HOW FAR A CONIC OF $\mathrm{PG}(2,q)$ EXTENDS TO AN $(n,3)$-ARC: THE EXACT MAXIMUM FOR $q=3,5,7,11,13,17,19$

KOREA SUPERINTELLIGENCE LABS

Abstract. Let $\mathcal{O}$ be a conic of $\mathrm{PG}(2,q)$, q odd, and let $m(q)$ be the largest m for which some m points off $\mathcal{O}$ can be added to it so that the union is a $(q+1+m,3)$-arc, i.e. meets every line in at most three points. Dönmez, Akpınar and Pelen (arXiv:2608.18192) ask for $m(q)$ as a function of q; they prove $m(q)\geq 5$ for every odd prime power q and $m(q)\geq 6$ for every odd prime power $q\geq 11$, and report from the literature that $m(7)=m(9)=7$ and from a randomized search that $m(13)\geq 8$. We determine

$$m(3)=5,\quad m(5)=5,\quad m(7)=7,\quad m(11)=7,\quad m(13)=8,\quad m(17)=9,\quad m(19)=10.$$

Each value is a pair of machine-checked statements: an explicit witness set, verified by evaluating the arc condition inside the Lean 4 kernel, and an upper bound obtained from one exhaustive depth-first packing search over the q^2 points off the conic (257 437, 1 105 078, 18 657 443 and 55 313 459 search nodes at $q=11,13,17,19$), whose completeness is a proved lemma and whose run is a single compiled evaluation; no certificate is stored. Outside the formal development the same search gives $m(9)=7$ and $m(23)=11$. Consequences: $q=11$ is the first odd q at which a conic does not extend to an $(n,3)$-arc of the largest size $m_3(2,q)$ of its plane ($19<21$ and $22<23$); the five-point construction of the source is of maximum size exactly at $q=3,5$ and its six-point construction at none of $q=11,13,17,19$; and $m(q)/q$ decreases from 1.00 at $q=7$ to 0.48 at $q=23$, so that $m(q)=\lfloor (q+1)/2\rfloor$, which fits $q=17,19$, fails at $q=23$. No growth law is claimed.

1. Introduction

An $(n,3)$-*arc* of the Desarguesian projective plane $\mathrm{PG}(2,q)$ is a set of n points meeting every line in at most three points and some line in exactly three points; such arcs are the geometric form of the $[n,3,n-3]_q$ near-MDS codes (Dodunekov–Landgev [10]). For odd q a conic of $\mathrm{PG}(2,q)$ has $q+1$ points and meets every line in 0, 1 or 2 points, so it is an $(n,2)$-arc with room to spare, and one may ask how many further points it admits before some line carries four.

Dönmez, Akpınar and Pelen [1] construct, for every odd prime power q, $[q+6,3,q+3]$ near-MDS codes by adding five points to a conic (their Theorem 3.1), and $[q+7,3,q+4]$ codes by adding a sixth point for every odd prime power $q\geq 11$ (their Theorem 5.1 and Corollary 5.3; by their Proposition 5.2 admissible parameters exist for no odd $q\leq 9$). Their concluding section poses, as Problem 6.2:

> "Determine the maximal m, as a function of q, for which a conic of $\mathrm{PG}(2,q)$ can be extended by m points to a $(q+1+m,3)$-arc."

The same problem statement records what is known: the character-sum method of their Proposition 5.2 gives $m\gg\log q$; "$m\leq q+1$ for odd q: the trivial bound gives $n\leq 2q+3$, and equality would make the arc a maximal $\{2q+3;3\}$-arc, which does not exist in $\mathrm{PG}(2,q)$ for odd q" (citing Ball–Blokhuis–Mazzocca [8]); "Conjecturally $m\leq q-2$ for $q\geq 8$: the NMDS length conjecture, as reported in [14], asserts that the maximal length of an $[n,3,n-3]$ NMDS code is at most $2q-1$ for $q\geq 8$" (their [14] is Lu–Zhou [9]); and, verbatim,

> "Randomized computer search suggests the truth is indeed linear in q: for $q=7$ and $q=9$ a conic extends to $(n,3)$-arcs of the maximal sizes $n=15=2q+1$ and $n=17=2q-1$ [15, 16] respectively, the latter attaining the conjectural bound, and for $q=13$ to a $(22,3)$-arc."

(Their [15, 16] are Marcugini–Milani–Pambianco [2, 3].) So the source proves $m(q)\geq 5$ for all odd q and $m(q)\geq 6$ for odd $q\geq 11$, reports $m(7)=m(9)=7$ and $m(13)\geq 8$, names no value at $q=11$, and leaves $q = 13$ open; none of its twenty-four numbered statements determines $m(q)$ for any q. All section, theorem and equation numbers of [1] quoted in this paper are those of the e-print compiled from the arXiv source of version 1 (18 pages; the numbers were read from the compiled `.aux` file and the quotations from the text of that compilation).

Results. Writing $m(q)$ for the maximum in Problem 6.2, we prove

q	3	5	7	9*	11	13	17	19	23*
$m(q)$	5	5	7	7	**7**	**8**	**9**	**10**	**11**
$n = q + 1 + m(q)$	9	11	15	17	**19**	**22**	**27**	**30**	**35**

Bold entries are new. The entries without a star are theorems verified in Lean 4 (Section 4); the two starred entries are computed by the same exhaustive search outside the formal development and are reported as such (Section 5). The values $m(7) = 7$ and $m(9) = 7$ reproduce the source's reported values and were the reproduction gate for everything else; $m(13) = 8$ upgrades the source's $(22, 3)$-arc to a largest one; $m(11)$, $m(17)$, $m(19)$ are values the source does not address.

Each value rests on two statements of different kinds. The lower bound is an explicit set T of m points, listed in Appendix A, for which the arc condition on $\mathcal{O} \cup T$ is evaluated by the Lean 4 kernel (`decide`); these statements depend on no axiom beyond `propext` and `Quot.sound` (and `Classical.choice` once, at $q = 3$). The upper bound is a single exhaustive depth-first search over the q^2 points off the conic with per-line capacity pruning, run once as compiled code (`native_decide`) at target $m+1$; the search is a plain enumeration with no symmetry reduction, and the only fact used about it is the proved lemma that a negative answer implies that no feasible set of the target size exists (Section 3). The arithmetic of the plane is verified separately at $q = 7$ (kernel only) and $q = 11$: the coordinate lists are a projective plane of order q, the conic is the $q + 1$ points of $X_1^2 = X_0X_2$ inside it, and the candidate list is exactly the complement (Section 6).

Consequences. Three things follow from the table, none of them a growth law.

The conic stops reaching the plane maximum at $q = 11$. Write $m_3(2, q)$ for the largest n for which $\mathrm{PG}(2, q)$ contains an $(n, 3)$-arc. For $q \leq 9$ the conic extends to an arc of that size (the source quotes this for $q = 7, 9$; see Remark 5.1). At $q = 11$ it does not: $19 < 21 = m_3(2, 11)$ (Marcugini–Milani–Pambianco [4]), and at $q = 13$ again $22 < 23 = m_3(2, 13)$ [5]. The extension halves of these inequalities are theorems here; the values 21 and 23 are taken from the literature.

The source's constructions are never of maximum size where its sixth-point theorem applies. The five-point construction of [1, Theorem 3.1] is of maximum size exactly at $q = 3$ and $q = 5$, where $m(q) = 5$, and at no other q determined here; the six-point construction of [1, Theorem 5.1], defined for odd $q \geq 11$, is of maximum size at none of $q = 11, 13, 17, 19$ (nor at $q = 23$), since $m(q) \geq 7$ there.

The growth is slow and the data do not settle its rate. The ratio $m(q)/q$ takes the values 1.00, 0.78, 0.64, 0.62, 0.53, 0.53, 0.48 at $q = 7, 9, 11, 13, 17, 19, 23$, far below the conjectural $m \leq q-2$; $m(q) = \lfloor (q + 1)/2 \rfloor$ fits $q = 17, 19$ and fails at $q = 23$ ($11 \neq 12$). Whether $m(q)/q$ has a positive limit is open, and nothing here bears on it beyond the table (Remark 5.5).

Prior work on the values. We found no determination of $m(q)$ for any $q \geq 11$ in print. The complete $(k, 3)$-arcs of $\mathrm{PG}(2, q)$ are classified for $q \leq 13$ by Coolsaet and Sticker [7]; since a largest extension of a conic is a complete $(n, 3)$-arc, the values $m(11)$ and $m(13)$ are in principle extractable from that classification. They are not stated in it: we have read that paper in full — in the copy deposited at Ghent University, Wiley's version of record not being reachable to us — and its tables are indexed by the size of the arc and by its automorphism group, with no arc identified there as containing a conic. We found the values stated nowhere else, and the source of August 2026 poses the question as open. We record this so that a reader with the classification's arcs in hand can cross-check Theorems 4.4 and 4.5 against them rather than re-derive them. For $q = 17, 19$ no classification of $(n, 3)$-arcs of the plane exists; there not even $m_3(2, q)$ is known [6]. For $q = 7$, Innamorati and Zannetti [11] give a proof without a computer that the $(15, 3)$-arc of $\mathrm{PG}(2, 7)$ is unique up to projectivity — the uniqueness itself they credit to Marcugini, Milani and Pambianco — and describe that arc as the union of a conic with the seven points of a complete quadrangle, namely its four vertices together with its three diagonal points, the nine lines joining those seven points being external to the conic. That is $8 + 7 = 15$, so their description is the structural form of $m(7) = 7$.

Method, in one paragraph. Because a line meets $\mathcal{O}$ in at most two points, the arc condition on $\mathcal{O} \cup T$ is a capacity on T per line: at most one point of T on a secant, two on a tangent, three on an external line. So $m(q)$ is the optimum of an exact set-packing problem with q^2 items and $q^2 + q + 1$ constraints, and a depth-first search that adds candidate points in a fixed order and discards, after each addition, every candidate lying on a line that has just become saturated, decides whether $m + 1$ points can be packed. The values were first computed twice in C, in two independently generated coordinate models of the plane, and at $q \leq 13$ also by a CDCL solver on a CNF encoding; the CDCL route was then abandoned for the certificate because it does not scale (Remark 4.9). The certificate is the Lean 4 development described in Sections 3 and 7: the search procedure is written in Lean, compiled to a shared library, and evaluated once per q by `native_decide`, and the lemma converting its negative answer into a bound on every extension is proved in the kernel. The formal development takes q prime (its field is $\mathbb{Z}/q\mathbb{Z}$); $q = 9$ is therefore computed outside it, in an explicit model of $\mathbb{F}_9$, and $q = 23$ is outside it for cost (Section 5).

2. Preliminaries

Throughout, q is an odd prime power, $\mathbb{F}_q$ the field with q elements and $\mathrm{PG}(2, q)$ the projective plane over it. Points and lines are nonzero triples up to scalars; the point $P = (P_0, P_1, P_2)$ lies on the line $L = (L_0, L_1, L_2)$ iff $P_0L_0 + P_1L_1 + P_2L_2 = 0$. Every point has a unique *normalised* representative of one of the forms

$$(1, y, z), \qquad (0, 1, z), \qquad (0, 0, 1), \qquad y, z \in \mathbb{F}_q, \tag{1}$$

and so does every line; the lists of these $q^2 + q + 1$ triples are the coordinate model used by the formal development, with points and lines drawn from the same list.

Definition 2.1. The conic is $\mathcal{O} = \mathcal{O}_q = \{X_1^2 = X_0X_2\}$, i.e. in normalised coordinates

$$\mathcal{O} = \{(1, y, y^2) : y \in \mathbb{F}_q\} \cup \{(0, 0, 1)\}.$$

This is the conic $\{(x^2, x, 1) : x \in \mathbb{F}_q\} \cup \{(1, 0, 0)\}$ of [1, eq. (2.3)] re-normalised ($x \neq 0$ gives $(1, x^{-1}, x^{-2})$, $x = 0$ gives $(0, 0, 1)$, and $(1, 0, 0)$ is the point with $y = 0$). For odd q every conic is projectively equivalent to it, so nothing below depends on the choice.

Definition 2.2. A finite set $T \subseteq \mathrm{PG}(2, q) \setminus \mathcal{O}$ *extends* $\mathcal{O}$ if every line of $\mathrm{PG}(2, q)$ carries at most three points of $\mathcal{O} \cup T$. Let

$$m(q) = \max\{|T| : T \text{ extends } \mathcal{O}\}.$$

The maximum exists since T ranges over subsets of a finite set and $T = \emptyset$ extends $\mathcal{O}$. Definition 2.2 is the "at most three" half of the $(n, 3)$-arc condition on $\mathcal{O} \cup T$; the "some line in exactly three" half is automatic once $T \neq \emptyset$:

Lemma 2.3. *If $T \neq \emptyset$ extends $\mathcal{O}$ then $\mathcal{O} \cup T$ is a $(q + 1 + |T|, 3)$-arc; indeed every secant of $\mathcal{O}$ through a point of T carries exactly three points of $\mathcal{O} \cup T$.*

Proof. Let $P \in T$. The $q + 1$ lines through P partition the $q + 1$ points of $\mathcal{O}$, and for odd q a point off a conic lies on 0 or 2 of its tangents — in the words of [1, §2.2], which cites Hirschfeld [12] for it, "[a] point $P \notin \mathcal{O}$ is *exterior* if it lies on two tangents and *interior* if it lies on none" — so at most two of those lines carry one conic point and the others carry 0 or 2; since $q + 1 \geq 4$, some line through P is a secant. It carries the two conic points and P, hence at least three points of $\mathcal{O} \cup T$, and at most three by hypothesis. Consequently $m(q)$ is exactly the quantity of Problem 6.2 of [1]. $\square$

This lemma is not part of the formal development: each witness theorem below exhibits its trisecant explicitly instead, so that the formal statement is the full arc condition.

Lemma 2.4 (line capacities)**.** *Let T extend $\mathcal{O}$ and let L be a line. Then $|L \cap T| \le 3 - |L \cap \mathcal{O}|$: at most 1 on each of the $\binom{q+1}{2}$ secants, at most 2 on each of the $q+1$ tangents, at most 3 on each of the $q(q-1)/2$ external lines. Conversely $T \subseteq \mathrm{PG}(2,q) \setminus \mathcal{O}$ extends $\mathcal{O}$ if it satisfies these capacities.*

Proof. $|L \cap (\mathcal{O} \cup T)| = |L \cap \mathcal{O}| + |L \cap T|$ because T is disjoint from $\mathcal{O}$. The line counts are the standard ones for a conic in odd characteristic. □

So $m(q)$ is the optimum of an exact set-packing problem with q^2 items (the points off $\mathcal{O}$) and q^2+q+1 capacity constraints. The capacity distribution was checked against these formulas at every q considered.

3. The search and what is proved about it

The formal development separates the search from the geometry.

3.1. **The abstract packing search.** An *instance* I consists of a number n of candidate points, a list of lines with capacities caps_l, an incidence table $\mathrm{On}(l,x)$, and for each candidate p the list $\mathrm{plns}(p)$ of lines through it. A list T of candidate indices is *feasible* when every line l carries at most caps_l members of T. The search `go` takes a target t, a fuel, the chosen list T and the remaining candidates A (increasing indices) and returns a Boolean:

> if $t \le |T|$ return `true`; if $A = [\,]$ return `false`; if the fuel is 0 return `true`; otherwise, with $A = p :: A'$ and fuel $f+1$, return $\texttt{go}(t, f, p :: T, A' \restriction \mathrm{keep}) \ \vee\ \texttt{go}(t, f, T, A')$,

where $A' \restriction \mathrm{keep}$ drops from A' every candidate lying on a line through p that is saturated by $p :: T$ (`full`, `keep` in Appendix A). The first branch takes p, the second skips it. There is no symmetry reduction, no ordering heuristic beyond the fixed candidate order of (1), and no memoisation. The fuel is cosmetic: A strictly shrinks in both branches, so with fuel n the out-of-fuel branch is never taken; it makes the recursion structural.

The only statement proved about `go` is the direction that carries information.

Theorem 3.1 (completeness of the search)**.** *Let I be an instance with n candidates and $t \in \mathbb{N}$. If $\texttt{go}(t, n, [\,], [0, \ldots, n-1]) = \texttt{false}$, then every feasible list S of candidate indices, listed in increasing order, has $|S| < t$.*

Proof sketch. This is `no_feasible_of_search`, from the general form `go_eq_false`: for all fuel f, chosen T, remaining A and every sublist S of A with $S^{\mathrm{rev}} +\!\!+\, T$ feasible, $\texttt{go}(t, f, T, A) = \texttt{false}$ implies $|S| + |T| < t$; by induction on f. In the inductive step with $A = p :: A'$, both branches returned `false`. If S avoids p it is a sublist of A' and the skip branch applies. If $S = p :: S'$, the one real step is that no $x \in S'$ is pruned: were x on a line l through p saturated by $p :: T$, then l would carry $\mathrm{caps}_l + 1$ members of $S'^{\mathrm{rev}} +\!\!+\, (p :: T)$, contradicting feasibility; so S' is a sublist of the filtered list and the take branch applies. The proof depends only on `propext` and `Quot.sound`. □

Nothing in this theorem mentions a plane, so no geometric error can leak into the search's completeness; and since only the `false` direction is proved, a bug in the pruning could make the search return `true` wrongly (which proves nothing) but never `false` wrongly.

3.2. **The instance for $\mathrm{PG}(2,q)$ and the bridge.** For prime q the instance I_q has as candidates the q^2 normalised triples off $\mathcal{O}$ in the order of (1), as lines all q^2+q+1 normalised triples, capacity $3 - |L \cap \mathcal{O}|$ for the line L, and incidence $P_0L_0 + P_1L_1 + P_2L_2 \equiv 0 \pmod q$; this is computed in natural-number arithmetic, without any library, so that the module compiles to a shared library. Write $\texttt{searchN}(q,t)$ for `go` on I_q at target t from the empty set.

The geometric side works over $\mathbb{Z}/q\mathbb{Z}$: a list T of triples satisfies $\texttt{ExtendsConic}\, q\, T$ when T has no repeated entry, every entry is one of the q^2 normalised points off the conic, and for every normalised line L the number of entries of the list $\mathcal{O} +\!\!+\, T$ on L is at most three. This is Definition 2.2 for the set of entries of T, given that the conic list is duplicate-free and disjoint from the candidates (both verified at $q = 7, 11$ in Section 6, and true by construction for every q).

Theorem 3.2 (bridge)**.** *Let q be prime and $t \in \mathbb{N}$. If the candidate list has no repeated entry and* $\texttt{searchN}(q, t) = \texttt{false}$*, then every list T with* $\texttt{ExtendsConic}\, q\, T$ *has $|T| < t$.*

Proof sketch. This is `le_of_search`. Given T, take the index list S of those candidates that lie in T, in increasing order; it is a sublist of $[0, \dots, q^2 - 1]$ by construction, its length is $|T|$ because T and the filtered candidate list are permutations of each other (both duplicate-free, each contained in the other), and for each line the count of S transfers to the count of T by the incidence identity $\texttt{onN} = \texttt{OnLine}$ (a cast lemma $a \bmod q = 0 \iff (a : \mathbb{Z}/q\mathbb{Z}) = 0$) and the capacity identity. Then Theorem 3.1 gives $|S| < t$. The proof depends on `propext`, `Quot.sound` and `Classical.choice`. □

Each upper bound below is Theorem 3.2 at $t = m(q) + 1$ with both hypotheses discharged by evaluation: the search by `native_decide` (compiled), and the duplicate-freeness of the candidate list by `native_decide` as well, except at $q = 11$ where it is a kernel `decide`. Each such theorem therefore carries, besides `propext`, `Quot.sound`, `Classical.choice`, two axioms of the form `<decl>._native.native_decide.ax_1_k` asserting what the compiled evaluations returned (one at $q = 11$); no other native evaluation occurs in the proofs of the values.

4. The values

In each statement T is the explicit list given in Appendix A (points written as normalised triples over $\mathbb{Z}/q\mathbb{Z}$), "T extends $\mathcal{O}_q$" is $\texttt{ExtendsConic}\, q\, T$, and "$L$ carries exactly three points" is the count of entries of $\mathcal{O}_q \mathbin{+\!\!+} T$ on L being 3. Part (c) of each is the statement that m is the greatest element of $\{|S| : S \text{ extends } \mathcal{O}_q\}$, which is $m(q) = m$.

4.1. **The small cases and the reproduction gate.**

Proposition 4.1 ($q = 3$)**.** *(a) The five-point list T_3 extends $\mathcal{O}_3$, $|T_3| = 5$, and the line $(1, 0, 0)$ carries exactly three points of $\mathcal{O}_3 \cup T_3$. (b) Every S extending $\mathcal{O}_3$ has $|S| \le 5$. (c) $m(3) = 5$.*

Proposition 4.2 ($q = 5$)**.** *(a) The five-point list T_5 extends $\mathcal{O}_5$, $|T_5| = 5$, and the line $(1, 0, 4)$ carries exactly three points of $\mathcal{O}_5 \cup T_5$. (b) Every S extending $\mathcal{O}_5$ has $|S| \le 5$. (c) $m(5) = 5$.*

Proposition 4.3 ($q = 7$)**.** *(a) The seven-point list T_7 extends $\mathcal{O}_7$, $|T_7| = 7$, and the line $(1, 0, 3)$ carries exactly three points of $\mathcal{O}_7 \cup T_7$. (b) Every S extending $\mathcal{O}_7$ has $|S| \le 7$. (c) $m(7) = 7$.*

Proof of Propositions 4.1–4.3. Part (a) is the evaluation of the arc condition on the listed points by the kernel. Part (b) is Theorem 3.2 at targets 6, 6, 8, the exhaustive search over the 9, 25, 49 points off the conic returning `false` (14, 541 and 3 665 nodes in the sense of Table 1). Part (c) combines them. □

Propositions 4.1 and 4.2 are routine: the arcs have $n = 9$ and $n = 11$ points, the first being the affine plane $\mathrm{AG}(2, 3)$ (the trivial bound $2q + 3$, attained since $3 \mid q$). Proposition 4.3 reproduces the source's reported value $n = 15 = 2q + 1$ and, with the computation of $m(9)$ in Section 5, was the gate before any new value was claimed.

4.2. **New values.**

Theorem 4.4 ($q = 11$)**.** *(a) The seven-point list*

$$T_{11} = [(1, 0, 1), (1, 1, 3), (1, 2, 0), (1, 3, 6), (1, 6, 0), (1, 7, 4), (1, 9, 6)]$$

extends $\mathcal{O}_{11}$, $|T_{11}| = 7$, and the line $(1, 0, 7)$ carries exactly three points of $\mathcal{O}_{11} \cup T_{11}$; so $\mathcal{O}_{11} \cup T_{11}$ is a $(19, 3)$-arc of $\mathrm{PG}(2, 11)$*. (b) Every S extending $\mathcal{O}_{11}$ has $|S| \le 7$: no eight points extend the conic of* $\mathrm{PG}(2, 11)$ *to a $(20, 3)$-arc. (c) $m(11) = 7$.*

Theorem 4.5 ($q = 13$)**.** *(a) The eight-point list*

$$T_{13} = [(1, 0, 1), (1, 1, 0), (1, 2, 7), (1, 3, 7), (1, 6, 0), (1, 7, 8), (1, 11, 3), (1, 12, 5)]$$

extends $\mathcal{O}_{13}$, $|T_{13}| = 8$, and the line $(1, 0, 4)$ carries exactly three points of $\mathcal{O}_{13} \cup T_{13}$; so $\mathcal{O}_{13} \cup T_{13}$ is a $(22, 3)$-arc of $\mathrm{PG}(2, 13)$*. (b) Every S extending $\mathcal{O}_{13}$ has $|S| \le 8$: no nine points extend the conic of* $\mathrm{PG}(2, 13)$ *to a $(23, 3)$-arc. (c) $m(13) = 8$.*

Theorem 4.6 ($q = 17$)**.** *(a) The nine-point list*

$$T_{17} = [\,(1,0,1),(1,1,0),(1,2,0),(1,3,2),(1,4,3),\\ (1,8,12),(1,9,15),(1,13,8),(1,15,3)\,]$$

extends $\mathcal{O}_{17}$, $|T_{17}| = 9$, and the line $(1,0,2)$ carries exactly three points of $\mathcal{O}_{17} \cup T_{17}$; so $\mathcal{O}_{17} \cup T_{17}$ is a $(27,3)$-arc of $\mathrm{PG}(2,17)$*. (b) Every S extending $\mathcal{O}_{17}$ has $|S| \le 9$: no ten points extend the conic of* $\mathrm{PG}(2,17)$ *to a $(28,3)$-arc. (c) $m(17) = 9$.*

Theorem 4.7 ($q = 19$)**.** *(a) The ten-point list*

$$T_{19} = [\,(1,0,1),(1,1,3),(1,2,7),(1,5,10),(1,7,4),\\ (1,8,6),(1,10,9),(1,11,17),(1,17,10),(1,18,15)\,]$$

extends $\mathcal{O}_{19}$, $|T_{19}| = 10$, and the line $(1,0,2)$ carries exactly three points of $\mathcal{O}_{19} \cup T_{19}$; so $\mathcal{O}_{19} \cup T_{19}$ is a $(30,3)$-arc of $\mathrm{PG}(2,19)$*. (b) Every S extending $\mathcal{O}_{19}$ has $|S| \le 10$: no eleven points extend the conic of* $\mathrm{PG}(2,19)$ *to a $(31,3)$-arc. (c) $m(19) = 10$.*

Proof of Theorems 4.4–4.7. Part (a): the Boolean form of `ExtendsConic` is evaluated on the listed points by the kernel — `decide` at $q = 11, 13$ and `decide +kernel` at $q = 17, 19$, where the elaborator's own reduction exceeded its budget after 4.5 minutes while the kernel does not — together with the count on the named line. These evaluations rest on no axiom beyond `propext` and `Quot.sound`.

Part (b) is Theorem 3.2 at target $m + 1$, i.e. $8, 9, 10, 11$. The search over the 121, 169, 289, 361 candidates returns `false`, with 257 437, 1 105 078, 18 657 443 and 55 313 459 nodes respectively in the sense of Table 1. It is run once, as compiled code loaded from a shared library, inside a single `native_decide`; nothing is stored. The compiled evaluation was timed at 1.68 s at $q = 11$ and 9.3 s at $q = 13$; at $q = 17$ and $q = 19$ it was not timed apart from its module, which checks in 302 s and 1 005 s (Section 7).

Part (c) combines (a) and (b). □

Remark 4.8 (what the source knew)*.* At $q = 11$ the source's proved bound is $m(11) \ge 6$ [1, Cor. 5.3], and its Problem 6.1 exhibits five points, "$(7,0),(0,7),(9,10),(7,4),(1)$", extending the conic (2.3) to a $(17,3)$-arc. We checked that set directly (outside the formal development, in the source's own coordinates with $(x,y) \leftrightarrow (x,y,1)$ and $(a) \leftrightarrow (a,1,0)$): 17 points, every line carrying at most three, 25 trisecants. At $q = 13$ the source's randomized search found a $(22,3)$-arc and claimed no maximality; Theorem 4.5(a) is such an arc made explicit and Theorem 4.5(b) is its maximality. For $q = 17, 19$ the source's bound is $m \ge 6$ and no other value is discussed.

4.3. **The searches.** Table 1 collects the exhaustions. A *node* there is one partial set visited by the C implementation, which runs the same search with one addition: it abandons a branch as soon as the candidates still available cannot reach the target. That cut-off is sound for the decision question and leaves the answer unchanged, but `go` does not have it and so traverses a larger tree; the counts are reported as a size measure of the exhaustion and no proof depends on them.

Remark 4.9 (why not a SAT certificate)*.* The packing was also encoded in CNF (a clause forbidding each over-capacity subset of each line, and a sequential counter for "at least k points") and given to a CDCL solver (`cadical`). At $q \le 13$ it agrees with the search in every cell — satisfiable at $k = m(q)$, unsatisfiable at $k = m(q) + 1$ in 0.03 s, 4.9 s and 31.4 s for $q = 7, 11, 13$ — and at $q = 17, 19$ it confirms the lower bounds in 0.25 s and 1.13 s. But the unsatisfiable instance at $q = 17$ had not finished after 946 s and was stopped, where the depth-first search takes about two compiled minutes (Table 1: 4.6 s in C, times the measured factor). The per-line capacity propagation is what the depth-first search exploits and the clause learning does not, so a certificate route through LRAT proofs would have been strictly more expensive here than a single compiled run of a procedure whose completeness is proved; the CNF was kept as a cross-check, not as the proof, and no solver certificate is stored.

5. Consequences, and the two values outside the formal development

Remark 5.1 ($q = 11$ is the first miss of the plane maximum)*.* Let $m_3(2,q)$ be the largest size of an $(n,3)$-arc of $\mathrm{PG}(2,q)$. For $q = 3$ the trivial bound $2q + 3 = 9$ is attained (Proposition 4.1); for $q = 5$,

q	candidates	target	nodes (C)	C, `gcc -O2`	certificate
3	9	6	14	—	Lean, compiled
5	25	6	541	—	Lean, compiled
7	49	8	3 665	—	Lean, compiled
9	81	8	—	—	C only ($\mathbb{F}_9$ model)
11	121	8	257 437	0.07 s	Lean, compiled (1.68 s)
13	169	9	1 105 078	0.30 s	Lean, compiled (9.3 s)
17	289	10	18 657 443	4.6 s	Lean, compiled
19	361	11	55 313 459	16.2 s	Lean, compiled
23	529	12	533 924 978	—	C only

TABLE 1. The exhaustive decision searches at target $m(q) + 1$; every one of them answers "no". "C only" marks the two values outside the formal development. The node counts are those of the C search in the candidate order of (1); no count is given for $q = 9$, whose search runs in a separate $\mathbb{F}_9$ model with its own point enumeration and whose count is therefore not comparable with the rest of the column. Compiled Lean is 24–31 times slower than `gcc -O2` on the identical procedure (1.68 s vs. 0.07 s at $q = 11$; 9.3 s vs. 0.30 s at $q = 13$).

$m_3(2, 5) = 11$ [6] is attained by Proposition 4.2; for $q = 7, 9$ the source itself quotes $n = 15$ and $n = 17$ as "the maximal sizes", attained by Proposition 4.3 and by $m(9) = 7$ below. So for every odd $q \leq 9$ a conic extends to an $(n, 3)$-arc of the largest size in its plane. At $q = 11$ it does not: Theorem 4.4 gives 19, while $m_3(2, 11) = 21$. That is the result of Marcugini–Milani–Pambianco [4], who determine the largest size of an $(n, 3)$-arc of PG$(2, 11)$ and find it to be 21, with exactly two such arcs up to equivalence. At $q = 13$, Theorem 4.5 gives 22 while $m_3(2, 13) = 23$: the same authors [5] show that the largest *complete* $(n, 3)$-arc of PG$(2, 13)$ has 23 points, and a largest $(n, 3)$-arc is complete, so the two maxima agree. Both values are recorded again in [7, §1] and in [6]. These two literature values are not part of the formal development; the extension halves (19 and 22) are, and do not depend on them. Exact values of $m_3(2, q)$ are known for the prime powers $q \leq 16$ and, as of the last revision of Ball's table of bounds [6], for no larger q; so the comparison cannot be continued to $q = 17, 19$, where that table gives only $28 \leq m_3(2, 17) \leq 33$ and $31 \leq m_3(2, 19) \leq 39$.

Remark 5.2 (the source's constructions). The five points of [1, Theorem 3.1] give $m(q) \geq 5$ for every odd prime power q, and by Propositions 4.1 and 4.2 this is the maximum at $q = 3, 5$; at every other q in the table $m(q) \geq 7$, so the $(q + 6, 3)$-arcs of that theorem are not of maximum size for any $q \geq 7$ considered here. The sixth point of [1, Theorem 5.1] exists for odd $q \geq 11$ and gives $m(q) \geq 6$; at $q = 11, 13, 17, 19$ (and at 23) the $(q + 7, 3)$-arcs so obtained are not of maximum size either. Thus the source's arcs are never of maximum size in the range where its sixth-point theorem applies.

Remark 5.3 ($m(9) = 7$, computed). $\mathbb{Z}/9\mathbb{Z}$ is not a field and the formal development takes q prime, so PG$(2, 9)$ was treated outside it, in an explicit model $\mathbb{F}_9 = \mathbb{F}_3[x]/(x^2 + 1)$, by the same depth-first search in C over the 81 points off the conic: the optimum is $m(9) = 7$, i.e. a $(17, 3)$-arc containing the conic and none larger. This reproduces the source's second reported value ($n = 17 = 2q - 1$). It is not a theorem of this paper. Formalising it requires a computable model of $\mathbb{F}_9$ (the library's general finite-field construction is not computable); the search itself is small.

Remark 5.4 ($m(23) = 11$, computed). Over the 529 points of PG$(2, 23)$ off the conic (553 lines), the C search ran to completion: 533 925 769 nodes in 1 864 CPU-seconds, optimum 11, i.e. a $(35, 3)$-arc containing a conic and none larger, and the decision run at target 12 visits 533 924 978 nodes. The witness

$$(1, 0, 1), (1, 1, 0), (1, 2, 0), (1, 5, 8), (1, 7, 16), (1, 8, 5), (1, 9, 10), (1, 13, 14), (1, 15, 5), (1, 17, 11), (1, 20, 4)$$

was re-verified independently (maximum line intersection 3, 142 trisecants). This value is not a theorem of this paper. The exhaustion visits 9.7 times the nodes of the one at $q = 19$, over longer candidate and line lists, so a single `native_decide` for it runs for hours where the module at $q = 19$ takes 1 005 s — past the budget under which the development was checked. (The C timings do not settle the figure more finely than that: the 1 864 s above is the run that produced the value, and the decision run at target 12 was not timed.) The route, if wanted, is to split the root of the search into about eight files, which needs one further lemma (that `go` from the empty set is `false` iff each of the root subproblems is), the induction of Theorem 3.1 being unchanged.

Remark 5.5 (growth). The values, with the ratio and the defect against q:

q	3	5	7	9	11	13	17	19	23
$m(q)$	5	5	7	7	7	8	9	10	11
$m(q)/q$	1.67	1.00	1.00	0.78	0.64	0.62	0.53	0.53	0.48
$2m(q) - q$	7	5	7	5	3	3	1	1	-1

The conjectural bound $m \leq q - 2$ would allow $9, 11, 15, 17, 21$ at $q = 11, 13, 17, 19, 23$. The source's sentence "the truth is indeed linear in q" is consistent with these data but the slope is small, and $m(q)/q$ is still falling at $q = 23$: the data do not support $m(q) \sim q/2$, and $m(q) = \lfloor (q + 1)/2 \rfloor$, which fits $q = 17$ and $q = 19$, fails at $q = 23$. Whether $m(q)/q$ has a positive limit is open; the only general bounds in print are $m \gg \log q$ from the source's character-sum method and $m \leq q + 1$. We claim no growth law.

6. Controls

The upper bounds are statements about lists of coordinate triples; the following verify that those lists are the geometry they are meant to be, and that the checker rejects what it should.

Proposition 6.1 (the model, $q = 7$ and $q = 11$)**.** *For $q = 7$ (verified entirely inside the kernel) and for $q = 11$ (with the pair sweep run compiled): the point list has $q^2 + q + 1$ distinct entries (57, resp. 133); every line carries exactly $q + 1$ of them (8, resp. 12); every two distinct points lie on exactly one common line (the 1 596, resp. 8 778 pairs); the conic list has $q + 1$ distinct entries, all in the plane, and a point of the plane is on it iff $P_1^2 = P_0 P_2$; every line meets the conic in at most two points; and the candidate list has exactly q^2 entries (49, resp. 121), each a point of the plane not on the conic.*

Without this, the search would be a packing statement about an unidentified hypergraph; the last three conjuncts are what make the count of $\mathcal{O} \mathbin{+\!\!+} T$ on a line equal to $|(\mathcal{O} \cup T) \cap L|$ rather than merely bound it. The same definitions are used at every q.

Proposition 6.2 (too large, refuted)**.** *The eight-point list $T_{11} \mathbin{+\!\!+} [(1, 0, 2)]$ does not extend $\mathcal{O}_{11}$, and the line $(1, 5, 5)$ carries exactly four points of $\mathcal{O}_{11} \cup T_{11} \cup \{(1, 0, 2)\}$.*

Proposition 6.3 (too small, refuted)**.** *It is false that every S extending $\mathcal{O}_{11}$ has $|S| \leq 6$.*

Proposition 6.4 (non-vacuity)**.** *For every $k \leq 7$ there is an S extending $\mathcal{O}_{11}$ with $|S| = k$.*

Proof sketch. Proposition 6.1: each conjunct is a kernel evaluation at $q = 7$ (`decide`, `decide +kernel`); at $q = 11$ the three length conjuncts and the duplicate-freeness of the conic are kernel evaluations and the remaining seven are `native_decide` (so seven native axioms on that statement). Proposition 6.2: the Boolean checker returns `false` on the eight-point list, and the count on $(1, 5, 5)$ is 4, both in the kernel. Proposition 6.3 follows from Theorem 4.4(a). Proposition 6.4: the prefixes of T_{11} of every length $0, \ldots, 7$ pass the checker (one kernel evaluation over all eight). $\square$

Proposition 6.3 shows the upper-bound theorem is sharp and Proposition 6.4 that its hypothesis is inhabited at every size up to the maximum; the control in the other direction — that a false "no eight points" is impossible — is Theorem 3.1 itself.

7. Verification

Every statement pinned above — Propositions 4.1–4.3, Theorems 4.4–4.7, Propositions 6.1–6.4, and Theorems 3.1 and 3.2 — is verified in Lean 4 [13] (toolchain v4.32.0) with Mathlib [14] (the release tagged v4.32.0); the statements are reproduced verbatim in Appendix A. The development has nine modules: `Search` (the abstract instance, `go`, Theorem 3.1) and `Core` (the natural-number model of $\mathrm{PG}(2, q)$ and the instance I_q), both free of Mathlib so that they compile to one shared library `libConicCore.so` (two `lean` invocations emitting C, two `leanc` compilations and one `gcc -shared`); `Geom` (the model over $\mathbb{Z}/q\mathbb{Z}$, `ExtendsConic`, Theorem 3.2); `Q7` ($q = 3, 5, 7$), `Q11`, `Q13`, `Q17`, `Q19` (one witness, one upper bound and one `IsGreatest` statement each); and `Controls`. Counting the sources, they hold 46 theorem declarations (7, 0, 13, 9, 3, 3, 3, 3, 5 in `Search`, `Core`, `Geom`, `Q7`, `Q11`, `Q13`, `Q17`, `Q19`, `Controls`) and 34 definitions (30 `def`s, two `abbrev`s and two `structure`s, besides one `Decidable` instance). Twenty-nine of the theorem statements are reproduced verbatim in Appendix A: Theorems 3.1 and 3.2 together with the general form `go_eq_false` from which the first is read off, and the 26 statements of Sections 4 and 6.

Axioms, from `#print axioms` on every headline in the compiled environment: Theorem 3.1 (`no_feasible_of_search`) uses `propext` and `Quot.sound`; Theorem 3.2 (`le_of_search`) those two and `Classical.choice`; every witness statement (part (a) of each value) uses `propext` and `Quot.sound` only, except the one at $q = 3$, which picks up `Classical.choice` through a simplification; every upper bound and every `IsGreatest` statement uses `propext`, `Classical.choice`, `Quot.sound` and the two native-evaluation axioms of its `native_decide` calls (the search, and the duplicate-freeness of the candidate list), one at $q = 11$; the model check at $q = 11$ carries seven native axioms and the other four controls none. Every `native_decide` in the development evaluates concrete data; none is a step of reasoning. There is no `sorry` and no axiom declared by hand.

The searches must run compiled: with the shared library loaded (`--load-dynlib`) the $q = 11$ exhaustion takes 1.68 s against 57.4 s in Lean's interpreter (34×), which is what puts $q = 17$ and $q = 19$ within reach. Two elaborator settings are needed: the module for $q = 17$ raises the heartbeat limit to 2 000 000 and the one for $q = 19$ disables it, because the elaborator's heartbeat clock keeps running through a compiled `native_decide` (the search at $q = 19$ exhausted 8 000 000 heartbeats although it does no elaboration; the neighbouring interpreted side condition costs 0.10 s and is not the bill); the witness modules also raise the recursion depth for the kernel evaluations. One full check takes, in seconds: `Search` 1.5, `Core` 1.7, `Geom` 6, `Q7` 9, `Q11` 23, `Q13` 26, `Q17` 302 (peak 7.3 GB), `Q19` 1 005 (peak 8.0 GB), `Controls` 237 (peak 9.3 GB). These are whole-module times, which include the `import Mathlib` and the kernel evaluations of the witness; the exhaustions at $q = 17$ and $q = 19$ were never timed apart from their modules, and the figures quoted for them elsewhere in this paper are the C times of Table 1 multiplied by the measured compiled-to-C factor, not measurements. Including the C searches (about 5 CPU-minutes for all $q \leq 19$ in both models and 31 CPU-minutes for $q = 23$), the CDCL cross-check (about 17 minutes, 946 s of it the stopped $q = 17$ instance) and three failed runs while the heartbeat setting was found (about 40 minutes), the items above total about two CPU-hours with peak resident memory 9.3 GB. The source of the development, the build script and the C program are currently held in a private repository; repository reference to be added at submission.

Appendix A. The formal statements

The Lean statements corresponding to the results above, verbatim (statements only; proofs, docstrings and attributes omitted), preceded by the definitions they use, grouped by module. In `Search`, `I.On l x` is incidence of candidate `x` with line `l`, `I.cnt l T` the number of members of `T` on `l`, `I.feas T` feasibility, `I.full T p` the lines through `p` saturated by `T`, `I.keep F x` survival of `x` against the lines `F`. In `Core`, `pgL q` is the point (and line) list (1), `conicL q` the conic, `candL q` the points off it, `capN q L` the capacity $3 - |L \cap \mathcal{O}|$, and `searchN q m` the search at target `m`. In `Geom`, `toZ` casts a natural-number triple to $(\mathbb{Z}/q\mathbb{Z})^3$, and `q` is an unconstrained section variable: `le_of_search` is stated for every `q`, and primality enters only where the statement is used, since it is primality that makes the list a plane.

Search.

```
structure Inst where
```

```
  n : Nat
  caps : Array Nat
  onl : Array (Array Bool)
  plns : Array (Array Nat)

def On (I : Inst) (l x : Nat) : Bool := (I.onl.getD l #[]).getD x false
def cnt (I : Inst) (l : Nat) (T : List Nat) : Nat := T.countP (fun x => I.On l x)
def feas (I : Inst) (T : List Nat) : Bool :=
  (List.range I.caps.size).all (fun l => decide (I.cnt l T ≤ I.caps.getD l 0))
def full (I : Inst) (T : List Nat) (p : Nat) : List Nat :=
  (I.plns.getD p #[]).toList.filter
    (fun l => decide (l < I.caps.size) && decide (I.caps.getD l 0 ≤ I.cnt l T))
def keep (I : Inst) (F : List Nat) (x : Nat) : Bool := F.all (fun l => !I.On l x)
def go (I : Inst) (tgt : Nat) : Nat → List Nat → List Nat → Bool
  | fuel, T, A =>
    if tgt ≤ T.length then true
    else
      match A, fuel with
      | [], _ => false
      | _ :: _, 0 => true
      | p :: rest, f + 1 =>
          go I tgt f (p :: T) (rest.filter (I.keep (I.full (p :: T) p)))
            || go I tgt f T rest

theorem go_eq_false :
    ∀ (fuel : Nat) (T A S : List Nat), I.go tgt fuel T A = false → S.Sublist A →
      I.feas (S.reverse ++ T) = true → S.length + T.length < tgt

theorem no_feasible_of_search (h : I.go tgt I.n [] (List.range I.n) = false)
    (S : List Nat) (hS : S.Sublist (List.range I.n)) (hf : I.feas S = true) :
    S.length < tgt
```

Core.

```
abbrev TripN := Nat × Nat × Nat
def pgL (q : Nat) : List TripN :=
  ((List.range q).flatMap (fun y => (List.range q).map (fun z => (1, y, z))))
    ++ (List.range q).map (fun z => (0, 1, z))
    ++ [(0, 0, 1)]
def onN (q : Nat) (L P : TripN) : Bool :=
  (P.1 * L.1 + P.2.1 * L.2.1 + P.2.2 * L.2.2) % q == 0
def conicL (q : Nat) : List TripN :=
  (List.range q).map (fun y => (1, y, y * y % q)) ++ [(0, 0, 1)]
def candL (q : Nat) : List TripN := (pgL q).filter (fun P => !(conicL q).contains P)
def pgN (q : Nat) : Array TripN := (pgL q).toArray
def candN (q : Nat) : Array TripN := (candL q).toArray
def capN (q : Nat) (L : TripN) : Nat := 3 - (conicL q).countP (onN q L)
def instN (q : Nat) : Inst :=
  let lines := pgN q
  let cand := candN q
  { n := cand.size
    caps := lines.map (capN q)
    onl := lines.map (fun L => cand.map (onN q L))
    plns := (Array.range cand.size).map (fun i =>
      (Array.range lines.size).filter (fun l => onN q (lines.getD l (0,0,0)) (cand.getD i (0,0,0)))) }
def searchN (q m : Nat) : Bool :=
  let I := instN q
  I.go m I.n [] (List.range I.n)
```

Geom.

```
abbrev Trip (q : ℕ) := ZMod q × ZMod q × ZMod q
def toZ {q : ℕ} (P : TripN) : Trip q := ((P.1 : ZMod q), (P.2.1 : ZMod q), (P.2.2 : ZMod q))
def OnLine {q : ℕ} (L P : Trip q) : Prop :=
  P.1 * L.1 + P.2.1 * L.2.1 + P.2.2 * L.2.2 = 0
def pgPts (q : ℕ) : List (Trip q) := (pgL q).map toZ
def conicPts (q : ℕ) : List (Trip q) := (conicL q).map toZ
def candPts (q : ℕ) : List (Trip q) := (candL q).map toZ

structure ExtendsConic (q : ℕ) (T : List (Trip q)) : Prop where
  nodup : T.Nodup
  off : ∀ P ∈ T, P ∈ candPts q
  arc : ∀ L ∈ pgPts q, ((conicPts q ++ T).countP (fun P => decide (OnLine L P))) ≤ 3

theorem le_of_search {tgt : ℕ} (hcand : (candPts q).Nodup)
    (hsearch : searchN q tgt = false)
    {T : List (Trip q)} (hT : ExtendsConic q T) : T.length < tgt
```

Q7 ($q = 3, 5, 7$; Propositions 4.1–4.3).

```
namespace Q3
def T : List (Trip 3) :=
  [(1,0,1), (1,1,0), (1,2,0), (0,1,1), (0,1,2)]
theorem extends_max :
    ExtendsConic 3 T ∧ T.length = 5 ∧
      ∃ L ∈ pgPts 3, ((conicPts 3 ++ T).countP (fun P => decide (OnLine L P))) = 3
theorem upper (S : List (Trip 3)) (h : ExtendsConic 3 S) : S.length ≤ 5
theorem isGreatest :
    IsGreatest {m : ℕ | ∃ S : List (Trip 3), ExtendsConic 3 S ∧ S.length = m} 5
end Q3

namespace Q5
def T : List (Trip 5) :=
  [(1,0,1), (1,1,0), (1,2,0), (1,3,2), (0,1,1)]
theorem extends_max :
    ExtendsConic 5 T ∧ T.length = 5 ∧
      ∃ L ∈ pgPts 5, ((conicPts 5 ++ T).countP (fun P => decide (OnLine L P))) = 3
theorem upper (S : List (Trip 5)) (h : ExtendsConic 5 S) : S.length ≤ 5
theorem isGreatest :
    IsGreatest {m : ℕ | ∃ S : List (Trip 5), ExtendsConic 5 S ∧ S.length = m} 5
end Q5

namespace Q7
def T : List (Trip 7) :=
  [(1,0,1), (1,1,0), (1,2,2), (1,4,6), (1,5,6), (1,6,4), (0,1,6)]
theorem extends_max :
    ExtendsConic 7 T ∧ T.length = 7 ∧
      ∃ L ∈ pgPts 7, ((conicPts 7 ++ T).countP (fun P => decide (OnLine L P))) = 3
theorem upper (S : List (Trip 7)) (h : ExtendsConic 7 S) : S.length ≤ 7
theorem isGreatest :
    IsGreatest {m : ℕ | ∃ S : List (Trip 7), ExtendsConic 7 S ∧ S.length = m} 7
end Q7
```

The trisecants exhibited in the proofs of `extends_max` are the lines $(1, 0, 0)$, $(1, 0, 4)$, $(1, 0, 3)$ respectively.

Q11 (Theorem 4.4).

```
def T : List (Trip 11) :=
  [(1,0,1), (1,1,3), (1,2,0), (1,3,6), (1,6,0), (1,7,4), (1,9,6)]
```

```
theorem extends_seven :
    ExtendsConic 11 T ∧ T.length = 7 ∧
      ∃ L ∈ pgPts 11, ((conicPts 11 ++ T).countP (fun P => decide (OnLine L P))) = 3
theorem upper (S : List (Trip 11)) (h : ExtendsConic 11 S) : S.length ≤ 7
theorem isGreatest :
    IsGreatest {m : ℕ | ∃ S : List (Trip 11), ExtendsConic 11 S ∧ S.length = m} 7
```

The trisecant exhibited is $(1, 0, 7)$.

Q13 (Theorem 4.5).

```
def T : List (Trip 13) :=
  [(1,0,1), (1,1,0), (1,2,7), (1,3,7), (1,6,0), (1,7,8), (1,11,3), (1,12,5)]
theorem extends_max :
    ExtendsConic 13 T ∧ T.length = 8 ∧
      ∃ L ∈ pgPts 13, ((conicPts 13 ++ T).countP (fun P => decide (OnLine L P))) = 3
theorem upper (S : List (Trip 13)) (h : ExtendsConic 13 S) : S.length ≤ 8
theorem isGreatest :
    IsGreatest {m : ℕ | ∃ S : List (Trip 13), ExtendsConic 13 S ∧ S.length = m} 8
```

The trisecant exhibited is $(1, 0, 4)$.

Q17 (Theorem 4.6).

```
def T : List (Trip 17) :=
  [(1,0,1), (1,1,0), (1,2,0), (1,3,2), (1,4,3), (1,8,12), (1,9,15), (1,13,8), (1,15,3)]
theorem extends_max :
    ExtendsConic 17 T ∧ T.length = 9 ∧
      ∃ L ∈ pgPts 17, ((conicPts 17 ++ T).countP (fun P => decide (OnLine L P))) = 3
theorem upper (S : List (Trip 17)) (h : ExtendsConic 17 S) : S.length ≤ 9
theorem isGreatest :
    IsGreatest {m : ℕ | ∃ S : List (Trip 17), ExtendsConic 17 S ∧ S.length = m} 9
```

The trisecant exhibited is $(1, 0, 2)$.

Q19 (Theorem 4.7).

```
def T : List (Trip 19) :=
  [(1,0,1), (1,1,3), (1,2,7), (1,5,10), (1,7,4), (1,8,6), (1,10,9), (1,11,17), (1,17,10), (1,18,15)]
theorem extends_max :
    ExtendsConic 19 T ∧ T.length = 10 ∧
      ∃ L ∈ pgPts 19, ((conicPts 19 ++ T).countP (fun P => decide (OnLine L P))) = 3
theorem upper (S : List (Trip 19)) (h : ExtendsConic 19 S) : S.length ≤ 10
theorem isGreatest :
    IsGreatest {m : ℕ | ∃ S : List (Trip 19), ExtendsConic 19 S ∧ S.length = m} 10
```

The trisecant exhibited is $(1, 0, 2)$.

Controls (Propositions 6.1–6.4).

```
theorem model_seven :
    (pgPts 7).length = 57 ∧ (pgPts 7).Nodup ∧
    (∀ L ∈ pgPts 7, (pgPts 7).countP (fun P => decide (OnLine L P)) = 8) ∧
    (∀ P ∈ pgPts 7, ∀ P' ∈ pgPts 7, P ≠ P' →
      (pgPts 7).countP (fun L => decide (OnLine L P) && decide (OnLine L P')) = 1) ∧
    (conicPts 7).length = 8 ∧
    (∀ P ∈ pgPts 7, (P ∈ conicPts 7 ↔ P.2.1 ^ 2 = P.1 * P.2.2)) ∧
    (∀ L ∈ pgPts 7, (conicPts 7).countP (fun P => decide (OnLine L P)) ≤ 2) ∧
    (candPts 7).length = 49 ∧
    (conicPts 7).Nodup ∧ (∀ P ∈ conicPts 7, P ∈ pgPts 7) ∧
    (∀ P ∈ candPts 7, P ∈ pgPts 7 ∧ P ∉ conicPts 7)

theorem model_eleven :
    (pgPts 11).length = 133 ∧ (pgPts 11).Nodup ∧
    (∀ L ∈ pgPts 11, (pgPts 11).countP (fun P => decide (OnLine L P)) = 12) ∧
```

```
    (∀ P ∈ pgPts 11, ∀ P' ∈ pgPts 11, P ≠ P' →
      (pgPts 11).countP (fun L => decide (OnLine L P) && decide (OnLine L P')) = 1) ∧
    (conicPts 11).length = 12 ∧
    (∀ P ∈ pgPts 11, (P ∈ conicPts 11 ↔ P.2.1 ^ 2 = P.1 * P.2.2)) ∧
    (∀ L ∈ pgPts 11, (conicPts 11).countP (fun P => decide (OnLine L P)) ≤ 2) ∧
    (candPts 11).length = 121 ∧
    (conicPts 11).Nodup ∧ (∀ P ∈ conicPts 11, P ∈ pgPts 11) ∧
    (∀ P ∈ candPts 11, P ∈ pgPts 11 ∧ P ∉ conicPts 11)

theorem too_large_refuted :
    ¬ ExtendsConic 11 (Q11.T ++ [(1, 0, 2)]) ∧
      ((conicPts 11 ++ (Q11.T ++ [(1, 0, 2)])).countP
        (fun P => decide (OnLine ((1, 5, 5) : Trip 11) P))) = 4

theorem too_small_refuted : ¬ (∀ S : List (Trip 11), ExtendsConic 11 S → S.length ≤ 6)

theorem all_sizes_realised (k : ℕ) (hk : k ≤ 7) :
    ∃ S : List (Trip 11), ExtendsConic 11 S ∧ S.length = k
```

References

[1] S. Dönmez, A. Akpınar and R. M. Pelen, *Near-MDS codes of lengths $q+6$ and $q+7$ from conics in* $\mathrm{PG}(2,q)$*, q odd*, arXiv:2608.18192v1 [cs.IT] (18 August 2026). Theorem, proposition, equation and problem numbers cited in this paper are those of the e-print compiled from the version 1 source (18 pages), and all quotations are verbatim from it.

[2] S. Marcugini, A. Milani and F. Pambianco, *Classification of the* $[n,3,n-3]_q$ *NMDS codes over* $\mathrm{GF}(7)$*,* $\mathrm{GF}(8)$ *and* $\mathrm{GF}(9)$, Ars Combin. 61 (2001) 263–269, Zbl 1075.94023 (reference [15] of [1]). This journal assigns no DOI.

[3] S. Marcugini, A. Milani and F. Pambianco, *NMDS codes of maximal length over* $\mathbb{F}_q$*,* $8 \le q \le 11$, IEEE Trans. Inform. Theory 48 (2002), no. 4, 963–966, doi:10.1109/18.992802 (reference [16] of [1]).

[4] S. Marcugini, A. Milani and F. Pambianco, *Maximal* $(n,3)$*-arcs in* $\mathrm{PG}(2,11)$, Discrete Math. 208/209 (1999) 421–426, doi:10.1016/S0012-365X(99)00202-2.

[5] S. Marcugini, A. Milani and F. Pambianco, *Maximal* $(n,3)$*-arcs in* $\mathrm{PG}(2,13)$, Discrete Math. 294 (2005), no. 1–2, 139–145, doi:10.1016/j.disc.2004.04.043.

[6] S. Ball, *Table of bounds on three dimensional linear codes or* (n,r) *arcs in* $\mathrm{PG}(2,q)$, `http://web.mat.upc.edu/simeon.michael.ball/codebounds.html` (last revised 12 November 2019; consulted 7 September 2026).

[7] K. Coolsaet and H. Sticker, *The complete* $(k,3)$*-arcs of* $\mathrm{PG}(2,q)$*,* $q \le 13$, J. Combin. Des. 20 (2012), no. 2, 89–111, doi:10.1002/jcd.20293 (published online 25 August 2011).

[8] S. Ball, A. Blokhuis and F. Mazzocca, *Maximal arcs in Desarguesian planes of odd order do not exist*, Combinatorica 17 (1997) 31–41, doi:10.1007/BF01196129 (reference [2] of [1]).

[9] J. Lu and Y. Zhou, *On the equivalence of NMDS codes*, IEEE Trans. Inform. Theory 71 (2025) 9506–9515, doi:10.1109/TIT.2025.3627921 (reference [14] of [1]).

[10] S. Dodunekov and I. Landgev, *On near-MDS codes*, J. Geom. 54 (1995) 30–43, doi:10.1007/BF01222850 (reference [7] of [1], where the second author's name is transliterated Landjev, as it is in his later papers).

[11] S. Innamorati and M. Zannetti, *The shape of the* $(15,3)$*-arc of* $\mathrm{PG}(2,7)$, Mathematics 9 (2021), no. 5, article 486, doi:10.3390/math9050486.

[12] J. W. P. Hirschfeld, *Projective Geometries over Finite Fields*, 2nd edn., Oxford Mathematical Monographs, Clarendon Press, Oxford, 1998, xiv+555 pp., ISBN 0-19-850295-8 (reference [10] of [1]).

[13] L. de Moura and S. Ullrich, *The Lean 4 theorem prover and programming language*, in: Automated Deduction — CADE 28, Lecture Notes in Computer Science 12699, Springer, 2021, 625–635, doi:10.1007/978-3-030-79876-5_37.

[14] The mathlib Community, *The Lean mathematical library*, in: Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP 2020), ACM, 2020, 367–381, doi:10.1145/3372885.3373824.