six 5-reach lines.

These 5-reach lines are crucial in the proof of the geproci property. They show that, although $Z_{\mathbf{H}_4}$ is not a half-grid, it still contains large and highly organized collinear sub-configurations. In the incidence tables and unordered pairs below, only the subscripts of the points p_i are displayed.

The first 30-point subset

The first half-grid used in the proof is obtained from the following 5-reach lines:

$$\ell_1, \ell_{24}, \ell_{25}, \ell_{32}, \ell_{37}, \ell_{44}.$$

Their incidences with the points of $Z_{\mathbf{H}_4}$ are:

line	point indices
ℓ_1	1, 29, 32, 33, 36
ℓ_{24}	4, 39, 40, 47, 48
ℓ_{25}	5, 13, 31, 41, 52
ℓ_{32}	6, 15, 34, 38, 58
ℓ_{37}	7, 14, 35, 37, 57
ℓ_{44}	8, 16, 30, 42, 51.

Thus their union contains the following 30 points:

$$\begin{aligned} Z_1 = \{ & p_1, p_4, p_5, p_6, p_7, p_8, p_{13}, p_{14}, p_{15}, p_{16}, \\ & p_{29}, p_{30}, p_{31}, p_{32}, p_{33}, p_{34}, p_{35}, p_{36}, p_{37}, p_{38}, \\ & p_{39}, p_{40}, p_{41}, p_{42}, p_{47}, p_{48}, p_{51}, p_{52}, p_{57}, p_{58} \}. \end{aligned}$$

Inside Z_1 there is a $(5, 5)$ -grid. One family of lines is

$$\ell_1, \ell_{25}, \ell_{32}, \ell_{37}, \ell_{44},$$

and the other family is

$$\ell_2, \ell_{26}, \ell_{31}, \ell_{38}, \ell_{43}.$$

The incidences of the second family are:

line	point indices
ℓ_2	1, 30, 31, 34, 35
ℓ_{26}	5, 16, 33, 37, 58
ℓ_{31}	6, 14, 32, 42, 52
ℓ_{38}	7, 15, 29, 41, 51
ℓ_{43}	8, 13, 36, 38, 57.

These two families form a $(5, 5)$ -grid of 25 points. The remaining five points of Z_1 lie on the line ℓ_{24} .

The role of the line ℓ_{24} can also be described through the following incidence. The point p_4 determines ten lines meeting the quadric containing the $(5, 5)$ -grid in pairs of grid points:

$$\begin{aligned} & \{5, 6\}, \{7, 8\}, \{13, 14\}, \{15, 16\}, \{29, 30\}, \\ & \{31, 32\}, \{33, 34\}, \{35, 36\}, \{37, 38\}, \{41, 42\}. \end{aligned}$$

The same property is shared by the points $p_{39}, p_{40}, p_{47}, p_{48}$, and these five points lie on ℓ_{24} .

The second 30-point subset

The complementary subset

$$Z_2 = Z_{\mathbf{H}_4} \setminus Z_1$$

has the same type of structure. It is covered by the six 5-reach lines

$$\ell_7, \ell_{17}, \ell_{51}, \ell_{60}, \ell_{65}, \ell_{70}.$$

Their incidences are:

line	point indices
ℓ_7	2, 17, 20, 21, 24
ℓ_{17}	3, 49, 50, 53, 54
ℓ_{51}	9, 19, 25, 45, 59
ℓ_{60}	10, 23, 27, 43, 55
ℓ_{65}	11, 22, 26, 44, 56
ℓ_{70}	12, 18, 28, 46, 60.

Hence

$$Z_2 = \{p_2, p_3, p_9, p_{10}, p_{11}, p_{12}, p_{17}, p_{18}, p_{19}, p_{20}, \\ p_{21}, p_{22}, p_{23}, p_{24}, p_{25}, p_{26}, p_{27}, p_{28}, p_{43}, p_{44}, \\ p_{45}, p_{46}, p_{49}, p_{50}, p_{53}, p_{54}, p_{55}, p_{56}, p_{59}, p_{60}\}.$$

Again, Z_2 contains a $(5, 5)$ -grid. One family of lines is

$$\ell_7, \ell_{51}, \ell_{60}, \ell_{65}, \ell_{70},$$

and the second family is

$$\ell_8, \ell_{54}, \ell_{58}, \ell_{63}, \ell_{71}.$$

The incidences of the second family are:

line	point indices
ℓ_8	2, 18, 19, 22, 23
ℓ_{54}	9, 24, 28, 44, 55
ℓ_{58}	10, 20, 26, 46, 59
ℓ_{63}	11, 17, 27, 45, 60
ℓ_{71}	12, 21, 25, 43, 56

The remaining five points of Z_2 lie on the line ℓ_{17} .

These explicit incidences explain the geometric mechanism used in the proof of Theorem 4.1. Each of the sets Z_1 and Z_2 contains a $(5, 5)$ -grid and five additional collinear points. Together with the quintic-cone construction in the proof, these incidences show that each subset is a $(6, 5)$ -half-grid: after projection from a general point, it is a complete intersection of type $(6, 5)$. Equivalently, each subset is contained in a cone of degree 5 with vertex at the center of projection.

Taking the union of the two degree 5 cones gives a cone of degree 10 through the whole configuration $Z_{\mathbf{H}_4}$. This is the degree 10 cone used in the proof of the $(6, 10)$ -geproci property.

At the same time, the incidence table also explains why $Z_{\mathbf{H}_4}$ is not a half-grid. The irreducible sextic and Bézout exclude a plane curve of degree below 6 through all 60

projected points, so any complete-intersection type has both degrees at least 6, hence is $(6, 10)$ up to order. Such a half-grid structure would require covering lines containing at least six configuration points. The incidence computation shows that the maximum number of collinear points in $Z_{\mathbf{H}_4}$ is 5. Thus $Z_{\mathbf{H}_4}$ is not a half-grid, even though it is the union of two 30-point half-grids.

4.4 Computational Checks

The proof of Theorem 4.1 is geometric and follows [31]. The accompanying Singular scripts also give a theorem-level certificate on the complete published configuration. The first script checks all 60 points, the symmetric (60_{15}) incidence pattern, all 72 five-point lines, and two disjoint 30-point covers by six mutually skew lines. The geometric proof identifies the covered subsets as half-grids.

The configuration itself is fixed: the calculation concerns the published 60-point set $Z_{\mathbf{H}_4}$, not a general configuration of 60 points. What varies is the center of projection. To represent its generic point, retain the homogeneous parameters and write

$$q = [q_0 : q_1 : q_2 : q_3]$$

over the rational function field $\mathbb{Q}(q_0, q_1, q_2, q_3)$, with the golden ratio adjoined. Thus the q_i are algebraically independent in the symbolic part of the calculation; they are not assigned numerical values. On the dense chart $q_3 \neq 0$, one may equivalently normalize $q_3 = 1$. After clearing denominators, the nonvanishing conditions define the complement of a proper closed subset of this chart. Consequently, the identities and rank statements obtained over the function field hold on a nonempty Zariski-open set of projection centers, which is precisely the meaning of a general projection.

Over this function field, the script checks all $\binom{60}{2} = 1770$ projective pairs, so the projected points are generically distinct. It then computes the 24-dimensional space $I(Z_{\mathbf{H}_4})_6$: the degree-six evaluation matrix has 84 columns and rank 60, hence its kernel has dimension $84 - 60 = 24$. A member is a cone with vertex q precisely when its 56 fifth partial derivatives vanish at q . Indeed, for a homogeneous form of degree 6 in characteristic zero, Euler's identities propagate the vanishing of all derivatives of order 5 to every lower order, which is equivalent to multiplicity 6 at q . Following the general-point matrix method of [18], the unique kernel vector is obtained from signed maximal minors of a 23×24 matrix and is verified against all 56 conditions and all 60 points. Restriction to the target plane identifies these degree-six cones linearly with the plane sextics through $\pi_q(Z_{\mathbf{H}_4})$: conversely, the cone from q over any such plane sextic belongs to the same kernel. Thus the one-dimensional kernel computed below is exactly $I(\pi_q(Z_{\mathbf{H}_4}))_6$, not merely a subspace of it.

The decic is the product of the two quintics belonging to the two half-grids, and coprimality with the sextic is proved symbolically by the quadratic-field norm calculation. More explicitly, a common nonconstant factor over $\mathbb{Q}(q_0, q_1, q_2, q_3)(\varphi)$ would have a nonconstant field norm dividing the norms of both curves. Therefore the verified equality

$$\gcd(N(f_6), N(f_{10})) = 1$$

implies $\gcd(f_6, f_{10}) = 1$.

The rank calculation also takes place entirely over $\mathbb{Q}(q_0, q_1, q_2, q_3)(\varphi)$. A nonzero 23×23 minor of the fifth-derivative matrix proves that its rank is at least 23. The nonzero vector of signed maximal minors is then checked against all 56 rows, so it lies in the kernel and proves that the rank is at most 23. Thus the rank is exactly 23, its kernel is one-dimensional, and $\dim I(\pi_q(Z_{\mathbf{H}_4}))_6 = 1$. Together with the symbolic injectivity and coprimality checks, this gives 60 distinct common zeros of relatively prime curves of degrees 6 and 10; Bézout then gives the asserted complete intersection.

More precisely, the short drivers invoke the exact certificate entry points, while their implementations and input data are printed in Section A.3:

- Listing 4.1 invokes the incidence and six-line-cover certificate;
- Listing 4.2 invokes the exact generic $(6, 10)$ -geproci certificate.

4.4.1 SINGULAR certificate drivers

Printed dependencies: Listings A.7, A.9, and A.10.

Listing 4.1: Incidence and six-line-cover driver for the 60-point $\mathbf{H}_4$ configuration

```
// H4 configuration certificate.

ring R = (0,u),(x0,x1,x2,x3),dp;
minpoly = u2-u-1;
execute(read("h4/data/h4_points.inc"));
execute("list L="+H4DATA+";");
execute(read("h4/data/h4_half_grids.inc"));
int configuration_ok=0;
LIB "h4/h4_configuration.lib";

verifyH4Configuration();
if (configuration_ok!=1) { exit; }
"PASS: exact H4 configuration";
exit;
```

Printed dependencies: Listings A.8–A.10.

Listing 4.2: Certificate driver for the exact generic $(6, 10)$ projection of $\mathbf{H}_4$

```
// Generic H4 projection certificate.

execute(read("h4/data/h4_points.inc"));
ring Rs = (0,u),(x0,x1,x2,x3),dp;
minpoly = u2-u-1;
execute("list L="+H4DATA+";");
int generic_ok=0;

execute(read("h4/h4_generic_projection.inc"));
verifyGenericProjection();
if (generic_ok!=1) { exit; }
"PASS: exact generic (6,10)-geproci certificate";
exit;
```

4.5 Conclusion

The $\mathbf{H}_4$ configuration shows that the geproci property is richer than the known grid and half-grid constructions. Although it is the disjoint union of two 30-point half-grids, the full 60-point configuration is neither a grid nor a half-grid. Its general projection to $\mathbb{P}^2$ is a complete intersection of type $(6, 10)$.

From the point of view of Chapter 2, the main ingredients are classical: linear projection, complete intersections, and Bézout’s theorem. What is special here is the way these ingredients interact with the symmetry of the H_4 root system. The example is therefore both concrete and exceptional. It provides strong evidence that any classification of geproci point sets must include genuinely sporadic configurations, not only the infinite families coming from grids and half-grids.

5 The Weddle Surface and Analogous Loci

The definitions of Weddle loci used in this chapter refine the preliminary discussion in Chapter 2. The classical quartic is fixed in Definition 5.1, while the higher-degree analog is Definition 5.4.

Throughout this chapter, the general geometric statements taken from [6] are understood over $\mathbb{C}$, as in that source. The exact specialization in Listing 5.1 is computed over $\mathbb{Q}$. Its matrix ranks, codimensions, Hilbert series, and degrees are unchanged after the flat extension of scalars $\mathbb{Q} \subset \mathbb{C}$. No assertion in positive characteristic is made.

In the previous chapter we studied the $\mathbf{H}_4$ configuration of 60 points in $\mathbb{P}^3$. A crucial step in the proof of its geproci property was the existence of a cone of degree 6 with prescribed general vertex passing through all points of the configuration. After projection from this vertex, the cone becomes a plane sextic containing the projected point set. Together with a second curve of degree 10, this gives the complete intersection of type $(6, 10)$.

This observation places the $\mathbf{H}_4$ example in a broader and much older circle of questions. Instead of fixing the vertex and asking whether a cone through a given point set exists, one may fix the point set and ask for all possible vertices of such cones. More precisely, given points

$$p_1, \dots, p_s \in \mathbb{P}^3,$$

we ask for the locus of points $v \in \mathbb{P}^3$ such that there exists a cone of prescribed degree with vertex v passing through $p_1, \dots, p_s$.

This point of view goes back to classical projective geometry. In the first non-trivial case, one considers quadric cones through six general points in $\mathbb{P}^3$. The locus of their vertices is a quartic surface, the classical *Weddle surface* introduced by Weddle [30] and later studied by Baker [1], Hudson [23], and many others. Thus the Weddle surface may be regarded as the classical prototype of a vertex locus for cones through a finite point configuration.

The aim of this chapter is to explain this construction and then to pass from the classical Weddle surface to higher-degree analogs. We begin with quadric cones through six points, then discuss cubic cones through ten points, and finally indicate the general pattern for cones of degree n . In this way the chapter connects the unexpected-cone

phenomenon used in the $\mathbf{H}_4$ construction with the nineteenth-century problem of determining vertices of cones through prescribed point sets.

The aim is not to reproduce the full classical theory, but rather to isolate the features that are relevant for the present thesis: the degree of the vertex locus, its singularities at the base points, the lines contained in it, and the curves obtained when one imposes one additional base point.

5.1 The Classical Weddle Surface

The classical Weddle construction answers this question for the smallest case in which the condition is non-trivial. A plane conic depends on six coefficients up to scale, so six projected points impose exactly the expected number of conditions. Therefore, for a variable projection center v , the condition that the six projected points lie on a conic cuts out a surface in the space of possible vertices.

Let

$$\mathcal{P} = \{p_1, \dots, p_6\} \subset \mathbb{P}^3$$

be a set of six points in general position. We ask for the locus of points $v \in \mathbb{P}^3$ such that there exists a quadric cone with vertex v passing through the six points of $\mathcal{P}$.

Equivalently, if we project the points $p_1, \dots, p_6$ from v onto a plane, then v is a vertex of such a cone precisely when the six projected points lie on a plane conic. Since six general points in the plane do not lie on a conic, this condition imposes a non-trivial equation on v . The resulting locus is the classical Weddle surface.

Definition 5.1 (Classical Weddle surface)

Let $\mathcal{P} = \{p_1, \dots, p_6\} \subset \mathbb{P}^3$ be six general points. The *Weddle surface* associated with $\mathcal{P}$ is the closure of the locus of all points $v \in \mathbb{P}^3$ for which there exists a quadric cone with vertex v containing $\mathcal{P}$.

The Weddle surface is a surface of degree 4. We denote it by W_4 . Its basic geometry is very rigid and is governed by the six base points.

Theorem 5.2 (Classical properties of the Weddle surface, [1, 9])

Let $W_4 \subset \mathbb{P}^3$ be the Weddle surface determined by six general points $p_1, \dots, p_6$. Then:

1. W_4 is a quartic surface.
2. Each base point p_i is an ordinary double point of W_4 .
3. The 15 lines $\overline{p_i p_j}$, $1 \leq i < j \leq 6$, are contained in W_4 .
4. Through each node p_i pass exactly five of these lines.

SKETCH OF THE CLASSICAL ARGUMENT. The construction may be described by projection from a variable point v . After projecting the six base points from v to a plane, the condition that their images lie on a conic is expressed by a determinant. In the Macaulay-duality description, multiplication between the relevant four-dimensional graded pieces is represented by a 4×4 matrix of linear forms. Its determinant has degree 4 and defines the quartic Weddle scheme.

The singularity at a base point is explained as follows. If v approaches one of the base points, then all cones with vertex v and passing through the remaining points acquire a distinguished limiting behavior. In local coordinates this gives multiplicity 2 at each base point. Moreover, if v lies on the line $\overline{p_i p_j}$, then the line through v and the two base points is contained in every cone with vertex v passing through p_i and p_j . This shows that each join $\overline{p_i p_j}$ is contained in the vertex locus. The full statements are classical and can be found, for instance, in Baker [1] and Dolgachev [9]. $\square$

Figure 5.1 separates the surface parametrizing the admissible vertices from the particular quadric cone selected by one such vertex.

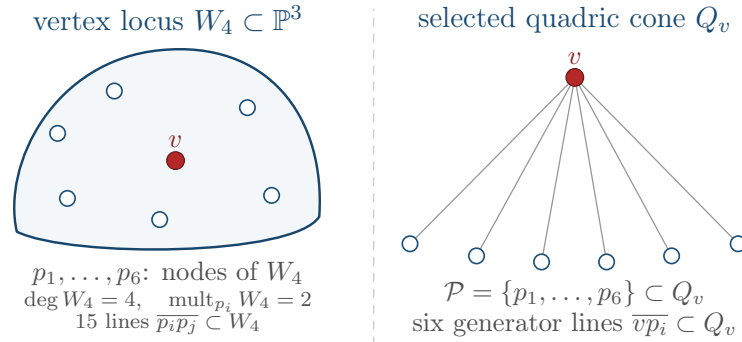


Figure 5.1: Two-panel schematic of the classical Weddle construction. The left panel represents the vertex-locus surface W_4 , containing the six fixed base points as nodes and a chosen point v . The right panel represents the corresponding quadric cone Q_v through the same points; its gray segments are the generator lines $\overline{vp_i}$. The surface patch and all point positions are purely diagrammatic and assert no affine realization or collinearity. In particular, Q_v and W_4 are distinct objects, and the 15 lines $\overline{p_i p_j} \subset W_4$ are not the displayed generators.

There is another classical feature which will be useful later. If one adds a seventh general point, the condition becomes stronger: one now looks for vertices of quadric cones through seven points. The locus is no longer a surface but a curve. In the classical case this curve has degree 6 and genus 3, as described by Baker [1]. Equivalently, each six-element subset of the seven points determines a Weddle quartic, and the curve of vertices is obtained as part of the common locus of these quartics.

5.2 Cubic Cones and the Surface W_{10}

We now pass from quadric cones to cubic cones. This is the first case where the degree of the Weddle-type surface is substantially larger, and it already displays the pattern that appears in arbitrary degree.

Let

$$\mathcal{P} = \{p_1, \dots, p_{10}\} \subset \mathbb{P}^3$$

be ten general points. A cubic cone in $\mathbb{P}^3$ with vertex v is obtained by taking a plane cubic after projection from v . Thus v is a vertex of a cubic cone through $\mathcal{P}$ precisely when the ten projected points lie on a plane cubic.

The vector space of plane cubics has dimension

$$\binom{3+2}{2} = 10.$$

Thus ten projected points impose ten linear conditions on plane cubics. The condition that a nonzero cubic passes through all of them is the vanishing of a determinant. The corresponding locus of vertices is a surface, denoted here by W_{10} .

Theorem 5.3 (Cubic Weddle surface, [14])

Let $\mathcal{P} = \{p_1, \dots, p_{10}\} \subset \mathbb{P}^3$ be ten general points. The locus of vertices of cubic cones through $\mathcal{P}$ is a surface $W_{10} \subset \mathbb{P}^3$ of degree 10. Moreover:

1. *each base point p_i is a triple point of W_{10} ;*
2. *every line $\overline{p_i p_j}$, $1 \leq i < j \leq 10$, is contained in W_{10} ;*
3. *hence W_{10} contains the 45 joining lines determined by the ten base points.*

IDEA OF PROOF. The multiplication-matrix construction below realizes W_{10} as the determinant of a 10×10 matrix of linear forms, hence as a degree-10 surface. The triple points and joining lines are part of Emch's classical analysis [14]. $\square$

Figure 5.2 makes the same distinction between the cubic vertex locus and the cone determined by a selected vertex.

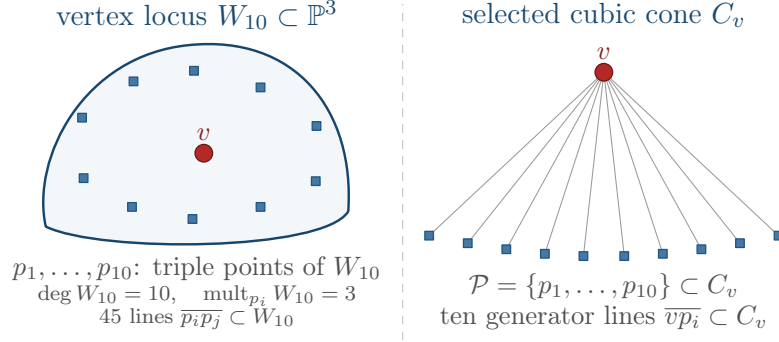


Figure 5.2: Two-panel schematic of the cubic Weddle construction. The left panel represents the vertex-locus surface W_{10} , containing the ten fixed base points as triple points and a chosen point v . The right panel represents the corresponding cubic cone C_v through the same points; its gray segments are the generator lines $\overline{vp_i}$. The surface patch and all point positions are purely diagrammatic and assert no affine realization or collinearity. In particular, C_v and W_{10} are distinct objects, and the 45 lines $\overline{p_i p_j} \subset W_{10}$ are not the displayed generators.

5.2.1 Determinantal description

The Weddle scheme admits a direct determinantal description. For the general geometric statements let $k = \mathbb{C}$. For a rational point configuration the same multiplication matrix is formed over $k = \mathbb{Q}$ and then extended to $\mathbb{C}$. Let

$$p_i = [a_i : b_i : c_i : d_i], \quad L_i = a_i x_0 + b_i x_1 + c_i x_2 + d_i x_3,$$

and put

$$A_r = k[x_0, x_1, x_2, x_3] / (L_1^3, \dots, L_r^3).$$

For a variable point $q = [q_0 : q_1 : q_2 : q_3]$, write $L_q = q_0 x_0 + q_1 x_1 + q_2 x_2 + q_3 x_3$. Macaulay duality identifies the cubic Weddle scheme with the degeneracy locus of

$$[A_r]_2 \xrightarrow{\cdot L_q} [A_r]_3.$$

This is the Macaulay-duality description of the Weddle scheme [6]. For ten general points both graded pieces have dimension 10. Consequently the determinant of the resulting 10×10 matrix of linear forms defines W_{10} and has degree 10, in agreement with

$$m = \binom{n+2}{3} = \frac{n(n+1)(n+2)}{6}$$

for $n = 3$. The corresponding exact computation is invoked by Listing 5.1; its implementation and rational input are printed in Section A.4.

Figure 5.3 summarizes the multiplication-matrix construction.

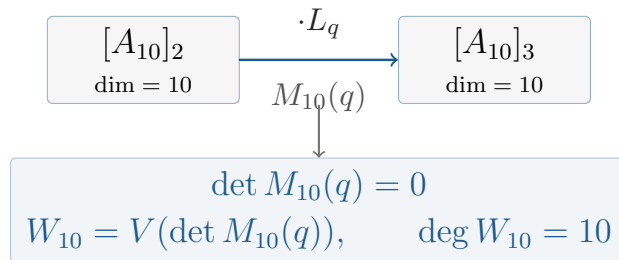


Figure 5.3: Multiplication-matrix construction of the cubic Weddle surface: the determinant of the 10×10 matrix $M_{10}(q)$ has degree 10.

5.3 The Curve of Vertices for Eleven Points

We next impose one additional base point. Let

$$\mathcal{P} = \{p_1, \dots, p_{11}\} \subset \mathbb{P}^3$$

be eleven general points. We ask for the locus of vertices of cubic cones through all eleven points.

For ten points the vertex locus was the surface W_{10} . Adding an eleventh point cuts this locus down. More precisely, if we choose two different ten-element subsets of $\mathcal{P}$, each subset determines a cubic Weddle surface of degree 10. Their intersection contains the locus of vertices of cubic cones through all eleven points.

Let $W_{10}^{(a)}$ and $W_{10}^{(b)}$ be two such Weddle surfaces obtained from two ten-element subsets sharing nine base points. For general $\mathcal{P}$, the situation covered by [6, Theorem 5.1] ensures that the two surfaces have no common surface component. Bézout's theorem therefore gives their complete intersection total degree

$$10 \cdot 10 = 100.$$

The modern determinantal description separates two curve schemes [6, Propositions 5.2 and 5.3]:

1. the 3-Weddle scheme Γ_{45} of all eleven points, an arithmetically Cohen–Macaulay curve cut out by the maximal minors of a 9×10 matrix of linear forms and having degree 45;
2. the 3-Weddle scheme Σ_{55} of the nine common points, which has degree 55 and contains the $36 = \binom{9}{2}$ reduced joining lines.

The ACM property and the degree of Γ_{45} can also be read directly from Hilbert–Burch. The maximal-minor ideal has height 2, so Theorem 2.15 gives

$$0 \longrightarrow R(-10)^9 \longrightarrow R(-9)^{10} \longrightarrow I_{\Gamma_{45}} \longrightarrow 0, \quad R = k[q_0, q_1, q_2, q_3].$$

Hence

$$\mathrm{HS}_{R/I_{\Gamma_{45}}}(t) = \frac{1 - 10t^9 + 9t^{10}}{(1 - t)^4}.$$

The numerator has a double zero at $t = 1$, and one half of its second derivative at 1 is 45. This proves that the projective ACM curve has degree 45.

For later reference, write L_{36} for the reduced union of the $\binom{9}{2} = 36$ joining lines occurring in the general nine-point scheme, and write R_{19} for the degree-19 residual scheme obtained in the rational specialization used below. Thus Γ_{45} and Σ_{55} are the two determinantal curve schemes, whereas L_{36} and R_{19} describe the internal degree ledger of Σ_{55} .

Four logically distinct inputs are being used here. Proposition 5.2 of [6] proves the general ACM statement and degree 45 for Γ_{45} . Proposition 5.3, together with the cubic expected-codimension verification recorded in Remark 5.5, gives degree 55 for Σ_{55} , which contains the reduced secant scheme L_{36} . The passage to general points uses the semicontinuity argument recorded there; it is not inferred from the specialization below. The matrices and ideals in that computation are defined over $\mathbb{Q}$; their ranks, Krull dimensions, Hilbert series, and degrees are unchanged after extension of scalars to $\mathbb{C}$. Finally, Listing 5.1 invokes an exact calculation over $\mathbb{Q}$ for one explicitly listed configuration: it verifies the two determinantal degrees, their containments in two coprime degree-10 surfaces, and a specialized residual degree 19. It does not by itself prove generic radicality or irreducibility of the residual scheme.

At the level of one-dimensional cycles, the complete degree accounting is

$$100 = 45 + 55 = 45 + 36 + 19.$$

Emch’s earlier value 52 for the eleven-point curve is incorrect [14, 6]. No equality stronger than the stated cycle and ideal calculations, and no generic irreducibility assertion about R_{19} , is needed. Figure 5.4 records the corrected Bézout ledger.

5.3.1 Exact rational SINGULAR specialization driver

Printed dependencies: Listings A.1, A.11, and A.12.

Listing 5.1: Certificate driver for the exact rational cubic Weddle specialization

```
// Rational cubic Weddle specialization.
option(redSB);

LIB "library/certificate_utils.lib";
<"weddle/data/cubic_weddle_points.inc";
<"weddle/cubic_weddle_certificate.inc";

int cubicWeddleSuccess = 0;
cubicWeddleSuccess = verifyCubicWeddleSpecialization();
if (cubicWeddleSuccess != 1)
{
    exit;
}

"PASS: exact rational cubic Weddle specialization";
exit;
```

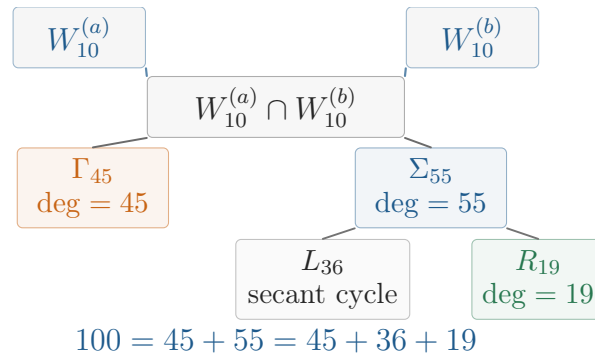


Figure 5.4: Degree ledger for the one-dimensional cycles in the exact rational specialization: $100 = 45 + 55$, while the degree-55 contribution has the internal ledger $55 = 36 + 19$. The diagram records degree accounting, not an unconditional equality of reduced schemes.

This picture should be compared with the classical case of quadric cones. For seven points in $\mathbb{P}^3$, the analogous locus is a curve of degree 6 and genus 3. In the cubic case the degree is much larger, but the principle is the same: adding one point beyond the number needed to define the Weddle surface lowers the dimension of the vertex locus from a surface to a curve.

5.4 Weddle Surfaces for Cones of Degree n

In accordance with the convention at the beginning of this chapter and with [6], all statements in this section about general point sets are made over $\mathbb{C}$.

The preceding two cases are part of a uniform construction. Let $n \geq 2$. A degree- n cone in $\mathbb{P}^3$ with vertex v corresponds, after projection from v , to a plane curve of

degree n . The dimension of the vector space of plane curves of degree n , and hence the natural number of base points for defining a Weddle-type surface, is

$$\binom{n+2}{2}.$$

For $n = 2$ this gives 6 points, and for $n = 3$ it gives 10 points.

Definition 5.4 (Generalized Weddle surface)

Let

$$\mathcal{P} = \{p_1, \dots, p_{\binom{n+2}{2}}\} \subset \mathbb{P}^3$$

be a general set of $\binom{n+2}{2}$ points. The *Weddle surface of n -ic cones* associated with $\mathcal{P}$ is the closure of the locus of all points $v \in \mathbb{P}^3$ for which there exists a cone of degree n with vertex v containing $\mathcal{P}$.

[6, Theorem 3.2] gives the degree

$$m = \binom{n+2}{3} = \frac{n(n+1)(n+2)}{6}.$$

We denote the corresponding generalized Weddle surface by W_m . Thus:

$$n = 2 \Rightarrow m = 4, \quad n = 3 \Rightarrow m = 10, \quad n = 4 \Rightarrow m = 20.$$

Figure 5.5 packages this degree formula for the first cases.

$$m = \binom{n+2}{3}$$

n	2	3	4
$ \mathcal{P} $	6	10	15
m	4	10	20

Figure 5.5: Degree of the generalized Weddle surface as a function of the cone degree n .

Remark 5.5 (General Weddle pattern)

For the Weddle-type surfaces considered here, Theorem 3.2 of [6] gives

$$\deg W_m = m = \binom{n+2}{3}.$$

In the two surface cases used explicitly in this chapter, this gives the quartic Weddle surface for $n = 2$ and the cubic-cone surface W_{10} for $n = 3$. Their base-point mul-

tiplicities and joining-line incidences are the documented statements of Theorems 5.2 and 5.3; we make no corresponding multiplicity claim here for arbitrary n .

When one increases the number of base points by one, the vertex locus drops from a surface to an arithmetically Cohen–Macaulay curve. Its degree is

$$\binom{\binom{n+2}{3}}{2}.$$

This formula is proved in [6, Proposition 5.2]. This gives degree 6 in the quadric case and degree 45, rather than 52, in the cubic case. The formula follows from the maximal minors of a $((\binom{n+2}{3}) - 1) \times \binom{n+2}{3}$ matrix of linear forms and the expected-codimension-two condition.

5.5 Conclusion

The Weddle construction provides a classical and very concrete way of producing special surfaces and curves from linear systems of cones. In the classical case of quadric cones through six points, one obtains a quartic surface with six nodes and fifteen lines. In the cubic case, ten base points give a surface of degree 10, with triple points at the base points and with all 45 joining lines contained in the surface.

The general construction for cones of degree n follows the same principle. The natural number of base points is $\binom{n+2}{2}$, and Theorem 3.2 of [6] gives the degree of the corresponding Weddle surface as

$$\binom{n+2}{3}.$$

Adding one further general base point lowers the expected dimension of the vertex locus and, by Proposition 5.2 of [6], leads to an ACM curve of degree $\binom{\binom{n+2}{3}}{2}$. The cubic value is 45. The exact rational specialization in Listing 5.1 verifies the corresponding degree ledger $100 = 45 + 55 = 45 + 36 + 19$ for its listed configuration.

For the purposes of this thesis, the main point is that Weddle loci offer a bridge between classical projective geometry and explicit computation. The construction is governed by projections and linear systems, while its geometry is encoded by concrete determinantal equations. This makes Weddle surfaces a useful source of examples in the study of special point configurations in projective space.

This also clarifies the sextic cone in the $\mathbf{H}_4$ chapter. Symmetry forces a degree-6 cone through the configuration with a general prescribed vertex, so the vertex condition holds generically rather than on a proper locus. Classical Weddle loci and unexpected cones therefore express the same interpolation problem for finite point sets.

6 Sextactic and Type-9 Points on the Fermat Cubic

A smooth plane cubic carries a natural elliptic-curve group law, and the torsion structure of that group law is reflected in the contact geometry of the curve. Flex points—the 9 points of order dividing 3—are characterized by their tangent lines meeting the curve with multiplicity 3. The 27 sextactic points form the set $F[6] \setminus F[3]$, with the precise decomposition

$$24 \text{ points of exact order } 6 + 3 \text{ nonzero points of order } 2.$$

At each of them an irreducible conic has contact order 6 [4, 27]. The 72 points of exact order 9—the *type-9 points* in the terminology of Gattazzo [16]—carry a tangent-residual map of period 3 [24, Section 3]. This chapter develops these phenomena for the Fermat cubic $F: x^3 + y^3 + z^3 = 0$, following the work of Szemberg–Szpond and Merta–Zięba [27, 24].

6.1 Introduction

Recall from Chapter 2 that the relevant higher-contact notions are fixed in Definition 2.19. Proposition 2.21 gives the torsion counts on a smooth complex cubic, Theorem 6.1 records the tangent-residual behavior of the Fermat-cubic sextactic points, and Theorem 6.3 records the coordinate field obtained for the type-9 points in the cited computation.

6.2 Sextactic Points on the Fermat Cubic

The terminology in this section follows Definition 2.19; the classical source for sextactic contact is Cayley [4], while the explicit Fermat-cubic enumeration is due to Szemberg and Szpond [27]. Consider the Fermat cubic

$$F: x^3 + y^3 + z^3 = 0, \tag{6.1}$$

viewed as an elliptic curve with the flex point $o = [1 : -1 : 0]$ as identity. Figure 6.1 shows the flex count and the local tangent picture.

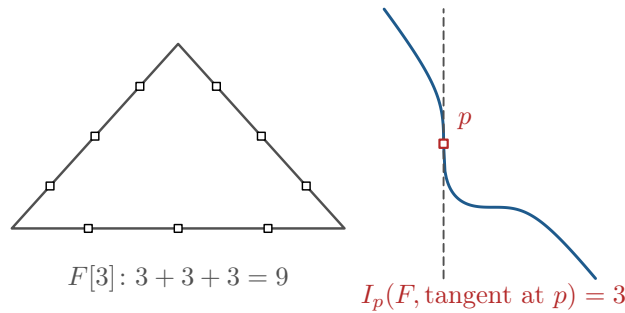


Figure 6.1: The 9 flexes of the Fermat cubic $F: x^3 + y^3 + z^3 = 0$ via the Hessian triangle $V(xy z)$, with a local flex tangent inset.

Theorem 6.1 ([24, Theorem 3.1])

For any sextactic point $s \in F$, the residual intersection of the tangent line to F at s is a flex point of F . Moreover, through each flex point of F pass exactly three tangent lines to F at sextactic points.

In group-law terms, the tangent-residual map sends s to $\nu(s) = -2s$. Since $6s = o$ and $s \notin F[3]$, we have $\nu(s) \in F[3]$, so the residual point is a flex. This argument includes both the 24 sextactic points of exact order 6 and the 3 nonzero points of order 2.

The auxiliary calculation below verifies the degree-9 flex scheme. The count $9 \cdot 3 = 27$ then uses the $[-2]$ -preimage description from Theorem 6.1; the script is a check of this deduction rather than a coordinate enumeration.

Printed dependencies: Listings A.1 and A.4.

Listing 6.1: Auxiliary exact check of the sextactic-point count

```
LIB "library/certificate_utils.lib";
LIB "sextactic_type9/fermat_checks.lib";

int sextacticCountCheckPassed = 0;
sextacticCountCheckPassed = verifySextacticCountCheck();
require(sextacticCountCheckPassed == 1,
  "the sextactic count check did not complete");

"PASS: sextactic count sanity check";
exit;
```

Figure 6.2 illustrates this sextactic contact and tangent-residual behavior.

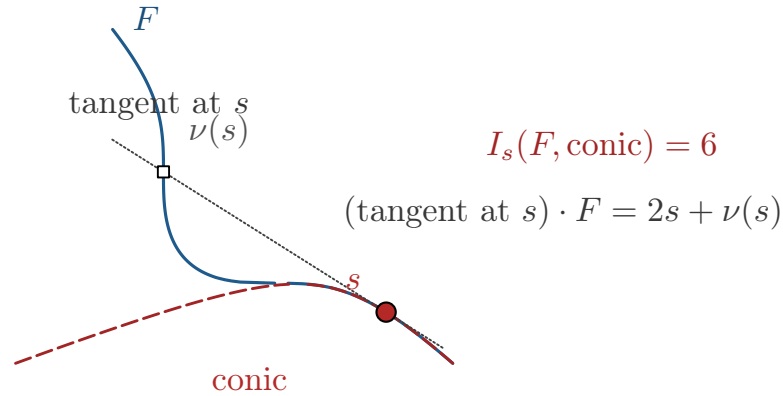


Figure 6.2: A sextactic point on the Fermat cubic with its hyperosculating conic (sixth-order contact). The tangent at the sextactic point meets F residually at a flex point.

6.2.1 Tangent lines and incidence

Each of the 27 sextactic points determines a tangent line to F ; denote these lines by L_i , $1 \leq i \leq 27$. By Theorem 6.1, these lines form nine triads: exactly three pass through each flex point of F .

6.2.2 Shadow points of the sextactic conic arrangement

Remark 6.2 (Terminology clarification)

The 54 *shadow points* are distinguished intersection points of the 108 conics through pairs of sextactic points described in Theorem 6.5. They lie off the Fermat cubic and therefore are not additional sextactic points [24, Theorem D(i)].

In the arrangement of these 108 conics, each of the 54 shadow points has multiplicity 6 or 9; the other intersections described there are ordinary double points [24, Theorem D(i)].

6.3 Points of Exact Order 9 on the Fermat Cubic

Here we use the torsion notation fixed in Proposition 2.21 and the type-9 terminology of Definition 2.19; the terminology and the classical count are due to Gattazzo [16]. By Abel's theorem in the form used in [24, Theorem 2.1], the type-9 points are precisely $F[9] \setminus F[3]$. Proposition 2.21 gives $|F[9]| = 9^2 = 81$ and $|F[3]| = 9$, hence there are $81 - 9 = 72$ such points, all of exact order 9.

Theorem 6.3 (Coordinate field of the type-9 points [24, Theorem A])

The 72 points of exact order 9 on $F: x^3 + y^3 + z^3 = 0$ have coordinates defined over $\mathbb{Q}(\sqrt[3]{3}, \zeta_9)$, where ζ_9 is a primitive 9th root of unity.

The next driver verifies the degree-9 flex scheme and checks the numerical consequence $|F[9]| - |F[3]| = 81 - 9 = 72$ of the complex torsion theorem. The coordinate field in Theorem 6.3 is not inferred from this auxiliary calculation.

Printed dependencies: Listings A.1 and A.4.

Listing 6.2: Auxiliary exact check of the type-9 torsion count

```
LIB "library/certificate_utils.lib";
LIB "sextatic_type9/fermat_checks.lib";

int type9CountCheckPassed = 0;
type9CountCheckPassed = verifyType9CountCheck();
require(type9CountCheckPassed == 1,
  "the type-9 count check did not complete");

"PASS: type-9 torsion count sanity check";
exit;
```

Fix a type-9 point $t \in F$. Figure 6.3 shows its tangent-residual behavior.

Figure 6.4 records the count $81 - 9 = 72$ and the period-3 orbit structure.

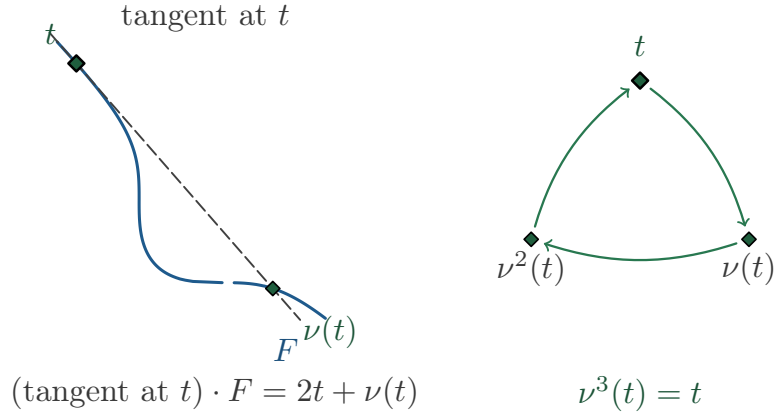


Figure 6.3: Tangent-residual behavior at a type-9 point on the Fermat cubic: the residual intersection is another point of type 9.

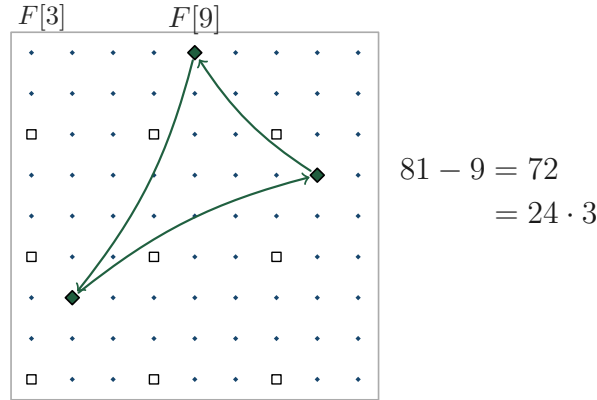


Figure 6.4: The 9-torsion structure on the Fermat cubic: the 72 points of exact order 9 are $F[9] \setminus F[3]$.

6.3.1 The tangent-residual map and its period-3 orbit structure

For the Fermat cubic, the tangent-residual map ν introduced in Chapter 2 sends a point to the residual intersection of its tangent line with the curve. If $p = [a : b : c] \in F$, then

$$\nu([a : b : c]) = [a(b^3 - c^3) : b(c^3 - a^3) : c(a^3 - b^3)].$$

The 72 type-9 points form a *complete intersection* of type $(3, 24)$ in the sense of Definition 2.3 [24, Section 2], and the tangent at any type-9 point meets F residually at another type-9 point [24, Theorem 3.1']. The degree-24 equation and the explicit point coordinates belong to the cited coordinate computation. The auxiliary scripts here check only the flex and torsion counts and the tangent-residual identities used in the argument; they do not construct those coordinates or that equation.

Theorem 6.4 (Period-3 mapping [24, Section 3])

The restriction of ν to the set $\mathcal{T}$ of type-9 points satisfies $\nu^3 = \text{id}$. Consequently, $\mathcal{T}$ decomposes into 24 disjoint orbits of size 3, namely $\{t, \nu(t), \nu^2(t)\}$.

PROOF SKETCH. In the group law, $\nu(t) = -2t$, so $\nu^2(t) = 4t$ and $\nu^3(t) = -8t$. Since $9t = o$ for every type-9 point, we have $-8t = -8t + 9t = t$, proving $\nu^3 = \text{id}$. The orbits have size exactly 3 because no type-9 point is fixed by ν (that would require $-2t = t$, i.e. $3t = o$, contradicting exact order 9). $\square$

Listing 6.3 verifies separately that the displayed candidate for $\nu(p)$ lies on F and on the tangent at p . The final count of 24 orbits uses the group-law identity $\nu(t) = -2t$ proved above.

Printed dependencies: Listings A.1 and A.4.

Listing 6.3: Auxiliary exact check of the tangent-residual period

```
LIB "library/certificate_utils.lib";
LIB "sextactic_type9/fermat_checks.lib";

int tangentResidualCheckPassed = 0;
tangentResidualCheckPassed = verifyTangentResidualCheck();
require(tangentResidualCheckPassed == 1,
    "the tangent-residual check did not complete");

"PASS: tangent-residual period-3 sanity check";
exit;
```

6.4 Conics through Sextactic and Type-9 Points

The higher-contact terminology used here is fixed in Definition 2.19. We also use the terminology for complete intersections from Definition 2.3. A conic that meets the cubic F at two distinct points, each with multiplicity 3, accounts for all $2 \cdot 3 = 6$ intersection points (by Bézout, $\deg F \cdot \deg D = 3 \cdot 2 = 6$, where D denotes the conic). The following two theorems enumerate such conics.

Theorem 6.5 (Conics through sextactic points [24, Theorem 4.5])

There are exactly 108 conics meeting F at two distinct sextactic points, each with multiplicity 3. Each sextactic point lies on exactly 8 of these conics.

The double-counting check is $27 \times 8/2 = 108$.

Theorem 6.6 (Conics through type-9 points [24, Theorem 4.5'])

There are exactly 324 conics meeting F at two distinct type-9 points, each with multiplicity 3. Each type-9 point lies on exactly 9 of these conics.

The double-counting check is $72 \times 9/2 = 324$. Figure 6.5 displays the two double-counting ledgers side by side.

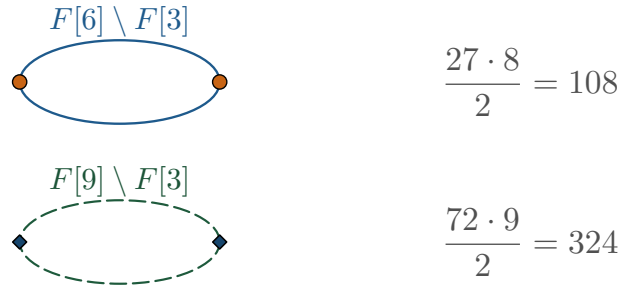


Figure 6.5: Incidence counts for conics through distinguished pairs on the Fermat cubic: sextactic points give 108 conics, while type-9 points give 324 conics.

6.4.1 Intersection structure of conics through a fixed point

For a fixed sextactic point, the 8 conics through it meet pairwise in points away from F . There are 24 such intersection points: 22 at which exactly two conics meet, and 2 at which three conics are concurrent. None of these 24 points lie on F [24, Theorem 4.6].

For a fixed type-9 point, the 9 conics through it produce 30 intersection points off the cubic: 27 pairwise intersections and 3 triple intersections. Again, none of these points lie on F [24, Theorem 4.6].

6.5 Conclusion

The torsion tower on the Fermat cubic yields three tiers of distinguished points: the 9 flexes in $F[3]$, the 27 sextactic points in $F[6] \setminus F[3]$ (24 of exact order 6 and 3 nonzero points of order 2), and the 72 type-9 points of exact order 9. The tangent-residual map $\nu: p \mapsto -2p$ maps sextactic points to flexes and cycles type-9 points in period-3 orbits. The associated conic configurations (108 conics through sextactic pairs and 324 through type-9 pairs) exhibit remarkably uniform incidence patterns. Figure 6.6 summarizes these three tiers.

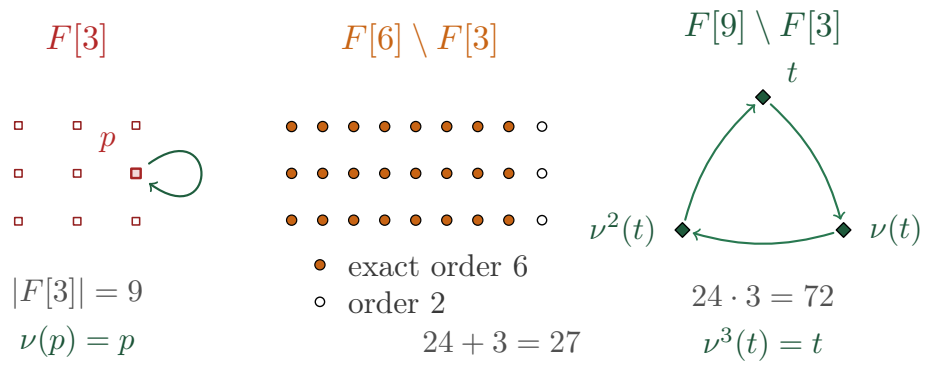


Figure 6.6: Summary of the distinguished Fermat cubic tiers: flexes, sextactic points, and type-9 points, together with the tangent-residual behavior at each level.

7 Concluding remarks

The aim of this thesis was to look at several special point configurations in projective geometry from a common perspective. Although the examples considered here come from rather different sources—line arrangements, root systems, classical surfaces, and torsion points on a cubic curve—they share a similar feature: each of them lies at the boundary between general theory and exceptional geometry.

In Chapter 3 we studied Yoshinaga’s arrangement of 18 lines in the projective plane, using the symbolic-power notation of Definition 2.8. This arrangement, obtained as a deformation of the sixth Fermat arrangement, gives a concrete configuration of 57 singular points for which the containment

$$I^{(3)} \subseteq I^2$$

fails. The proof combines the geometry of the arrangement with explicit computations in SINGULAR. The product of the defining lines, and in the full singular-locus case its product with a suitable cubic, provide explicit witnesses for the two non-containments.

Chapter 4 was devoted to the $\mathbf{H}_4$ configuration. This is a set of 60 points in $\mathbb{P}^3$ coming from the H_4 root system. Its general projection to the plane is a complete intersection of type $(6, 10)$, so it is a geproci set. What makes this example especially interesting is that it is neither a grid nor a half-grid. Thus it shows that the class of geproci configurations is genuinely larger than the known infinite line-based constructions.

In Chapter 5 we returned to a classical construction: the Weddle surface. Starting from the quartic surface parametrizing vertices of quadric cones through six points in $\mathbb{P}^3$, we discussed analogous loci for cones of higher degree. The cubic case already exhibits a rich structure: ten base points give a surface of degree 10, with triple points at the base points and all pairwise joining lines contained in the surface. Adding one more point leads naturally to curves of cone vertices and to delicate Bézout decompositions. In the cubic case the eleven-point locus is an ACM curve of degree 45 [6]; the exact rational specialization in Listing 5.1 verifies the one-dimensional degree ledger $100 = 45 + 55 = 45 + 36 + 19$ for its listed configuration.

Finally, in Chapter 6 we considered the Fermat cubic

$$x^3 + y^3 + z^3 = 0$$

and the special point configurations arising from its group law. The flexes are the 3-torsion points, the sextactic points form $F[6] \setminus F[3]$, and the type-9 points are the points of exact order 9. The tangent-residual map $\nu(p) = -2p$ organizes these sets in a simple way: it sends sextactic points to flexes and decomposes the type-9 points into 24 orbits of length 3. The associated conic configurations give further incidence structures attached to the torsion tower of the Fermat cubic.

The common theme of these chapters is that symmetry creates exceptional behavior. In a general configuration one does not expect symbolic-power containments to fail in such a controlled way, nor a random set of points in $\mathbb{P}^3$ to project to a complete intersection, nor torsion points on a cubic to support such rigid conic configurations. In each case the special symmetry of the configuration forces additional algebraic structure. Computation is useful here not as a replacement for geometry, but as a way of making this structure explicit and verifiable.

Several natural questions remain open. The first concerns geproci sets. Apart from grids, half-grids, and the $\mathbf{H}_4$ example, are there other sporadic geproci configurations? It would be interesting to search systematically for them, especially among point sets with large symmetry groups.

A second question concerns containment problems. The known counterexamples to

$$I^{(3)} \subseteq I^2$$

for point ideals in $\mathbb{P}^2$ are highly special and are closely related to arrangements with many multiple points. It remains unclear whether fundamentally different counterexamples exist, or whether symmetry and arrangement-theoretic constructions are essential in this setting.

The Weddle construction also suggests further problems. For cones of degree n , the degree of the corresponding Weddle surface follows a clear formula, and the curve obtained after imposing one additional base point has the determinantal degree $\binom{n+2}{3}$. The decomposition of nearby Weddle schemes into common-line schemes and residual schemes remains more subtle. A systematic treatment of these residual schemes would be desirable.

Finally, the Fermat cubic points toward higher torsion levels. One may ask whether points of exact order 12, or higher torsion points, support analogous configurations of curves with prescribed contact. Such questions continue the central theme: highly symmetric configurations connect classical geometry with exact computation and provide testing grounds for general conjectures.

A Computational Sources

The drivers printed in Chapters 3, 4, 5, and 6 invoke the certificate implementations, exact input data, and reusable libraries reproduced below. Every project-specific path appearing in a `LIB`, `read`, or `<` instruction in those drivers, including a nested input, is printed in full in the listings that follow. They use standard bases, primary decompositions, intersections, and normal forms provided by SINGULAR [8, 17].

A.1 Reusable libraries

A.1.1 Shared certificate utilities

`require`: common fail-closed assertion.

`projectiveDegree`: projective degree of a homogeneous standard ideal, computed from the second Hilbert-series vector returned by SINGULAR.

This library is used by the Weddle driver in Listing 5.1 and by the Fermat drivers in Listings 6.1–6.3.

Listing A.1: Shared utilities for exact computational checks

```
1 // Shared certificate utilities.
2 version="version certificate_utils.lib 1.1.0 Jul_2026";
3 category="Commutative Algebra";
4 info="
5 LIBRARY: certificate_utils.lib
6 PURPOSE: Shared assertions and numerical invariants for the certificates.
7 ";
8 proc require(int condition, string message)
9 {
10     if (condition == 0)
11     {
12         "FAIL: " + message;
13         exit;
14     }
15 }
16 proc projectiveDegree(ideal standardIdeal)
17 {
18     bigintvec hVector = hilb(standardIdeal,2);
19     bigint result = 0;
20     int entry;
21     for (entry = 1; entry <= size(hVector); entry++)
22     {
23         result = result + hVector[entry];
24     }
```

```

25  return(result);
26  }

```

A.1.2 Symbolic powers library

The procedures in this library are intentionally scoped to radical ideals of finite reduced sets of projective points whose component point ideals are generated by linear forms. In this setting $I(Z)^{(m)} = \bigcap_{p \in Z} I(p)^m$. The routine is not a general implementation of symbolic powers for arbitrary radical ideals.

Public procedures in Listing A.2:

- `symbolicPower(I,m);`
- `checkContainment(I,m,r);`
- `containmentThreshold(I,m);`
- `witnessElement(I,m,r).`

Listing A.2: Symbolic powers library

```

1  // Symbolic powers for finite reduced projective point sets.
2
3  version="version symbolic_powers.lib 1.1.0 Jul_2026";
4  category="Commutative Algebra";
5  info="
6  LIBRARY: symbolic_powers.lib
7
8  OVERVIEW:
9  For a finite reduced projective point set Z whose point ideals are
10 generated by linear forms, compute intersection_{p in Z} I(p)^m and test
11 containments in ordinary powers. Arbitrary radical ideals are outside
12 the scope of this library.
13
14 PROCEDURES:
15 symbolicPower(I,m);          compute m-th symbolic power I^(m)
16 checkContainment(I,m,r);     check if I^(m) subset I^r
17 containmentThreshold(I,m);   find largest r with I^(m) subset I^r
18 witnessElement(I,m,r);      find element in I^(m) \ I^r if containment fails
19 ";
20
21 LIB "primdec.lib";
22
23 proc symbolicPower(ideal I, int m)
24 "USAGE:    symbolicPower(I,m); I ideal, m int
25 RETURN:    ideal, intersection of I(p)^m over the component point ideals I(p)
26 NOTE:      Scope: the reduced projective point ideals described above.
27 EXAMPLE:   example symbolicPower; shows an example
28 "
29 {
30   if (m <= 0)
31   {
32     ERROR("Exponent must be positive");
33   }
34
35   list L = primdecGTZ(I);
36   int n = size(L);
37

```

```

38  ideal result = 1;
39  ideal p;
40  ideal pm;
41
42  for (int i = 1; i <= n; i++)
43  {
44      p = L[i][2];
45      pm = p^m;
46      result = intersect(result, pm);
47  }
48
49  return(result);
50 }
51 example
52 {
53     "EXAMPLE: "; echo = 2;
54     ring R = 0,(x,y,z),dp;
55     ideal I = x, y;
56     ideal I3 = symbolicPower(I, 3);
57     I3;
58     ideal I3_ord = I^3;
59     I3_ord;
60     reduce(I3, std(I3_ord));
61 }
62
63 proc checkContainment(ideal I, int m, int r)
64 "USAGE:   checkContainment(I,m,r); I ideal, m,r int
65 RETURN:   int, 1 if I^(m) subset I^r, 0 otherwise
66 NOTE:     Requires m,r >= 1 and the point-ideal scope described above.
67 EXAMPLE:  example checkContainment; shows an example
68 "
69 {
70     if (r <= 0)
71     {
72         ERROR("checkContainment: r must be positive");
73     }
74
75     ideal Im = symbolicPower(I, m);
76     ideal Ir = I^r;
77
78     ideal Ir_std = std(Ir);
79     poly remainder;
80
81     for (int i = 1; i <= size(Im); i++)
82     {
83         remainder = reduce(Im[i], Ir_std);
84         if (remainder != 0)
85         {
86             return(0);
87         }
88     }
89
90     return(1);
91 }
92 example
93 {
94     "EXAMPLE: "; echo = 2;
95     ring R = 0,(x,y,z),dp;
96     ideal I = x*y, x*z, y*z;
97
98     checkContainment(I, 2, 1);
99
100    checkContainment(I, 2, 2);
101 }
102

```

```

103 proc containmentThreshold(ideal I, int m)
104 "USAGE:   containmentThreshold(I,m); I ideal, m int
105 RETURN:   int, largest r such that  $I^{(m)} \subset I^r$ 
106 NOTE:     Requires  $m \geq 1$ , a non-zero proper ideal, and the point-ideal
107           scope described above.
108 EXAMPLE:  example containmentThreshold; shows an example
109 "
110 {
111   if (m < 1)
112   {
113     ERROR("containmentThreshold: m must be at least 1");
114   }
115
116   int isZero = 1;
117   for (int i = 1; i <= size(I); i++)
118   {
119     if (I[i] != 0) { isZero = 0; }
120   }
121   if (isZero)
122   {
123     ERROR("containmentThreshold: zero ideal is not allowed");
124   }
125
126   ideal Istd = std(I);
127   if (reduce(1, Istd) == 0)
128   {
129     ERROR("containmentThreshold: unit ideal is not allowed");
130   }
131
132   ideal Im = symbolicPower(I, m);
133   ideal Ir = Istd;
134   int r = 1;
135   int contained;
136   int j;
137
138   while (1)
139   {
140     contained = 1;
141     for (j = 1; j <= size(Im); j++)
142     {
143       if (reduce(Im[j], Ir) != 0) { contained = 0; }
144     }
145
146     if (contained == 0) { return(r - 1); }
147
148     r++;
149     Ir = std(Ir * I);
150   }
151 }
152 example
153 {
154   "EXAMPLE: "; echo = 2;
155   ring R = 0,(x,y,z),dp;
156   ideal I = x, y;
157
158   int r = containmentThreshold(I, 2);
159   r;
160 }
161
162 proc witnessElement(ideal I, int m, int r)
163 "USAGE:   witnessElement(I,m,r); I ideal, m,r int
164 RETURN:   poly, element in  $I^{(m)} \setminus I^r$  if containment fails, 0 otherwise
165 NOTE:     Requires  $m, r \geq 1$  and the point-ideal scope described above.
166           Returns the original generator, not its normal-form remainder.
167 EXAMPLE:  example witnessElement; shows an example

```

```

168 "
169 {
170   if (r <= 0)
171   {
172     ERROR("witnessElement: r must be positive");
173   }
174
175   ideal Im = symbolicPower(I, m);
176   ideal Ir = I^r;
177
178   ideal Ir_std = std(Ir);
179   poly remainder;
180
181   for (int i = 1; i <= size(Im); i++)
182   {
183     remainder = reduce(Im[i], Ir_std);
184     if (remainder != 0)
185     {
186       return(Im[i]);
187     }
188   }
189
190   return(0);
191 }
192 example
193 {
194   "EXAMPLE:"; echo = 2;
195   ring R = 0,(x,y,z),dp;
196   ideal P1 = x, y;
197   ideal P2 = x, z;
198   ideal P3 = y, z;
199   ideal I = intersect(P1, P2, P3);
200
201   poly witness = witnessElement(I, 2, 2);
202   witness;
203
204   ideal I2_sym = symbolicPower(I, 2);
205   reduce(witness, std(I2_sym));
206
207   ideal I2 = I^2;
208   reduce(witness, std(I2));
209 }

```

A.1.3 Yoshinaga arrangement library

The library below provides construction routines for the Fermat and Yoshinaga line arrangements. The complete incidence and containment assertions are performed by the driver in Listing 3.1.

Public procedures in Listing A.3:

- `fermatArrangement();`
- `yoshinagaLines();`
- `yoshinagaSingularPoints();`
- `verifyNonContainment();`
- `countIntersectionPoints();`

- `hesseElement()`.

Listing A.3: Yoshinaga arrangement library

```

1 // Yoshinaga arrangement helpers.
2
3 version="version yoshinaga_arrangement.lib 1.0.1 Jul_2026";
4
5 LIB "primdec.lib";
6 LIB "library/symbolic_powers.lib";
7
8 proc fermatArrangement()
9 "USAGE:   fermatArrangement();
10 RETURN:  ideal, the Fermat arrangement Q_6(x,y,z)
11 NOTE:    Returns the product (x^6-y^6)(y^6-z^6)(z^6-x^6).
12 EXAMPLE: example fermatArrangement; shows an example
13 "
14 {
15   poly Q6 = (x^6 - y^6) * (y^6 - z^6) * (z^6 - x^6);
16   ideal I = Q6;
17   return(I);
18 }
19 example
20 {
21   "EXAMPLE:"; echo = 2;
22   ring R = 0,(x,y,z),dp;
23   ideal Q6 = fermatArrangement();
24   Q6;
25   if (deg(Q6[1]) != 18) { ERROR("Expected Fermat arrangement degree 18"); }
26 }
27
28 proc yoshinagaLines()
29 "USAGE:   yoshinagaLines();
30 RETURN:  ideal, Yoshinaga's 18 lines as product polynomial
31 NOTE:    Requires a primitive sixth root 'a' and uses c=15.
32 EXAMPLE: example yoshinagaLines; shows an example
33 "
34 {
35   int c = 15;
36
37   poly P1 = (x^3 - y^3) * (x + y - c*z) *
38             (a*x + a^5*y + c*z) * (a^5*x + a*y + c*z);
39
40   map tau = basering, y, z, x;
41   poly P2 = tau(P1);
42   poly P3 = tau(P2);
43
44   poly P = P1 * P2 * P3;
45
46   ideal I = P;
47   return(I);
48 }
49 example
50 {
51   "EXAMPLE:"; echo = 2;
52   ring R = (0,a),(x,y,z),dp;
53   minpoly = a^2 - a + 1;
54
55   ideal lines = yoshinagaLines();
56   lines;
57   if (deg(lines[1]) != 18) { ERROR("Expected Yoshinaga arrangement degree 18"); }
58 }
59
60 proc yoshinagaSingularPoints()

```

```

61 "USAGE:    yoshinagaSingularPoints();
62 RETURN:    ideal, the reduced singular-locus ideal
63 EXAMPLE:   example yoshinagaSingularPoints; shows an example
64 "
65 {
66     ideal lines = yoshinagaLines();
67     ideal J = jacob(lines);
68     ideal singLocus = lines, J;
69     ideal singRad = radical(singLocus);
70
71     return(singRad);
72 }
73 example
74 {
75     "EXAMPLE: "; echo = 2;
76     ring R = (0,a),(x,y,z),dp;
77     minpoly = a^2 - a + 1;
78
79     ideal sing = yoshinagaSingularPoints();
80     ideal singStd = std(sing);
81
82     if (dim(singStd) == 1)
83     {
84         print("homogeneous point ideal has Krull dimension 1");
85     }
86     else
87     {
88         ERROR("Expected homogeneous point ideal of Krull dimension 1");
89     }
90
91     bigintvec singH = hilb(singStd, 2); bigint singDegree = 0;
92     for (int i=1; i<=size(singH); i++) { singDegree = singDegree + singH[i]; }
93     if (singDegree == 57)
94     {
95         print("reduced singular locus degree: 57");
96     }
97     else
98     {
99         ERROR("Expected reduced singular locus of degree 57");
100     }
101 }
102
103 proc verifyNonContainment()
104 "USAGE:    verifyNonContainment();
105 RETURN:    int, 1 if  $I^{(3)}$  not subset  $I^2$  is verified; raises an error otherwise
106 NOTE:      Computes symbolic and ordinary powers and checks the witness  $L \cdot h$ .
107 EXAMPLE:   example verifyNonContainment; shows an example
108 "
109 {
110     ideal I = yoshinagaSingularPoints();
111
112     print("Computing  $I^{(3)}$ ...");
113     ideal I3_sym = symbolicPower(I, 3);
114
115     print("Computing  $I^2$ ...");
116     ideal I2_ord = I^2;
117
118     print("Checking containment  $I^{(3)}$  subset  $I^2$ ...");
119     int contains = checkContainment(I, 3, 2);
120
121     if (contains == 0)
122     {
123         print(" $I^{(3)}$  is not contained in  $I^2$ ");
124
125         // The witness includes the line product L.

```

```

126     poly witness = hesseElement();
127     if ((reduce(witness, std(I3_sym)) != 0) ||
128         (reduce(witness, std(I2_ord)) == 0))
129     {
130         ERROR("The explicit full witness L*h failed its membership checks");
131     }
132     print("Full witness L*h (in I^(3) \\ I^2):");
133     print(witness);
134
135     return(1);
136 }
137 else
138 {
139     ERROR("UNEXPECTED: I^(3) subset I^2 (containment holds)");
140     return(0);
141 }
142 }
143 example
144 {
145     "EXAMPLE: "; echo = 2;
146     ring R = (0,a),(x,y,z),dp;
147     minpoly = a^2 - a + 1;
148
149     int result = verifyNonContainment();
150     result;
151 }
152
153 proc countIntersectionPoints()
154 "USAGE:    countIntersectionPoints();
155 RETURN:    bigint, projective degree of the reduced singular-point ideal
156 NOTE:      Computes the degree from the Hilbert-series vector.
157 "
158 {
159     ideal sing = yoshinagaSingularPoints();
160     bigintvec singH = hilb(std(sing), 2); bigint count = 0;
161     for (int i=1; i<=size(singH); i++) { count = count + singH[i]; }
162     return(count);
163 }
164
165 proc hesseElement()
166 "USAGE:    hesseElement();
167 RETURN:    poly, the full witness L*h in I^(3) \\ I^2
168 NOTE:      L is the product of the 18 Yoshinaga lines and
169             h = x^3 + y^3 + z^3 - (3379/225)*xyz is the auxiliary cubic.
170             The factor h alone is not a witness.
171 "
172 {
173     ideal lines = yoshinagaLines();
174     poly h = x^3 + y^3 + z^3 - (3379/225)*x*y*z;
175     poly witness = lines[1] * h;
176     return(witness);
177 }
178 example
179 {
180     "EXAMPLE: "; echo = 2;
181     ring R = (0,a),(x,y,z),dp;
182     minpoly = a^2 - a + 1;
183     poly Lh = hesseElement();
184     deg(Lh);
185 }

```

A.1.4 Fermat auxiliary checks

The procedures below verify the flex scheme, a representative flex tangent, the torsion-theorem counts 27 and 72, and the two polynomial identities used for the tangent-residual period. The count procedures use the group-law and torsion results stated in Chapter 6; they do not enumerate the corresponding point coordinates.

Public procedures: `verifyFermatFlexCheck()`, `verifySextacticCountCheck()`, `verifyType9CountCheck()`, and `verifyTangentResidualCheck()`.

These procedures are invoked by the Fermat drivers in Listings 6.1–6.3; their shared assertions are supplied by Listing A.1.

Listing A.4: Exact auxiliary checks for the Fermat cubic

```

1 // Fermat-cubic checks in characteristic zero.
2
3 version="version fermat_checks.lib 1.0.0 Jul_2026";
4 category="Algebraic Geometry";
5 info="
6 LIBRARY: fermat_checks.lib
7 PURPOSE: Exact checks supporting the torsion arguments for the Fermat cubic.
8 ";
9
10 // Flexes and one tangent.
11 proc verifyFermatFlexCheck()
12 {
13   ring R = (0,w),(x,y,z),dp;
14   minpoly = w^2 + w + 1;
15
16   poly F = x^3 + y^3 + z^3;
17   matrix H = jacob(ideal(jacob(ideal(F))));
18   ideal flexes = std(ideal(F,det(H)));
19
20   require(dim(flexes) == 1,
21     "the projective flex scheme must be zero-dimensional");
22   require(projectiveDegree(flexes) == 9,
23     "the projective flex scheme must have degree nine");
24
25   poly tangent = x + w^2*y;
26   poly tangentRestriction = NF(F,std(ideal(tangent)));
27   require(tangentRestriction == z^3,
28     "the representative flex tangent must have triple contact");
29
30   "flex scheme degree: 9";
31   "representative flex tangent: triple contact";
32   return(1);
33 }
34
35 // Sextactic count from the [-2] theorem.
36 proc verifySextacticCountCheck()
37 {
38   ring R = (0,w),(x,y,z),dp;
39   minpoly = w^2 + w + 1;
40
41   poly F = x^3 + y^3 + z^3;
42   matrix H = jacob(ideal(jacob(ideal(F))));
43   ideal flexes = std(ideal(F,det(H)));
44   bigint flexCount = projectiveDegree(flexes);
45
46   require(dim(flexes) == 1,

```

```

47     "the projective flex scheme must be zero-dimensional");
48     require(flexCount == 9,
49         "the projective flex scheme must have degree nine");
50
51     // Three nonflex preimages per flex.
52     int preimagesPerFlex = 4;
53     bigint sextacticCount = flexCount * (preimagesPerFlex - 1);
54     require(sextacticCount == 27,
55         "the multiplication-by-minus-two count must equal 27");
56
57     "flexes: 9";
58     "sextactic points (multiplication-by-minus-two theorem): 27";
59     return(1);
60 }
61
62 // Type-9 count from torsion.
63 proc verifyType9CountCheck()
64 {
65     ring R = (0,w),(x,y,z),dp;
66     minpoly = w^2 + w + 1;
67
68     poly F = x^3 + y^3 + z^3;
69     matrix H = jacob(ideal(jacob(ideal(F))));
70     ideal flexScheme = std(ideal(F,det(H)));
71     bigint order3Count = projectiveDegree(flexScheme);
72
73     require(dim(flexScheme) == 1,
74         "the projective flex scheme must be zero-dimensional");
75     require(order3Count == 9,
76         "the projective flex scheme must have degree nine");
77
78     // The flexes are E[3], and |E[9]|=81.
79     bigint order9DividingCount = 9^2;
80     bigint type9Count = order9DividingCount - order3Count;
81     require(type9Count == 72,
82         "the exact-order-nine count must equal 81 minus 9");
83
84     "3-torsion points: 9";
85     "type-9 points (torsion theorem): 72";
86     return(1);
87 }
88
89 // Residual map and period.
90 proc verifyTangentResidualCheck()
91 {
92     ring R = 0,(a,b,c),dp;
93
94     poly F = a^3 + b^3 + c^3;
95     poly na = a*(b^3-c^3);
96     poly nb = b*(c^3-a^3);
97     poly nc = c*(a^3-b^3);
98
99     poly cubicIdentity =
100         reduce(na^3 + nb^3 + nc^3,std(ideal(F)));
101     poly tangentIdentity = a^2*na + b^2*nb + c^2*nc;
102
103     require(cubicIdentity == 0,
104         "the residual candidate must lie on the Fermat cubic");
105     require(tangentIdentity == 0,
106         "the residual candidate must lie on the tangent at [a:b:c]");
107
108     // The residual map is [-2].
109     int firstDifference = -2 - 1;
110     int secondDifference = 4 - 1;
111     int thirdDifference = -8 - 1;

```

```

112  int firstRemainder =
113      firstDifference - 9*(firstDifference div 9);
114  int secondRemainder =
115      secondDifference - 9*(secondDifference div 9);
116  int thirdRemainder =
117      thirdDifference - 9*(thirdDifference div 9);
118
119  require(firstRemainder != 0 && secondRemainder != 0,
120      "the first two iterates must not fix exact-order-nine points");
121  require(thirdRemainder == 0,
122      "the third iterate must fix 9-torsion");
123
124  int type9Count = 9^2 - 3^2;
125  int orbitCount = type9Count div 3;
126  require(type9Count == 72 && orbitCount == 24,
127      "the 72 type-9 points must form 24 three-cycles");
128
129  "residual formula: cubic and tangent identities verified";
130  "tangent-residual period on type-9 points: 3";
131  "three-cycles: 24";
132  return(1);
133 }

```

A.2 Yoshinaga certificate implementation

Listing A.5 contains the implementation invoked by Listing 3.1. Listing A.6 supplies the 48 triple points, 9 double points, and 18 projective lines. Its public entry point, `verifyYoshinagaCertificate()`, returns 1 only after all incidence and containment assertions have succeeded. Thus the complete printed dependencies of Listing 3.1 are Listings A.5–A.6.

Listing A.5: Certificate implementation for the Yoshinaga arrangement

```

1  // Yoshinaga certificate over Q(e).
2
3  version="version yoshinaga_certificate.lib 1.0.0 Jul_2026";
4  category="Commutative Algebra";
5  info="
6  LIBRARY: yoshinaga_certificate.lib
7  PURPOSE: Verify the exact Yoshinaga incidence and containment certificate.
8  PROCEDURES:
9      verifyYoshinagaCertificate()
10 ";
11
12 static proc yoshinagaPointIdeal(number pp, number qq, number rr)
13 {
14     matrix m[2][3] = pp,qq,rr, x,y,z;
15     ideal I = std(minor(m,2));
16     return(I);
17 }
18
19 static proc yoshinagaAllGeneratorsZero(ideal N)
20 {
21     for (int k=1; k<=size(N); k++)
22     {
23         if (N[k] != 0) { return(0); }
24     }
25     return(1);

```

```

26 }
27
28 proc verifyYoshinagaCertificate()
29 "USAGE:  verifyYoshinagaCertificate();
30 RETURN:  int, 1 after every exact assertion succeeds
31 NOTE:    Requires the data exported by yoshinaga_input.inc.
32 "
33 {
34   int i;
35   int j;
36   int k;
37   int a;
38   int b;
39
40   // Lines and their product.
41   if (size(P3) != 144)
42   {
43     ERROR("The triple-point list must contain 48 triples");
44   }
45   if (size(P2) != 27)
46   {
47     ERROR("The double-point list must contain 9 triples");
48   }
49
50   matrix arrangementLines[18][1];
51   poly L=1;
52   for (i=1; i<=18; i++)
53   {
54     arrangementLines[i,1]=
55       lineCoefficients[i,1]*x+
56       lineCoefficients[i,2]*y+
57       lineCoefficients[i,3]*z;
58     L=L*arrangementLines[i,1];
59   }
60
61   // Triple-point ideal.
62   ideal I = 1;
63   ideal Isym3 = 1;
64   ideal Jp;
65
66   "=== triple points only: I3^(3) vs I3^2 ===";
67   for (i=1; i<=(size(P3) div 3); i++)
68   {
69     Jp=yoshinagaPointIdeal(P3[3*i-2],P3[3*i-1],P3[3*i]);
70     I=intersect(I,Jp);
71     Isym3=intersect(Isym3,Jp^3);
72   }
73
74   ideal I3=std(I);
75   ideal I3_ord2=std(I3^2);
76   ideal I3_sym3=std(Isym3);
77   ideal N3=NF(I3_sym3,I3_ord2);
78
79   // Check the 57 points.
80   matrix singularPoints[57][3];
81   for (i=1; i<=48; i++)
82   {
83     for (j=1; j<=3; j++)
84     {
85       singularPoints[i,j]=P3[3*i-3+j];
86     }
87   }
88   for (i=1; i<=9; i++)
89   {
90     for (j=1; j<=3; j++)

```

```

91     {
92         singularPoints[48+i,j]=P2[3*i-3+j];
93     }
94 }
95
96 int sameProjectiveObject;
97 for (i=1; i<=57; i++)
98 {
99     if (singularPoints[i,1]==0 &&
100         singularPoints[i,2]==0 &&
101         singularPoints[i,3]==0)
102     {
103         ERROR("A listed Yoshinaga point has zero homogeneous coordinates");
104     }
105
106     for (j=i+1; j<=57; j++)
107     {
108         sameProjectiveObject=1;
109         for (a=1; a<=3; a++)
110         {
111             for (b=a+1; b<=3; b++)
112             {
113                 if (singularPoints[i,a]*singularPoints[j,b]
114                     -singularPoints[i,b]*singularPoints[j,a] != 0)
115                 {
116                     sameProjectiveObject=0;
117                 }
118             }
119         }
120         if (sameProjectiveObject==1)
121         {
122             ERROR("Two listed Yoshinaga points are equal projectively");
123         }
124     }
125 }
126
127 // Check the 18 lines.
128 for (i=1; i<=18; i++)
129 {
130     if (lineCoefficients[i,1]==0 &&
131         lineCoefficients[i,2]==0 &&
132         lineCoefficients[i,3]==0)
133     {
134         ERROR("A Yoshinaga line has zero homogeneous coefficients");
135     }
136
137     for (j=i+1; j<=18; j++)
138     {
139         sameProjectiveObject=1;
140         for (a=1; a<=3; a++)
141         {
142             for (b=a+1; b<=3; b++)
143             {
144                 if (lineCoefficients[i,a]*lineCoefficients[j,b]
145                     -lineCoefficients[i,b]*lineCoefficients[j,a] != 0)
146                 {
147                     sameProjectiveObject=0;
148                 }
149             }
150         }
151         if (sameProjectiveObject==1)
152         {
153             ERROR("Two Yoshinaga lines are equal projectively");
154         }
155     }

```

```

156 }
157
158 // Point-line incidences.
159 intmat pointLineIncidence[57][18];
160 int incidenceMultiplicity;
161 int totalIncidences=0;
162 poly incidenceValue;
163 for (i=1; i<=57; i++)
164 {
165     incidenceMultiplicity=0;
166     for (j=1; j<=18; j++)
167     {
168         incidenceValue =
169             lineCoefficients[j,1]*singularPoints[i,1]+
170             lineCoefficients[j,2]*singularPoints[i,2]+
171             lineCoefficients[j,3]*singularPoints[i,3];
172         if (incidenceValue==0)
173         {
174             pointLineIncidence[i,j]=1;
175             incidenceMultiplicity=incidenceMultiplicity+1;
176             totalIncidences=totalIncidences+1;
177         }
178     }
179     if (i<=48 && incidenceMultiplicity!=3)
180     {
181         ERROR("A listed triple point does not lie on exactly three lines");
182     }
183     if (i>48 && incidenceMultiplicity!=2)
184     {
185         ERROR("A listed double point does not lie on exactly two lines");
186     }
187 }
188 if (totalIncidences != 48*3+9*2)
189 {
190     ERROR("The Yoshinaga point-line incidence total is not 162");
191 }
192
193 // All 153 line pairs.
194 intmat pairHits[57][1];
195 int linePairCount=0;
196 int commonListedPoints;
197 int commonPointIndex;
198 for (i=1; i<=18; i++)
199 {
200     for (j=i+1; j<=18; j++)
201     {
202         linePairCount=linePairCount+1;
203         commonListedPoints=0;
204         commonPointIndex=0;
205         for (k=1; k<=57; k++)
206         {
207             if (pointLineIncidence[k,i]==1 &&
208                 pointLineIncidence[k,j]==1)
209             {
210                 commonListedPoints=commonListedPoints+1;
211                 commonPointIndex=k;
212             }
213         }
214         if (commonListedPoints!=1)
215         {
216             ERROR("A line pair does not meet at exactly one listed point");
217         }
218         pairHits[commonPointIndex,1]=pairHits[commonPointIndex,1]+1;
219     }
220 }

```

```

221  if (linePairCount != 153)
222  {
223      ERROR("The arrangement must have exactly 153 unordered line pairs");
224  }
225
226  int totalPairHits=0;
227  for (i=1; i<=57; i++)
228  {
229      totalPairHits=totalPairHits+pairHits[i,1];
230      if (i<=48 && pairHits[i,1]!=3)
231      {
232          ERROR("A triple point must account for three pairs of lines");
233      }
234      if (i>48 && pairHits[i,1]!=1)
235      {
236          ERROR("A double point must account for one pair of lines");
237      }
238  }
239  if (totalPairHits != 48*3+9)
240  {
241      ERROR("The line-pair incidence balance is not 48*3+9=153");
242  }
243
244  "projective singular points: 57";
245  "projective arrangement lines: 18";
246  "point-line multiplicities: 48*3 + 9*2";
247  "line-pair intersections covered: 153";
248
249  // Triple-point witness L.
250  poly h=x^3+y^3+z^3-(3379/225)*x*y*z;
251  poly W=L*h;
252
253  "number of triple points";
254  int triplePointCount=size(P3) div 3;
255  triplePointCount;
256  if (triplePointCount != 48)
257  {
258      ERROR("Expected 48 triple points in the explicit point list");
259  }
260  "NF(I3^(3), I3^2)";
261  if (yoshinagaAllGeneratorsZero(N3))
262  {
263      ERROR("Unexpected containment I3^(3) subset I3^2");
264  }
265  else
266  {
267      "I3^(3) subset I3^2 : NO (all generator normal forms checked)";
268  }
269  "witness check: L in I3^(3)";
270  if (NF(L,I3_sym3)==0)
271  {
272      "YES";
273  }
274  else
275  {
276      ERROR("L is not in I3^(3)");
277  }
278  "witness check: L in I3^2";
279  if (NF(L,I3_ord2)==0)
280  {
281      ERROR("L unexpectedly belongs to I3^2");
282  }
283  else
284  {
285      "NO";

```

```

286 }
287
288 // Full-locus witness L*h.
289 "=== all singular points: I^(3) vs I^2 ===";
290 I=1;
291 Isym3=1;
292 list Pall=P3+P2;
293 for (i=1; i<=(size(Pall) div 3); i++)
294 {
295     Jp=yoshinagaPointIdeal(Pall[3*i-2],Pall[3*i-1],Pall[3*i]);
296     I=intersect(I,Jp);
297     Isym3=intersect(Isym3,Jp^3);
298 }
299
300 ideal Iall=std(I);
301 ideal Iall_ord2=std(Iall^2);
302 ideal Iall_sym3=std(Isym3);
303 ideal Nall=NF(Iall_sym3,Iall_ord2);
304
305 "number of all singular points";
306 int allPointCount=size(Pall) div 3;
307 allPointCount;
308 if (allPointCount != 57)
309 {
310     ERROR("Expected 57 singular points in the combined point list");
311 }
312 "homogeneous ideal invariants";
313 "Krull dimension: "+string(dim(Iall));
314 bigintvec allH=hilb(std(Iall),2);
315 bigint allDegree=0;
316 for (i=1; i<=size(allH); i++)
317 {
318     allDegree=allDegree+allH[i];
319 }
320 "projective degree: "+string(allDegree);
321 if (dim(Iall) != 1)
322 {
323     ERROR(
324         "Expected the homogeneous 57-point ideal to have Krull dimension 1"
325     );
326 }
327 if (allDegree != 57)
328 {
329     ERROR("Expected the reduced point ideal to have projective degree 57");
330 }
331 "NF(I^(3), I^2)";
332 if (yoshinagaAllGeneratorsZero(Nall))
333 {
334     ERROR("Unexpected containment I^(3) subset I^2");
335 }
336 else
337 {
338     "I^(3) subset I^2 : NO (all generator normal forms checked)";
339 }
340 "witness check: L*h in I^(3)";
341 if (NF(W,Iall_sym3)==0)
342 {
343     "YES";
344 }
345 else
346 {
347     ERROR("L*h is not in I^(3)");
348 }
349 "witness check: L*h in I^2";
350 if (NF(W,Iall_ord2)==0)

```

```

351 {
352     ERROR("L*h unexpectedly belongs to I^2");
353 }
354 else
355 {
356     "NO";
357 }
358
359 "=== generator-degree check for I ===";
360 ideal S=std(Iall);
361 ideal M=minbase(S);
362 intvec dS=intvec(size(S));
363 intvec dM=intvec(size(M));
364 for (i=1; i<=size(S); i++)
365 {
366     dS[i]=deg(S[i]);
367 }
368 for (i=1; i<=size(M); i++)
369 {
370     dM[i]=deg(M[i]);
371 }
372 "size(std(I))";
373 size(S);
374 "degrees(std(I))";
375 dS;
376 "size(minbase(std(I)))";
377 size(M);
378 "degrees(minbase(std(I)))";
379 dM;
380
381 return(1);
382 }

```

Listing A.6: Exact projective input for the Yoshinaga certificate

```

1 // Yoshinaga input over Q(e).
2
3 // Nine double points.
4 list P2 =
5     15,15,2,
6     15,(15e-15),(-2e),
7     15,(-15e),(2e-2),
8     2/15,1,1,
9     (-2/15e),(e-1),1,
10    (2/15e-2/15),(-e),1,
11    15,2,15,
12    15,(2e-2),(-15e),
13    15,(-2e),(15e-15);
14
15 // 48 triple points.
16 list P3 =
17     0,0,(e+1),
18     (15e-15),(15e-15),(-e+1),
19     1/15,1/15,14/15,
20     (-1/15e),(-1/15e),(-1/15e+16/15),
21     (1/15e-1/15),(1/15e-1/15),(1/15e+1),
22     1,1,1,
23     (-e),(-e),1,
24     (e-1),(e-1),1,
25     (-15e),15,(-e+1),
26     (-1/15e),1/15,(-e-1/15),
27     (1/15e-1/15),(-1/15e),(-16/15e+1/15),
28     1/15,(1/15e-1/15),(-14/15e),
29     (-e),1,1,

```

```

30  (e-1), (-e), 1,
31  1, (e-1), 1,
32  (15e-15), 15, (e),
33  (1/15e-1/15), 1/15, (e-16/15),
34  1/15, (-1/15e), (14/15e-14/15),
35  (-1/15e), (1/15e-1/15), (16/15e-1),
36  (e-1), 1, 1,
37  1, (-e), 1,
38  (-e), (e-1), 1,
39  -16/15, -224/15, -16/15,
40  (-14/15e+1), (-1/15e-15), (-1/15e-14/15),
41  (14/15e+1/15), (1/15e-226/15), (1/15e-1),
42  14, 1, 1,
43  (e+15), (-e), 1,
44  (-e+16), (e-1), 1,
45  (-14/15e+1), (-15e+226/15), (e-1/15),
46  (14/15e+1/15), (-226/15e+15), (14/15e+1/15),
47  -16/15, (-224/15e+224/15), (16/15e),
48  (16e-15), 1, 1,
49  (14e-14), (-e), 1,
50  (15e-16), (e-1), 1,
51  (14/15e+1/15), (15e+1/15), (-e+14/15),
52  -16/15, (224/15e), (-16/15e+16/15),
53  (-14/15e+1), (226/15e-1/15), (-14/15e+1),
54  (-16e+1), 1, 1,
55  (-15e-1), (-e), 1,
56  (-14e), (e-1), 1,
57  (-2/225e+1/225), (-1/15e-1/15), (-1/15e+2/15),
58  (2/225e-1/225), (1/15e-2/15), (1/15e+1/15),
59  (e+1), 0, 0,
60  (-2/225e+1/225), (2/15e-1/15), (2/15e-1/15),
61  (-30e+15), (-e+2), (15e+15),
62  (30e-15), (e+1), (-15e+30),
63  (-30e+15), (2e-1), (-30e+15),
64  0, (-e-1), 0;
65
66  // 18 line coefficient triples.
67  matrix lineCoefficients[18][3] =
68    1, -1, 0,
69    1, e, 0,
70    1, (-e+1), 0,
71    1, -15, 1,
72    1, (-1/15), (-1/15),
73    1, 1, -15,
74    1, (e/15), ((-e+1)/15),
75    1, ((-e+1)/15), (e/15),
76    1, (-15e+15), (-e),
77    1, (15e), (e-1),
78    1, (e-1), (15e),
79    1, (-e), (-15e+15),
80    1, 0, -1,
81    1, 0, (-e+1),
82    1, 0, e,
83    0, 1, -1,
84    0, 1, (-e+1),
85    0, 1, e;

```

A.3 H_4 certificates

Both Chapter 4 drivers use the same listed coordinates of the 60 points. The incidence data distinguish the pairs defining the two five-line families from the two six-line covers.

The configuration driver in Listing 4.1 uses Listings A.7, A.9, and A.10. The generic projection driver in Listing 4.2 uses Listings A.8–A.10.

`verifyH4Configuration`: incidence certificate;

`verifyGenericProjection`: ring-changing projection certificate.

Listing A.7: Implementation of the H_4 configuration certificate

```

1 // H4 configuration over  $Q(u)$ , where  $u^2-u-1=0$ .
2
3 version = "version h4_configuration.lib 1.0.0 Jul_2026";
4
5 proc det3(poly a11, poly a12, poly a13,
6           poly a21, poly a22, poly a23,
7           poly a31, poly a32, poly a33)
8 {
9     return(
10         a11*(a22*a33-a23*a32)
11         -a12*(a21*a33-a23*a31)
12         +a13*(a21*a32-a22*a31));
13 }
14
15 proc areCollinear(matrix p, int i, int j, int k)
16 {
17     if (det3(p[i,1],p[i,2],p[i,3],
18             p[j,1],p[j,2],p[j,3],
19             p[k,1],p[k,2],p[k,3]) != 0) { return(0); }
20     if (det3(p[i,1],p[i,2],p[i,4],
21             p[j,1],p[j,2],p[j,4],
22             p[k,1],p[k,2],p[k,4]) != 0) { return(0); }
23     if (det3(p[i,1],p[i,3],p[i,4],
24             p[j,1],p[j,3],p[j,4],
25             p[k,1],p[k,3],p[k,4]) != 0) { return(0); }
26     if (det3(p[i,2],p[i,3],p[i,4],
27             p[j,2],p[j,3],p[j,4],
28             p[k,2],p[k,3],p[k,4]) != 0) { return(0); }
29     return(1);
30 }
31
32 proc verifyH4Configuration()
33 {
34     if (size(L) != 240)
35     { ERROR("H4 coordinate list must have 240 entries"); }
36
37     matrix pts[60][4];
38     int i; int j; int k; int a; int b;
39     for (i=1; i<=60; i++)
40     {
41         for (j=1; j<=4; j++) { pts[i,j]=L[4*(i-1)+j]; }
42     }
43
44     // Distinct points.
45     int same;
46     for (i=1; i<=60; i++)
47     {
48         for (j=i+1; j<=60; j++)
49         {
50             same=1;
51             for (a=1; a<=4; a++)
52             {
53                 for (b=a+1; b<=4; b++)
54                 {
55                     if (pts[i,a]*pts[j,b]-pts[i,b]*pts[j,a] != 0)
56                     { same=0; }

```

```

57     }
58     }
59     if (same==1) { ERROR("Repeated projective H4 point"); }
60 }
61 }
62
63 // The (60_15) incidences.
64 intmat incidence[60][60];
65 int rowSum; int colSum;
66 poly dot;
67 for (i=1; i<=60; i++)
68 {
69     for (j=1; j<=60; j++)
70     {
71         dot=0;
72         for (k=1; k<=4; k++) { dot=dot+pts[i,k]*pts[j,k]; }
73         if (dot==0) { incidence[i,j]=1; }
74     }
75 }
76 for (i=1; i<=60; i++)
77 {
78     rowSum=0; colSum=0;
79     for (j=1; j<=60; j++)
80     {
81         rowSum=rowSum+incidence[i,j];
82         colSum=colSum+incidence[j,i];
83     }
84     if (rowSum!=15 || colSum!=15)
85     { ERROR("The point-plane incidence is not (60_15)"); }
86 }
87
88 // Five-point lines.
89 int reach; int maxReach=0; int fivePairs=0;
90 intmat fivePartners[60][1];
91 for (i=1; i<=60; i++)
92 {
93     for (j=i+1; j<=60; j++)
94     {
95         reach=0;
96         for (k=1; k<=60; k++)
97         {
98             if (areCollinear(pts,i,j,k)==1) { reach=reach+1; }
99         }
100         if (reach>maxReach) { maxReach=reach; }
101         if (reach==5)
102         {
103             fivePairs=fivePairs+1;
104             fivePartners[i,1]=fivePartners[i,1]+1;
105             fivePartners[j,1]=fivePartners[j,1]+1;
106         }
107     }
108 }
109 if (maxReach!=5) { ERROR("Maximum H4 line reach must be 5"); }
110 if (fivePairs!=720) { ERROR("Expected 720 pairs on 5-reach lines"); }
111 for (i=1; i<=60; i++)
112 {
113     // Six 5-point lines give 6*4 distinct partners at each point.
114     if (fivePartners[i,1]!=24)
115     { ERROR("Each point must lie on six 5-reach lines"); }
116 }
117
118 // Two 30-point covers.
119 intmat covered[60][1];
120 matrix span[4][4];
121 for (a=1; a<=6; a++)

```

```

122 {
123     for (k=3; k<=5; k++)
124     {
125         if (areCollinear(pts,cover1[a,1],cover1[a,2],cover1[a,k])==0)
126         { ERROR("A listed Z1 cover is not collinear"); }
127         if (areCollinear(pts,cover2[a,1],cover2[a,2],cover2[a,k])==0)
128         { ERROR("A listed Z2 cover is not collinear"); }
129     }
130     for (k=1; k<=5; k++)
131     {
132         covered[cover1[a,k],1]=covered[cover1[a,k],1]+1;
133         covered[cover2[a,k],1]=covered[cover2[a,k],1]+1;
134     }
135 }
136 for (i=1; i<=60; i++)
137 {
138     if (covered[i,1]!=1)
139     { ERROR("The two six-line covers must partition 60 points"); }
140 }
141
142 for (a=1; a<=6; a++)
143 {
144     for (b=a+1; b<=6; b++)
145     {
146         for (j=1; j<=4; j++)
147         {
148             span[1,j]=pts[cover1[a,1],j];
149             span[2,j]=pts[cover1[a,2],j];
150             span[3,j]=pts[cover1[b,1],j];
151             span[4,j]=pts[cover1[b,2],j];
152         }
153         if (det(span)==0)
154         { ERROR("The six Z1 cover lines must be mutually skew"); }
155         for (j=1; j<=4; j++)
156         {
157             span[1,j]=pts[cover2[a,1],j];
158             span[2,j]=pts[cover2[a,2],j];
159             span[3,j]=pts[cover2[b,1],j];
160             span[4,j]=pts[cover2[b,2],j];
161         }
162         if (det(span)==0)
163         { ERROR("The six Z2 cover lines must be mutually skew"); }
164     }
165 }
166
167 configuration_ok=1;
168 "H4 projective points: 60";
169 "point-plane incidence: (60_15)";
170 "5-reach lines: "+string(fivePairs div 10);
171 "5-reach lines through each point: 6";
172 "maximum reach: "+string(maxReach);
173 "six-line-cover partition: 30+30";
174 }

```

Listing A.8: Ring-changing implementation of the generic H_4 projection certificate

```

1 // Generic H4 certificate over  $Q(q_0, \dots, q_3)(u)$ .
2
3 // Degree-six forms through  $H_4$ .
4 ideal monomial6;
5 matrix pointEvaluation6[60][84];
6 int i; int j; int a; int b; int c; int d; int col;
7 for (a=0; a<=6; a++)
8 {
9   for (b=0; b<=6-a; b++)
10   {
11     for (c=0; c<=6-a-b; c++)
12     {
13       d=6-a-b-c;
14       col=col+1;
15       monomial6[col]=x0^a*x1^b*x2^c*x3^d;
16       for (i=1; i<=60; i++)
17       {
18         pointEvaluation6[i,col]=
19           L[4*(i-1)+1]^a*L[4*(i-1)+2]^b
20           *L[4*(i-1)+3]^c*L[4*(i-1)+4]^d;
21       }
22     }
23   }
24 }
25
26 if (rank(pointEvaluation6)!=60)
27 { ERROR("The degree-six H4 evaluation matrix must have rank 60"); }
28
29 module pointKernel6=syz(pointEvaluation6);
30 ideal basis6;
31 poly basisForm;
32 for (j=1; j<=ncols(pointKernel6); j++)
33 {
34   basisForm=0;
35   for (col=1; col<=84; col++)
36   {
37     basisForm=basisForm+pointKernel6[col,j]*monomial6[col];
38   }
39   basis6[j]=basisForm;
40 }
41
42 if (size(L)!=240)
43 { ERROR("H4 coordinate list must have 240 entries"); }
44 if (size(basis6)!=24)
45 { ERROR("Expected dim I(H4)_6=24"); }
46
47 // Introduce the generic center q.
48 ring A = (0,q0,q1,q2,q3),(u,x0,x1,x2,x3),(dp(1),dp(4));
49 qring R = std(u2-u-1);
50 execute("list L="+H4DATA+");");
51 ideal basis6=imap(Rs,basis6);
52 execute(read("h4/data/h4_half_grids.inc"));
53
54 proc nf(poly f)
55 {
56   return(reduce(f,std(0)));
57 }
58
59 proc planeLine(matrix p, int p1, int p2)
60 {
61   poly aa=nf(p[p1,2]*p[p2,3]-p[p1,3]*p[p2,2]);
62   poly bb=nf(p[p1,3]*p[p2,1]-p[p1,1]*p[p2,3]);
63   poly cc=nf(p[p1,1]*p[p2,2]-p[p1,2]*p[p2,1]);
64   return(nf(aa*x0+bb*x1+cc*x2));

```

```

65 }
66
67 proc planeEval(poly f, matrix p, int p1)
68 {
69     return(nf(subst(subst(subst(
70         f,x0,p[p1,1]),x1,p[p1,2]),x2,p[p1,3]))));
71 }
72
73 proc spatialEval(poly f, matrix p, int p1)
74 {
75     return(nf(subst(subst(subst(subst(
76         f,x0,p[p1,1]),x1,p[p1,2]),x2,p[p1,3]),x3,p[p1,4]))));
77 }
78
79 proc geometricDegree(poly f)
80 {
81     f=nf(f);
82     poly f0=subst(f,u,0);
83     poly f1=subst(f,u,1)-f0;
84     int d0=-1; int d1=-1;
85     if (f0!=0) { d0=deg(f0); }
86     if (f1!=0) { d1=deg(f1); }
87     if (d0>d1) { return(d0); }
88     return(d1);
89 }
90
91 proc verifyGenericProjection()
92 {
93     if (size(L)!=240 || size(basis6)!=24)
94     { ERROR("The full H4 input data or sextic basis is missing"); }
95
96     // Multiplicity six at q.
97     matrix derivatives[56][24];
98     int row; int aa; int bb; int cc; int dd; int k;
99     poly derivative;
100     for (aa=0; aa<=5; aa++)
101     {
102         for (bb=0; bb<=5-aa; bb++)
103         {
104             for (cc=0; cc<=5-aa-bb; cc++)
105             {
106                 dd=5-aa-bb-cc;
107                 row=row+1;
108                 for (j=1; j<=24; j++)
109                 {
110                     derivative=basis6[j];
111                     for (k=1; k<=aa; k++)
112                     { derivative=diff(derivative,x0); }
113                     for (k=1; k<=bb; k++)
114                     { derivative=diff(derivative,x1); }
115                     for (k=1; k<=cc; k++)
116                     { derivative=diff(derivative,x2); }
117                     for (k=1; k<=dd; k++)
118                     { derivative=diff(derivative,x3); }
119                     derivative=subst(subst(subst(subst(
120                         derivative,x0,q0),x1,q1),x2,q2),x3,q3);
121                     derivatives[row,j]=nf(derivative);
122                 }
123             }
124         }
125     }
126     if (row!=56)
127     { ERROR("Expected 56 generic fifth-derivative conditions"); }
128
129     // A cofactor kernel vector.

```

```

130 ideal coneCoefficients;
131 matrix maximalMinor[23][23];
132 int sourceColumn;
133 for (col=1; col<=24; col++)
134 {
135     sourceColumn=0;
136     for (j=1; j<=24; j++)
137     {
138         if (j!=col)
139         {
140             sourceColumn=sourceColumn+1;
141             for (i=1; i<=23; i++)
142             {
143                 maximalMinor[i,sourceColumn]=derivatives[i,j];
144             }
145         }
146     }
147     coneCoefficients[col]=nf(det(maximalMinor));
148     if ((col mod 2)==0)
149     { coneCoefficients[col]=-coneCoefficients[col]; }
150 }
151 if (coneCoefficients[24]==0)
152 { ERROR("The selected generic 23x23 minor vanished"); }
153
154 // A nonzero minor and kernel vector give rank 23.
155 poly condition;
156 for (i=1; i<=56; i++)
157 {
158     condition=0;
159     for (j=1; j<=24; j++)
160     {
161         condition=nf(condition+derivatives[i,j]*coneCoefficients[j]);
162     }
163     if (condition!=0)
164     { ERROR("The generic cone failed a fifth-derivative condition"); }
165 }
166
167 poly cone6=0;
168 for (j=1; j<=24; j++)
169 {
170     cone6=nf(cone6+coneCoefficients[j]*basis6[j]);
171 }
172 if (cone6==0 || geometricDegree(cone6)!=6)
173 { ERROR("The generic degree-six cone was not constructed"); }
174
175 // Projection and plane sextic.
176 matrix pts[60][4];
177 matrix image[60][3];
178 for (i=1; i<=60; i++)
179 {
180     for (j=1; j<=4; j++) { pts[i,j]=L[4*(i-1)+j]; }
181
182     image[i,1]=nf(q3*pts[i,1]-q0*pts[i,4]);
183     image[i,2]=nf(q3*pts[i,2]-q1*pts[i,4]);
184     image[i,3]=nf(q3*pts[i,3]-q2*pts[i,4]);
185
186     if (spatialEval(cone6,pts,i)!=0)
187     { ERROR("The generic cone must contain every H4 point"); }
188 }
189
190 int same;
191 for (i=1; i<=60; i++)
192 {
193     if (image[i,1]==0 && image[i,2]==0 && image[i,3]==0)
194     { ERROR("A projected H4 point is identically zero"); }

```

```

195
196     for (j=i+1; j<=60; j++)
197     {
198         same=1;
199         for (aa=1; aa<=3; aa++)
200         {
201             for (bb=aa+1; bb<=3; bb++)
202             {
203                 if (nf(image[i,aa]*image[j,bb]
204                     -image[i,bb]*image[j,aa])!=0)
205                     { same=0; }
206             }
207         }
208         if (same==1)
209             { ERROR("The generic projection is not injective"); }
210     }
211 }
212
213 poly sextic=nf(subst(cone6,x3,0));
214 if (sextic==0 || geometricDegree(sextic)!=6)
215 { ERROR("Restricting the cone did not produce a plane sextic"); }
216 for (i=1; i<=60; i++)
217 {
218     if (planeEval(sextic,image,i)!=0)
219     { ERROR("The sextic must vanish on all projected points"); }
220 }
221
222 // Two quintics from the half-grids.
223 poly g1=1; poly h1=1; poly g2=1; poly h2=1;
224 for (i=1; i<=5; i++)
225 {
226     g1=nf(g1*planeLine(image,grid1A[i,1],grid1A[i,2]));
227     h1=nf(h1*planeLine(image,grid1B[i,1],grid1B[i,2]));
228     g2=nf(g2*planeLine(image,grid2A[i,1],grid2A[i,2]));
229     h2=nf(h2*planeLine(image,grid2B[i,1],grid2B[i,2]));
230 }
231 poly quintic1=nf(
232     planeEval(h1,image,4)*g1-planeEval(g1,image,4)*h1);
233 poly quintic2=nf(
234     planeEval(h2,image,3)*g2-planeEval(g2,image,3)*h2);
235 if (quintic1==0 || geometricDegree(quintic1)!=5
236     || quintic2==0 || geometricDegree(quintic2)!=5)
237 { ERROR("The two half-grid quintics were not constructed"); }
238
239 for (i=1; i<=6; i++)
240 {
241     for (j=1; j<=5; j++)
242     {
243         if (planeEval(quintic1,image,cover1[i,j])!=0)
244             { ERROR("The first quintic failed its 30-point half-grid"); }
245         if (planeEval(quintic2,image,cover2[i,j])!=0)
246             { ERROR("The second quintic failed its 30-point half-grid"); }
247     }
248 }
249
250 poly decic=nf(quintic1*quintic2);
251 if (geometricDegree(decic)!=10)
252 { ERROR("The reducible curve must have degree ten"); }
253 for (i=1; i<=60; i++)
254 {
255     if (planeEval(decic,image,i)!=0)
256     { ERROR("The decic must vanish on all projected points"); }
257 }
258
259 // Coprimality via field norms.

```

```

260 sextic=nf(sextic);
261 decic=nf(decic);
262 poly sexticA=subst(sextic,u,0);
263 poly sexticB=subst(sextic,u,1)-sexticA;
264 poly decicA=subst(decic,u,0);
265 poly decicB=subst(decic,u,1)-decicA;
266 poly norm6=sexticA^2+sexticA*sexticB-sexticB^2;
267 poly norm10=decicA^2+decicA*decicB-decicB^2;
268 if (norm6==0 || norm10==0
269     || geometricDegree(norm6)!=12 || geometricDegree(norm10)!=20)
270 { ERROR("The norm construction failed"); }
271 poly normGcd=nf(gcd(norm6,norm10));
272 if (geometricDegree(normGcd)!=0)
273 { ERROR("The sextic and decic norms must be coprime"); }
274
275 "projected H4 points: 60";
276 "general projection: injective";
277 "dim I_6: 1";
278 "degrees: (6,10)";
279 "gcd/norm certificate: coprime";
280 "complete intersection degree: 60";
281 generic_ok=1;
282 }

```

Listing A.9: Coordinates of the 60 points of $\mathbf{H}_4$ in the fixed ordering

```

1 // The 60 H4 points over Q(u), where u^2-u-1=0.
2
3 string H4DATA =
4 "1,0,0,0, 0,1,0,0, 0,0,1,0, 0,0,0,1,"
5 "+1,1,1,1, 1,1,1,-1, 1,1,-1,1, 1,1,-1,-1,"
6 "+1,-1,1,1, 1,-1,1,-1, 1,-1,-1,1, 1,-1,-1,-1,"
7 "+0,u,u2,1, 0,u,u2,-1, 0,u,-u2,1, 0,u,-u2,-1,"
8 "+0,u2,1,u, 0,u2,1,-u, 0,u2,-1,u, 0,u2,-1,-u,"
9 "+0,1,u,u2, 0,1,u,-u2, 0,1,-u,u2, 0,1,-u,-u2,"
10 "+u,0,1,u2, u,0,1,-u2, u,0,-1,u2, u,0,-1,-u2,"
11 "+u2,0,u,1, u2,0,u,-1, u2,0,-u,1, u2,0,-u,-1,"
12 "+1,0,u2,u, 1,0,u2,-u, 1,0,-u2,u, 1,0,-u2,-u,"
13 "+u,u2,0,1, u,u2,0,-1, u,-u2,0,1, u,-u2,0,-1,"
14 "+u2,1,0,u, u2,1,0,-u, u2,-1,0,u, u2,-1,0,-u,"
15 "+1,u,0,u2, 1,u,0,-u2, 1,-u,0,u2, 1,-u,0,-u2,"
16 "+u,1,u2,0, u,1,-u2,0, u,-1,u2,0, u,-1,-u2,0,"
17 "+u2,u,1,0, u2,u,-1,0, u2,-u,1,0, u2,-u,-1,0,"
18 "+1,u2,u,0, 1,u2,-u,0, 1,-u2,u,0, 1,-u2,-u,0";

```

Listing A.10: Five-line families and six-line covers used for H_4

```

1 // Six-line covers and two rulings of each embedded 5-by-5 grid.
2
3 intmat cover1[6][5] =
4   1,29,32,33,36,
5   4,39,40,47,48,
6   5,13,31,41,52,
7   6,15,34,38,58,
8   7,14,35,37,57,
9   8,16,30,42,51;
10
11 intmat cover2[6][5] =
12   2,17,20,21,24,
13   3,49,50,53,54,
14   9,19,25,45,59,
15   10,23,27,43,55,
16   11,22,26,44,56,
17   12,18,28,46,60;
18
19 intmat grid1A[5][2] =
20   1,29, 5,13, 6,15, 7,14, 8,16;
21 intmat grid1B[5][2] =
22   1,30, 5,16, 6,14, 7,15, 8,13;
23 intmat grid2A[5][2] =
24   2,17, 9,19, 10,23, 11,22, 12,18;
25 intmat grid2B[5][2] =
26   2,18, 9,24, 10,20, 11,17, 12,21;

```

A.4 Cubic Weddle specialization

The implementation below is the exact rational specialization invoked by Listing 5.1. Listing A.12 supplies the rational coordinates used in each required base ring. The public entry point `verifyCubicWeddleSpecialization()` returns 1 only after every assertion has succeeded. Its complete printed dependencies are Listings A.1, A.11, and A.12.

Listing A.11: Implementation of the cubic Weddle specialization

```

1 // Rational Weddle specialization; the general theory is over C.
2
3 proc monomialCoefficient(poly f, poly mon)
4 {
5   while (f != 0)
6   {
7     if (leadmonom(f) == mon)
8     {
9       return(leadcoef(f));
10    }
11    f = f - lead(f);
12  }
13  return(0);
14 }
15
16 // Multiplication matrices modulo G.
17 proc reductionMatrices(ideal G)
18 {
19   ideal standardG = std(G);
20   ideal degree3Basis = kbase(standardG,3);

```

```

21 ideal degree2Basis = maxideal(2);
22 int targetDimension = size(degree3Basis);
23 matrix C0[targetDimension][10];
24 matrix C1[targetDimension][10];
25 matrix C2[targetDimension][10];
26 matrix C3[targetDimension][10];
27 int row; int column;
28 poly normalForm;
29
30 for (column = 1; column <= 10; column++)
31 {
32     normalForm = NF(x0*degree2Basis[column],standardG);
33     for (row = 1; row <= targetDimension; row++)
34     {
35         C0[row,column] =
36             monomialCoefficient(normalForm,degree3Basis[row]);
37     }
38
39     normalForm = NF(x1*degree2Basis[column],standardG);
40     for (row = 1; row <= targetDimension; row++)
41     {
42         C1[row,column] =
43             monomialCoefficient(normalForm,degree3Basis[row]);
44     }
45
46     normalForm = NF(x2*degree2Basis[column],standardG);
47     for (row = 1; row <= targetDimension; row++)
48     {
49         C2[row,column] =
50             monomialCoefficient(normalForm,degree3Basis[row]);
51     }
52
53     normalForm = NF(x3*degree2Basis[column],standardG);
54     for (row = 1; row <= targetDimension; row++)
55     {
56         C3[row,column] =
57             monomialCoefficient(normalForm,degree3Basis[row]);
58     }
59 }
60 return(list(C0,C1,C2,C3,targetDimension));
61 }
62
63 proc secantLineIdeal(matrix pointMatrix, int firstPoint, int secondPoint)
64 {
65     matrix lineMatrix[3][4];
66     lineMatrix[1,1] = q0;
67     lineMatrix[1,2] = q1;
68     lineMatrix[1,3] = q2;
69     lineMatrix[1,4] = q3;
70     int coordinate;
71     for (coordinate = 1; coordinate <= 4; coordinate++)
72     {
73         lineMatrix[2,coordinate] = pointMatrix[firstPoint,coordinate];
74         lineMatrix[3,coordinate] = pointMatrix[secondPoint,coordinate];
75     }
76     return(minor(lineMatrix,3));
77 }
78
79 proc verifyCubicWeddleSpecialization()
80 {
81     // Linear general position and quotients.
82
83     ring X = 0,(x0,x1,x2,x3),dp;
84     matrix points = cubicWeddlePointMatrix();
85

```

```

86 int lgpTests = 0;
87 int a; int b; int c; int d; int col;
88 matrix fourPoints[4][4];
89 for (a = 1; a <= 8; a++)
90 {
91     for (b = a + 1; b <= 9; b++)
92     {
93         for (c = b + 1; c <= 10; c++)
94         {
95             for (d = c + 1; d <= 11; d++)
96             {
97                 for (col = 1; col <= 4; col++)
98                 {
99                     fourPoints[1,col] = points[a,col];
100                     fourPoints[2,col] = points[b,col];
101                     fourPoints[3,col] = points[c,col];
102                     fourPoints[4,col] = points[d,col];
103                 }
104                 require(det(fourPoints) != 0,
105                     "four-point determinant vanished");
106                 lgpTests++;
107             }
108         }
109     }
110 }
111 require(lgpTests == 330,
112     "expected 330 four-point determinants");
113
114 list dualForms;
115 int pointIndex;
116 for (pointIndex = 1; pointIndex <= 11; pointIndex++)
117 {
118     dualForms[pointIndex] =
119         points[pointIndex,1]*x0 + points[pointIndex,2]*x1
120         + points[pointIndex,3]*x2 + points[pointIndex,4]*x3;
121 }
122
123 ideal powers9 =
124     dualForms[1]^3,dualForms[2]^3,dualForms[3]^3,
125     dualForms[4]^3,dualForms[5]^3,dualForms[6]^3,
126     dualForms[7]^3,dualForms[8]^3,dualForms[9]^3;
127 ideal powers10a = powers9,dualForms[10]^3;
128 ideal powers10b = powers9,dualForms[11]^3;
129 ideal powers11 = powers10a,dualForms[11]^3;
130
131 list data9 = reductionMatrices(powers9);
132 list data10a = reductionMatrices(powers10a);
133 list data10b = reductionMatrices(powers10b);
134 list data11 = reductionMatrices(powers11);
135
136 require(data9[5] == 11, "dim [A_9]_3 must be 11");
137 require(data10a[5] == 10, "dim [A_10a]_3 must be 10");
138 require(data10b[5] == 10, "dim [A_10b]_3 must be 10");
139 require(data11[5] == 9, "dim [A_11]_3 must be 9");
140
141 matrix C9_0 = data9[1];
142 matrix C9_1 = data9[2];
143 matrix C9_2 = data9[3];
144 matrix C9_3 = data9[4];
145 matrix C10a_0 = data10a[1];
146 matrix C10a_1 = data10a[2];
147 matrix C10a_2 = data10a[3];
148 matrix C10a_3 = data10a[4];
149 matrix C10b_0 = data10b[1];
150 matrix C10b_1 = data10b[2];

```

```

151 matrix C10b_2 = data10b[3];
152 matrix C10b_3 = data10b[4];
153 matrix C11_0 = data11[1];
154 matrix C11_1 = data11[2];
155 matrix C11_2 = data11[3];
156 matrix C11_3 = data11[4];
157
158 // Determinantal ideals in q-coordinates.
159
160 ring Q = 0,(q0,q1,q2,q3),dp;
161
162 matrix M9 =
163     q0*imap(X,C9_0) + q1*imap(X,C9_1)
164     + q2*imap(X,C9_2) + q3*imap(X,C9_3);
165 matrix M10a =
166     q0*imap(X,C10a_0) + q1*imap(X,C10a_1)
167     + q2*imap(X,C10a_2) + q3*imap(X,C10a_3);
168 matrix M10b =
169     q0*imap(X,C10b_0) + q1*imap(X,C10b_1)
170     + q2*imap(X,C10b_2) + q3*imap(X,C10b_3);
171 matrix M11 =
172     q0*imap(X,C11_0) + q1*imap(X,C11_1)
173     + q2*imap(X,C11_2) + q3*imap(X,C11_3);
174
175 require(nrows(M9) == 11 && ncols(M9) == 10,
176     "9-point multiplication matrix must be 11x10");
177 require(nrows(M10a) == 10 && ncols(M10a) == 10,
178     "first 10-point multiplication matrix must be 10x10");
179 require(nrows(M10b) == 10 && ncols(M10b) == 10,
180     "second 10-point multiplication matrix must be 10x10");
181 require(nrows(M11) == 9 && ncols(M11) == 10,
182     "11-point multiplication matrix must be 9x10");
183
184 ideal I45 = minor(M11,9);
185 ideal standardI45 = std(I45);
186 bigint degree45 = projectiveDegree(standardI45);
187 require(size(I45) == 10, "I_45 must have ten maximal minors");
188 require(dim(standardI45) == 2,
189     "I_45 must define a projective curve");
190 require(degree45 == 45, "11-point Weddle degree must be 45");
191 require(deg(I45[1]) == 9,
192     "the maximal minors of M11 must have degree 9");
193
194 // Hilbert--Burch data and degrees.
195
196 resolution resolution45 = mres(standardI45,0);
197 intmat betti45 = betti(resolution45);
198 require(ncols(betti45) == 3,
199     "unexpected Betti-table shape for I_45");
200 int bettiRow;
201 int bettiRank0 = 0;
202 int bettiRank1 = 0;
203 int bettiRank2 = 0;
204 for (bettiRow = 1; bettiRow <= nrows(betti45); bettiRow++)
205 {
206     bettiRank0 = bettiRank0 + betti45[bettiRow,1];
207     bettiRank1 = bettiRank1 + betti45[bettiRow,2];
208     bettiRank2 = bettiRank2 + betti45[bettiRow,3];
209 }
210 require(bettiRank0 == 1 && bettiRank1 == 10 && bettiRank2 == 9,
211     "Hilbert-Burch ranks for I_45 must be 1,10,9");
212
213 ideal I55 = minor(M9,10);
214 ideal standardI55 = std(I55);
215 bigint degree55 = projectiveDegree(standardI55);

```

```

216 require(size(I55) == 11, "I_55 must have eleven maximal minors");
217 require(dim(standardI55) == 2,
218   "I_55 must define a projective curve");
219 require(degree55 == 55, "9-point Weddle degree must be 55");
220
221 poly surfaceA = det(M10a);
222 poly surfaceB = det(M10b);
223 require(deg(surfaceA) == 10 && deg(surfaceB) == 10,
224   "both Weddle surfaces must have degree 10");
225 require(deg(gcd(surfaceA,surfaceB)) == 0,
226   "the two Weddle surfaces must be coprime");
227
228 ideal completeIntersection = surfaceA,surfaceB;
229 ideal standardCI = std(completeIntersection);
230 bigint degreeCI = projectiveDegree(standardCI);
231 require(dim(standardCI) == 2,
232   "the two surfaces must meet in a projective curve");
233 require(degreeCI == 100,
234   "the complete intersection must have degree 100");
235 require(size(NF(completeIntersection,standardI45)) == 0,
236   "the degree-45 curve must lie in both surfaces");
237 require(size(NF(completeIntersection,standardI55)) == 0,
238   "the degree-55 scheme must lie in both surfaces");
239 require(dim(std(I45 + I55)) == 1,
240   "the degree-45 and degree-55 curves must not share a curve component");
241 require(degree45 + degree55 == degreeCI,
242   "45+55 must equal the complete-intersection degree");
243
244 // Residual after the 36 secants.
245
246 matrix qPoints = cubicWeddlePointMatrix();
247
248 list secantIdeals;
249 int firstPoint; int secondPoint; int secantCount = 0;
250 for (firstPoint = 1; firstPoint <= 8; firstPoint++)
251 {
252   for (secondPoint = firstPoint + 1; secondPoint <= 9; secondPoint++)
253   {
254     secantCount++;
255     secantIdeals[secantCount] =
256       std(secantLineIdeal(qPoints,firstPoint,secondPoint));
257   }
258 }
259 require(secantCount == 36, "expected 36 secant-line ideals");
260
261 ideal I19 = I55;
262 int secantIndex;
263 for (secantIndex = 1; secantIndex <= 36; secantIndex++)
264 {
265   require(size(NF(I19,secantIdeals[secantIndex])) == 0,
266     "the current scheme does not contain a listed secant");
267   I19 = std(quotient(I19,secantIdeals[secantIndex]));
268 }
269
270 ideal standardI19 = std(I19);
271 bigint degree19 = projectiveDegree(standardI19);
272 require(dim(standardI19) == 2,
273   "the residual scheme must be a projective curve");
274 require(degree19 == 19, "the residual scheme must have degree 19");
275
276 ideal Isecant = quotient(I55,I19);
277 ideal standardIsecant = std(Isecant);
278 bigint degree36 = projectiveDegree(standardIsecant);
279 require(dim(standardIsecant) == 2,
280   "the degree-36 secant scheme must be a projective curve");

```

```

281 require(degree36 == 36, "the secant scheme must have degree 36");
282 require(size(NF(I55,standardIsecant)) == 0,
283   "all 36 secants must lie in the 9-point Weddle scheme");
284 for (secantIndex = 1; secantIndex <= 36; secantIndex++)
285 {
286   require(size(NF(Isecant,secantIdeals[secantIndex])) == 0,
287     "a listed secant is not contained in the degree-36 secant scheme");
288 }
289
290 ideal reconstructedI55 = intersect(Isecant,I19);
291 ideal standardReconstructedI55 = std(reconstructedI55);
292 require(size(NF(I55,standardReconstructedI55)) == 0,
293   "I_55 is not contained in I_secant intersect I_19");
294 require(size(NF(reconstructedI55,standardI55)) == 0,
295   "I_secant intersect I_19 is not contained in I_55");
296 require(degree36 + degree19 == degree55,
297   "36+19 must equal 55");
298
299 "points: 11 (linear general position)";
300 "11-point Weddle curve: degree 45";
301 "9-point Weddle scheme: degree 55";
302 "secant scheme: degree 36";
303 "residual scheme: degree 19";
304 "Bezout ledger: 100 = 45 + 55 = 45 + 36 + 19";
305
306 return(1);
307 }

```

Listing A.12: Rational point input for the cubic Weddle specialization

```

1 // Eleven rational points.
2
3 proc cubicWeddlePointMatrix()
4 {
5   matrix pointMatrix[11][4] =
6     1,0,0,0,
7     0,1,0,0,
8     0,0,1,0,
9     0,0,0,1,
10    1,5,1,3,
11    1,7,5,4,
12    1,4,7,7,
13    1,2,2,5,
14    1,3,6,1,
15    1,6,4,6,
16    1,1,3,2;
17   return(pointMatrix);
18 }

```

Bibliography

- [1] Henry Frederick Baker. *Principles of Geometry, Volume IV: Higher Geometry*. Cambridge University Press, 1925.
- [2] Cristiano Bocci and Brian Harbourne. “Comparing powers and symbolic powers of ideals”. In: *J. Algebraic Geom.* 19.3 (2010), pp. 399–417. ISSN: 1056-3911. DOI: 10.1090/S1056-3911-09-00530-X.
- [3] Cristiano Bocci and Brian Harbourne. “The resurgence of ideals of points and the containment problem”. In: *Proc. Amer. Math. Soc.* 138.4 (2010), pp. 1175–1190. ISSN: 0002-9939. DOI: 10.1090/S0002-9939-09-10108-9.
- [4] Arthur Cayley. “On the Conic of Five-Pointic Contact at Any Point of a Plane Curve”. In: *Philosophical Transactions of the Royal Society of London* 149 (1859), pp. 371–400. ISSN: 02610523. URL: <http://www.jstor.org/stable/108705>.
- [5] Luca Chiantini and Juan Migliore. “Sets of points which project to complete intersections, and unexpected cones”. In: *Trans. Am. Math. Soc.* 374.4 (2021), pp. 2581–2607. ISSN: 0002-9947. DOI: 10.1090/tran/8290.
- [6] Luca Chiantini et al. *Weddle schemes*. 2026. DOI: 10.48550/arXiv.2606.25060. arXiv: 2606.25060 [math.AG].
- [7] Hailong Dao et al. “Symbolic powers of ideals”. In: *Singularities and Foliations. Geometry, Topology and Applications*. Vol. 222. Springer Proc. Math. Stat. Springer, Cham, 2018, pp. 387–432.
- [8] Wolfram Decker et al. *SINGULAR 4.4.1p5 — A computer algebra system for polynomial computations*. <https://www.singular.uni-kl.de>. 2026.
- [9] Igor V. Dolgachev. *Classical Algebraic Geometry: A Modern View*. Cambridge University Press, 2012. ISBN: 978-1-107-01765-8. DOI: 10.1017/CB09781139084437.
- [10] Marcin Dumnicki, Tomasz Szemberg, and Halszka Tutaj-Gasińska. “Counterexamples to the $I^{(3)} \subset I^2$ containment”. In: *J. Algebra* 393 (2013), pp. 24–29. ISSN: 0021-8693. DOI: 10.1016/j.jalgebra.2013.06.039.
- [11] Marcin Dumnicki, Tomasz Szemberg, and Halszka Tutaj-Gasińska. “Symbolic powers of planar point configurations”. In: *J. Pure Appl. Algebra* 217.6 (2013), pp. 1026–1036. ISSN: 0022-4049. DOI: 10.1016/j.jpaa.2012.09.026.

- [12] Lawrence Ein, Robert Lazarsfeld, and Karen E. Smith. “Uniform bounds and symbolic powers on smooth varieties”. In: *Invent. Math.* 144.2 (2001), pp. 241–252. ISSN: 0020-9910. DOI: 10.1007/s002220100121.
- [13] David Eisenbud. *Commutative algebra*. Vol. 150. Graduate Texts in Mathematics. With a view toward algebraic geometry. Springer-Verlag, New York, 1995, pp. xvi+785. ISBN: 0-387-94268-8. DOI: 10.1007/978-1-4612-5350-1.
- [14] Arnold Emch. “On the Weddle surface and analogous loci”. In: *Transactions of the American Mathematical Society* 27.3 (1925), pp. 270–278. DOI: 10.1090/S0002-9947-1925-1501311-7.
- [15] William Fulton. *Algebraic Curves: An Introduction to Algebraic Geometry*. Advanced Book Classics. Reprint of the 1969 original. Redwood City, CA: Addison-Wesley Publishing Company, 1989, pp. xix+226. ISBN: 0-201-51010-3.
- [16] Remo Gattazzo. “Points of type 9 on an elliptic cubic”. In: *Rend. Sem. Mat. Univ. Padova* 61 (1979), pp. 285–301.
- [17] Hubert Grassmann et al. “Standard bases, syzygies and their implementation in SINGULAR”. In: *Beiträge zur Angewandten Analysis und Informatik: Helmut Brakhage zu Ehren*. Ed. by Eberhard Schock. Aachen: Shaker Verlag, 1994, pp. 69–96. ISBN: 3-8265-0015-6.
- [18] B. Harbourne et al. “Unexpected hypersurfaces and where to find them”. In: *Mich. Math. J.* 70.2 (2021), pp. 301–339. ISSN: 0026-2285. DOI: 10.1307/mmj/1593741748.
- [19] Brian Harbourne. *The atlas of Waldschmidt constants*. 2017. URL: <http://www.math.unl.edu/~bharbourne1/GammaFile.html> (visited on 12/01/2017).
- [20] Brian Harbourne and Craig Huneke. “Are symbolic powers highly evolved?” In: *J. Ramanujan Math. Soc.* 28A (2013), pp. 247–266. ISSN: 0970-1249.
- [21] Robin Hartshorne. *Algebraic Geometry*. Vol. 52. Graduate Texts in Mathematics. Springer-Verlag, New York–Heidelberg, 1977, pp. xvi+496. ISBN: 0-387-90244-9.
- [22] Melvin Hochster and Craig Huneke. “Comparison of symbolic and ordinary powers of ideals”. In: *Invent. Math.* 147.2 (2002), pp. 349–369. ISSN: 0020-9910. DOI: 10.1007/s002220100176.
- [23] Ronald William Henry Turnbull Hudson. *Kummer’s Quartic Surface*. Reprinted 1990. Cambridge University Press, 1905.
- [24] Łukasz Merta and Maciej Zięba. “Sextactic and type-9 points on the Fermat cubic and associated objects”. In: *Journal of Algebra* 662 (2025), pp. 502–513. DOI: 10.1016/j.jalgebra.2024.08.013. arXiv: 2406.13802 [math.AG].
- [25] Juan C. Migliore. *Introduction to liaison theory and deficiency modules*. Vol. 165. Progress in Mathematics. Birkhäuser Boston, Inc., Boston, MA, 1998, pp. xiv+215. ISBN: 0-8176-4027-4. DOI: 10.1007/978-1-4612-1794-7.

- [26] Joseph H. Silverman. *The Arithmetic of Elliptic Curves*. 2nd. Vol. 106. Graduate Texts in Mathematics. Springer, Dordrecht, 2009, pp. xx+513. ISBN: 978-0-387-09493-9. DOI: 10.1007/978-0-387-09494-6.
- [27] Tomasz Szemberg and Justyna Szpond. “Sextactic points on the Fermat cubic curve and arrangements of conics”. In: *Journal of Symbolic Computation* 120 (2024), p. 102228. ISSN: 0747-7171. DOI: 10.1016/j.jsc.2023.102228.
- [28] Justyna Szpond. *Fermat-type arrangements*. 2019. arXiv: 1909.04089. URL: <https://arxiv.org/abs/1909.04089>.
- [29] Maria Tombarkiewicz and Maciej Zięba. “On Yoshinaga’s arrangement of lines and the containment problem”. In: *Bull. Math. Soc. Sci. Math. Roumanie (N.S.)* 64 (112).3 (2021), pp. 207–215.
- [30] Thomas Weddle. “On the theorems in space analogous to those of Pascal and Brianchon in a plane”. In: *Cambridge and Dublin Math. J.* 5 (1850), pp. 58–69.
- [31] Paulina Wiśniewska and Maciej Zięba. “Generic projections of the H4 configuration of points in $\mathbb{P}^3$ ”. In: *Advances in Applied Mathematics* 142 (2023), p. 102432. DOI: 10.1016/j.aam.2022.102432.

Maciej Zięba
imię, nazwisko/ name

Kraków, dnia/date 22.07.2026

OŚWIADCZENIE/ DECLARATION

Świadomy/a odpowiedzialności oświadczam, że przedłożona rozprawa doktorska pt.:

Aware of my responsibility, I declare that the submitted dissertation entitled:

Geometry of Symmetric Configurations of Points in Projective Spaces

jest mojego autorstwa.

is my own creation.

Jednocześnie oświadczam, że ww. rozprawa nie narusza praw autorskich w rozumieniu Ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (Dz. U. z 2021 r. poz. 1062 z późn. zmianami) oraz dóbr osobistych chronionych prawem cywilnym.

At the same time, I declare that the aforementioned dissertation does not infringe upon copyrights within the meaning of the Act of 4 February 1994 on Copyright and Related Rights (Journal of Laws of 2021, item 1062, as amended) and personal rights protected by civil law.

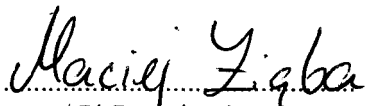
Przedłożona praca nie zawiera danych empirycznych ani też informacji, które uzyskałem/am w sposób niedozwolony.

The submitted work does not contain empirical data or information that I have obtained by unauthorised means.

Stwierdzam, iż przedstawiona praca w całości ani też w części nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z zakończeniem kształcenia w szkole doktorskiej.

I certify that the submitted work, in whole or in part, has not previously formed the basis of any other official procedure relating to the completion of training in doctoral school.

22.07.2026
Data/date


Podpis doktoranta/ PhD student's signature

Maciej Zięba
imię, nazwisko/ name

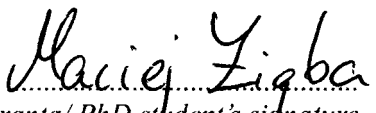
Kraków, dnia/date 22.07.2026

OŚWIADCZENIE / DECLARATION
o zgodności wersji elektronicznej z wydrukiem
on the conformity of the electronic version with the hard copy

Ja niżej podpisany/~~podpisana~~* oświadczam, że wersja elektroniczna jest zgodna z wersją papierową rozprawy doktorskiej.

I, the undersigned, declare that the electronic version is compatible with the paper version of the dissertation.*

22.07.2026
Data/date


Podpis doktoranta/ PhD student's signature

*Niewłaściwe skreślić/Delete as appropriate

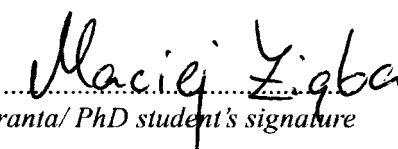
Maciej Zięba
imię, nazwisko/ name

Kraków, dnia/date 22.07.2026

OŚWIADCZENIE / DECLARATION

Ja niżej podpisany/~~podpisana~~* wyrażam zgodę/~~nie wyrażam zgody~~* na udostępnianie mojej rozprawy doktorskiej w wersji papierowej i elektronicznej dla celów naukowych i dydaktycznych.
I, the undersigned, consent/~~do not consent~~ to my dissertation being made available in hard copy and electronic format for research and teaching purposes.*

22.07.2026
Data/date


Podpis doktoranta/ PhD student's signature

*Niewłaściwe skreślić/Delete as appropriate