



Two New Ways to Formally Prove Dandelin-Gallucci's Theorem

David Braun, Nicolas Magaud, Pascal Schreck

► To cite this version:

David Braun, Nicolas Magaud, Pascal Schreck. Two New Ways to Formally Prove Dandelin-Gallucci's Theorem. ISSAC 2021 : International Symposium on Symbolic and Algebraic Computation, Jul 2021, Saint Petersburg (virtual event), Russia. 10.1145/3452143.3465550 . hal-03225987

HAL Id: hal-03225987

<https://hal.science/hal-03225987>

Submitted on 13 May 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Two New Ways to Formally Prove Dandelin-Gallucci's Theorem

David Braun
Nicolas Magaud
Pascal Schreck
david.braun@unistra.fr
magaud@unistra.fr
schreck@unistra.fr

ICube UMR 7357 CNRS Université de Strasbourg
Strasbourg, France

ABSTRACT

Mechanizing proofs of geometric theorems in 3D is significantly more challenging than in 2D. As a first noteworthy case study, we consider an iconic theorem of 3D geometry: Dandelin-Gallucci's theorem. We work in the very simple but powerful framework of projective incidence geometry, where only incidence relationships are considered. We study and compare two new and very different approaches to prove this theorem. First, we propose a new proof based on the well-known Wu's method. Second, we use an original method based on matroid theory to generate a proof script which is then checked by the Coq proof assistant. For each method, we point out which parts of the proof we manage to carry out automatically and which parts are more difficult to automate and require human interaction. We hope these first developments will lead to formally proving more 3D theorems automatically and that it will be used to formally verify some key properties of computational geometry algorithms in 3D.

ACM Reference Format:

David Braun, Nicolas Magaud, and Pascal Schreck. 2021. Two New Ways to Formally Prove Dandelin-Gallucci's Theorem. In *Proceedings of The 46th International Symposium on Symbolic and Algebraic Computation ISSAC 2021 (ISSAC'21)*. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnn>

1 INTRODUCTION

In this paper, we are interested in the mechanization of the Dandelin-Gallucci's theorem [15] which is specific of 3D incidence projective geometry. This effort is part of a long term project aiming at formally proving geometric algorithms. We investigate two main approaches: the first one is based on algebraic methods and the second one relies on a combinatorial proof search and produces a proof witness which is then checked by Coq [2, 10]. This article deals with incidence geometry and we focus on the proof of a non-trivial theorem which is a kind of benchmark of our methods.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ISSAC'21, July 18–22, 2021, Saint Petersburg, Russia

© 2021 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/Y/MM...\$15.00

<https://doi.org/10.1145/nnnnnnnn.nnnnnnn>

It is well known in geometry and computer graphics that there is a gap between 2D and 3D. This happens even if only very basic relationships between objects are considered. For instance, 2D projective incidence geometry only considers points and lines such that two distinct points define a line, two distinct lines define a point, each line is incident to at least three points, and the plane contains at least two distinct lines. Describing 3D incidence projective geometry is a bit more complex. Intuitively, we must handle new objects, e.g. planes, and new properties such as the fact that two lines do not necessarily intersect, that any plane meets any line, that two planes intersect along a line, etc. A set of axioms describing projective geometry is given in this paper. As we shift to 3D, the axiom system deals with more objects and properties and thus gets bigger, reflecting the gap between 2D and 3D. Another significant difference between 2D and 3D is the role Desargues property plays: it is not verified in general in 2D, whereas this is a theorem in 3D. The consequences are important: any projective incidence space whose dimension is greater than 3 is related to a division ring, but this is not true for the projective incidence plane. This qualitative gap between 2D and 3D extends to the mechanization of proofs in geometry: although there are several methods to automatically prove some families of 2D geometric theorems, there is not that many methods available for 3D geometry.

In a three-dimensional projective space, Dandelin-Gallucci's property states that given two sets of three skew lines, if each line of the first set intersects all lines of the second set, then any line which intersects all three lines of the first set intersects any line which intersects all three lines of the second set, as shown in Fig. 1. Dandelin-Gallucci's theorem establishes that Dandelin-Gallucci's property is equivalent to Pappus' property.

In the following, we study how to prove Dandelin-Gallucci's theorem as formally as possible. The configuration of Dandelin-Gallucci is interesting because it only involves lines, which may not be coplanar, making it a real 3D configuration. It is then a good candidate for such investigations. In addition, Dandelin-Gallucci's theorem makes the connection between the well-known 2D Pappus and 3D Dandelin-Gallucci's properties.

The methods that we use for proving the theorem are based on very different approaches: the first one relies on pure synthetic geometry, the second one is algebraic and relies on Wu's method, the third one is combinatorial and relies on matroid theory. The proof in the framework of synthetic geometry is known for a long time and we simply use it as a reference proof. The two other approaches (the

algebraic and combinatorial one) are original and are good candidates for automation. Indeed, most steps of the proofs can be carried out automatically. However, some human interaction is still required, either to encode geometry into algebra for the Wu method or to deal with existential quantification in the combinatorial approach.

In addition, our combinatorial approach yields a formal proof which is formally verified by Coq [2, 10]. Coq is a proof assistant based on type theory, where proof-checking boils down to type-checking. The user (or a program written by the user) will generate a formal statement and a proof script in the formal language of Coq and the system will automatically type-check (i.e. proof-check) it to verify that it actually proves the statement at stake.

The paper is organized as follows. In Sect. 2, we introduce the usual synthetic axiom system for projective geometry, state Dandelin-Gallucci's property and outline the synthetic proof that Pappus property implies Dandelin-Gallucci's property. In Sect. 3, we use the well-known algebraic method of the late Prof. Wu to prove the theorem in a 3D framework. In Sect. 4, we introduce the concept of matroids and ranks to deal with projective geometry as proposed by Michelucci and Schreck in [17]. In Sect. 5, we describe an automatic theorem prover based on this approach and apply it to proving Dandelin-Gallucci's theorem.

2 DANDELIN-GALLUCCI'S THEOREM

Let us first remind the reader what projective incidence geometry is and then give the statement of this theorem.

Dandelin-Gallucci's property lies in the framework of projective incidence geometry where there are no metric and no coordinates. Only points and lines are considered. The incidence relation, denoted by $\in$, is described through a very simple set of axioms of classical first order logic (See Tab. 1).

- (i) $\forall x y : \text{Point}, \exists d : \text{Line}, x \in d \wedge y \in d$
- (ii) $\forall x y : \text{Point}, \forall d e : \text{Line},$
 $x \in d \wedge x \in e \wedge y \in d \wedge y \in e \Rightarrow x = y \vee d = e$
- (iii) $\forall x y z t : \text{Point}, \forall l_{xy} l_{zt} l_{xz} l_{yt} : \text{Line},$
 $x \in l_{xy} \wedge y \in l_{xy} \wedge z \in l_{zt} \wedge t \in l_{zt} \wedge$
 $x \in l_{xz} \wedge z \in l_{xz} \wedge y \in l_{yt} \wedge t \in l_{yt} \wedge$
 $(\exists u : \text{Point}, u \in l_{xy} \wedge u \in l_{zt}) \Rightarrow$
 $(\exists v : \text{Point}, v \in l_{xz} \wedge v \in l_{yt})$
- (iv) $\forall d : \text{Line}, \exists x y z : \text{Point},$
 $x \neq y \wedge x \neq z \wedge y \neq z \wedge x \in d \wedge y \in d \wedge z \in d$

Table 1: Axiom system for projective incidence geometry

Some axioms can be added to reflect the dimension of the considered projective space. For instance, for 2D projective incidence, we add the following axioms:

- (ld2) $\exists d e : \text{Line}, d \neq e$
- (ud2) $\forall d e : \text{Line}, \exists x : \text{Point}, x \in d \wedge x \in e$

and for the 3D projective incidence spaces, we have:

- (ld3) $\exists d e : \text{Line}, \forall x : \text{Point}, x \notin d \vee x \notin e$
- (ud3) $\forall x : \text{Point}, \forall d e : \text{Line}, \exists f : \text{Line}, \exists y z : \text{Point}$
 $x \in f \wedge y \in f \wedge z \in f \wedge y \in d \wedge z \in e$

The last axiom (ud3) is characteristic of the 3D space and it is called the *transversal axiom*. Then, in an incidence space with

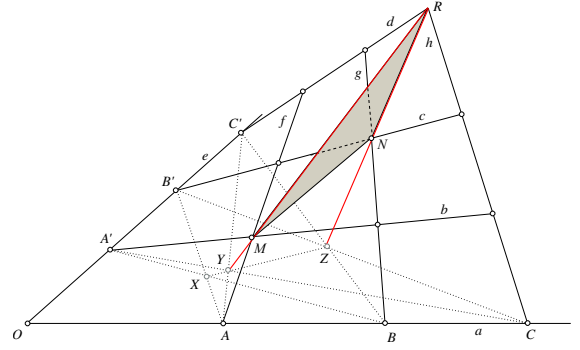


Figure 1: From Pappus to Dandelin-Gallucci (lines are noted with lower case letters $a, b, \dots$).

dimension at least 3, Dandelin-Gallucci's property can be informally stated as follows:

PROPERTY 1 (DANDELIN-GALLUCCI). *Three skew lines a, b and c being given and three other skew lines e, f and g being also given, such that every line in $\{a, b, c\}$ meets every line in $\{e, f, g\}$. Then, all pairs of lines d and h such that d meets lines e, f and g and h meets every lines a, b and c , are concurrent.*

It is important to note that this property is not satisfied by every projective incidence space. In fact it is related to Pappus's property which lives in a 2D plane and can be stated as follows:

PROPERTY 2 (PAPPUS). *In a projective incidence plane, let a and e be two distinct lines and let A, B, C, A', B' and C' be six different points with A, B and C belonging to a and A', B' and C' belonging to e . These points define respectively the lines $l_{AB'}, l_{A'B}, l_{AC'}, l_{A'C}, l_{BC'}, l_{B'C}$. The three intersection points $X = l_{AB'} \cap l_{A'B}$ and $Y = l_{AC'} \cap l_{A'C}$ and $Z = l_{BC'} \cap l_{B'C}$ are collinear.*

Note that Pappus' property is not a theorem of the general theory of projective incidence geometry. Nevertheless, it is one of the key properties of the so-called *fundamental theorem of incidence geometry*, which states that a projective incidence geometry can be built from a field (using coordinates) if and only if Pappus's property holds in this geometry.

Dandelin-Gallucci's *theorem* then establishes a strong link between an iconic 2D property (Pappus) and a truly 3-dimensional one (Dandelin-Gallucci's):

THEOREM 3 (DANDELIN-GALLUCCI). *In a projective incidence space whose dimension is greater than or equal to 3, Dandelin-Gallucci's property and Pappus' property are equivalent.*

Due to the lack of space, we do not detail the proof of the theorem here. This proof only uses basic knowledge on incidence geometry. Fig. 1 illustrates the configuration and names some interesting lines. This sketch highlights the role of Pappus points X, Y and Z which are not part of the initial configuration and are later used to construct the point R as the intersection of lines (YM) and (ZN) . We then show that this new point R is also the intersection of lines d and h . The details of proof can be found in Horváth's article [15].

As far as we know, this proof has not been mechanized into a formal geometric framework yet. We propose below two new proofs of this

theorem¹. The first one follows an algebraic approach popularized by the late Prof. Wu. Since this approach requires using coordinates and polynomials, only the direction "from Pappus to Dandelin-Gallucci" can be achieved. Indeed, without assuming that Pappus' property holds, the fundamental theorem of incidence geometry prevents us from using coordinates to represent this incidence projective structure. The second method is based on a combinatorial approach and it uses matroids. It is able to handle the two ways of the equivalence since no coordinates are considered.

3 WU'S METHOD IN 3D

The method proposed by Wu in the late seventies [23] is, until now, among the most powerful methods in the domain of geometric theorem proving.

Using Wu's method in order to prove Dandelin-Gallucci's theorem in an arbitrary setting is not possible. Indeed, Wu's method is based on a translation from geometry to algebra by describing geometric figures using polynomials. This translation is possible only if the incidence space comes from a field. Using the fundamental theorem of incidence geometry, this means that this approach is practicable only if Pappus's property holds in this space. Therefore, in this section, we can only show how to prove that Pappus's property implies Dandelin-Gallucci's property in a projective incidence space.

We remind the reader that Wu's method consists in:

- translating a geometric figure into a set of algebraic constraints, more specifically polynomials $h_1, h_2, \dots, h_n$ and then choosing an order on the variables $x_1, x_2, \dots, x_n$ such as x_j is the leading variable of h_j ,
- translating the property to be proved into a polynomial g ,
- using the pseudo-division technique several times to prove that g is in the saturation of the ideal $\langle h_1, h_2, \dots, h_n \rangle$ by the ideal $\langle I_1, \dots, I_n \rangle$ where I_j is the leading coefficient of h_j with respect to x_j . This means that g is in the ideal spanned by the hypotheses but not in the ideal spanned by the leading coefficients of polynomials h_i . In other words, g is a consequence of the hypotheses under some conditions of non-degeneracy.

Comprehensive descriptions of Wu's method are available in [8, 12, 23]. The two first steps are not easy to mechanize, and especially in 3D, as we will see below.

In the following, we use Maple standard tools available from RegularChains library. This fast and reliable library provides tools to decompose an algebraic system into irreducible triangular components [7].

Translating geometry to algebra.

Usually, homogeneous coordinates are used in projective geometry: a lifting to dimension $n+1$ is considered and the coordinates are taken up to a multiplicative coefficient. For instance, in 3D one considers coordinates (x, y, z, t) for $(x/t, y/t, z/t)$; $t = 0$ corresponds to a point at the infinity from the affine point of view—and simply a point in the plane of equation $t = 0$ from the projective point of view.

In 3D projective space, and using homogeneous coordinates, the fact that four points M, N, P , and Q are coplanar can be expressed as

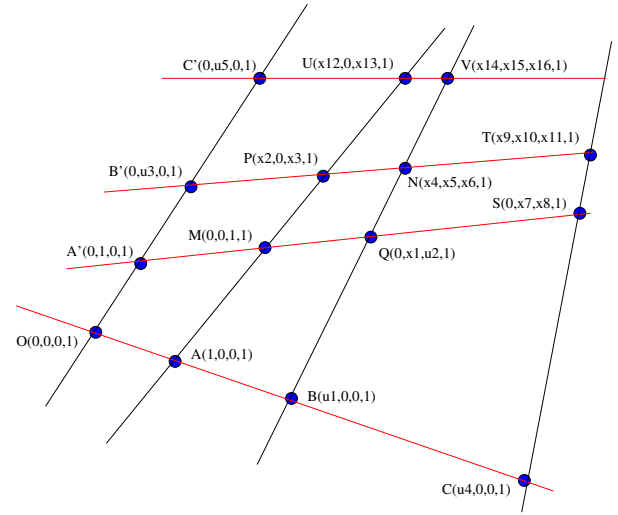


Figure 2: Coordinates for Dandelin-Gallucci's property

$\det(M, N, P, Q) = 0$: this is one of the algebraic equations that can be considered in the Wu's method. It is a bit more technical to express collinearity: three points M, N and P are collinear if and only if the rank of the matrix $\langle M, N, P \rangle$ is 2, meaning that all minors of order 3 have to be zero. But among the 4 minors, only 2 are needed. Fortunately, we can take all these equations into account since, like for Gröbner bases, computing characteristic sets eliminates redundant equations. The complexity of algorithms computing characteristic sets or regular chains are double-exponential. It remains a very active research fields and some practical improvements were recently proposed [1, 19].

To use these tools in Wu's method, coordinates for the figure corresponding to Dandelin-Gallucci's property have to be chosen. Classically, we operate these choices by using a construction of the figure which is based on the transversal axiom.

As it is simpler to construct points than lines, our construction proceeds as shown in Fig. 2: we first choose a reference by fixing the four points $O(0, 0, 0, 1)$, $A(1, 0, 0, 1)$ and $A'(0, 1, 0, 1)$ in the affine plane $z = 0$, and $M(0, 0, 1, 1)$ out of this plane. Such a choice can be done without loss of generality since the property is invariant under the action of the similarity group. Then, points B and B' are chosen on lines OA and OA' respectively, the parameters u_1 and u_3 indicate that the position on these lines are arbitrary (however, we choose not to fix them at infinity). The next move consists in fixing the point Q on line $A'M$: the reader can see that there is an unknown x_1 meaning that Q has to be on line $A'M$, and a parameter u_2 meaning that that point can be anywhere on that line. Points P and N are then computed from the previous points they have some unknowns as coordinates, x_2, x_3, x_4, x_5 and x_6 , which are solutions of equations translating the fact that the line $B'PN$ is the transversal line from B' meeting both lines AM and BQ .

Note also that the second coordinate of point P is 0 since P is in the affine plane $y = 0$, as A and M are. In a similar way, points C and C' are chosen on respectively lines OA and OA' , and points S, T, U and V are on transversal lines from respectively points C and

¹See <https://github.com/pascalschreck/Dandelin-Gallucci-for-Issac21> for the implementations.

C' . The conclusion of the theorem is that these points C', U, V, T, S and C are coplanar.

This configuration with the simplifications described above leads to consider sixteen unknowns and five parameters. Note that, for the sake of simplicity, we impose here that all the points are in the affine space: for instance, we chose $(0, 0, 1, 1)$ as M coordinates instead of $(0, 0, 1, u_6)$. Geometrically, it means that we do not consider the case where lines AM and $A'M$ are parallel. Indeed, allowing points at infinity would result in a much more complex system with five additional parameters, and seven more unknowns.

Thanks to the construction provided in Fig.2, we have an almost triangulated system and thus applying Wu's method is straightforward.

Let us go back to our theorem. All the constraints corresponding to Fig. 2 are related with incidence and are translated through the minors of a matrix. For instance, the fact that B', P and N are collinear is expressed as the fact that all minors of order 3 of the matrix are equal to zero:

$$\begin{pmatrix} 0 & x_2 & x_4 \\ u_3 & 0 & x_5 \\ 0 & x_3 & x_6 \\ 1 & 1 & 1 \end{pmatrix}$$

The minors implying respectively lines 1, 2, 3 and lines 1, 3, 4 are multiples of each other: they cannot be both used as equations for the figure. The minors implying respectively lines 2, 3, 4 and lines 1, 3, 4 lead to the equations: $u_3x_3 - u_3x_6 - x_3x_5 = 0$ and $x_2x_6 - x_3x_4 = 0$. The whole equation system is:

$$\begin{cases} -u_2 + 1 - x_1 & = 0 \\ x_2x_6 - x_3x_4 & = 0 \\ u_3x_3 - u_3x_6 - x_3x_5 & = 0 \\ -x_3 + 1 - x_2 & = 0 \\ -u_1(u_2x_5 - x_1x_6) & = 0 \\ u_1u_2 - u_1x_6 - u_2x_4 & = 0 \\ -u_4(x_{10}x_8 - x_{11}x_7) & = 0 \\ -u_4x_{11} + u_4x_8 - x_8x_9 & = 0 \\ -x_8 + 1 - x_7 & = 0 \\ -u_3(x_{11}x_2 - x_3x_9) & = 0 \\ -u_3x_{11} + u_3x_3 - x_{10}x_3 & = 0 \\ -u_5(x_{12}x_{16} - x_{13}x_{14}) & = 0 \\ u_5x_{13} - u_5x_{16} - x_{13}x_{15} & = 0 \\ -x_{13} + 1 - x_{12} & = 0 \\ -u_1(u_2x_{15} - x_{11}x_{16}) & = 0 \\ u_1u_2 - u_1x_{16} - u_2x_{14} & = 0 \end{cases}$$

and the conclusion is translated in the following equation:

$$g : u_4u_5x_{13} - u_4u_5x_8 - u_4x_{13}x_7 + u_5x_{12}x_8 = 0$$

In a standard PC computer² the function `Triangularization` of the Maple `RegularChains` library gives 397 irreducible triangular systems in about 4 minutes with the non-redundant system. Among them, only the number #334 is not degenerated. Note that, against intuition, if we consider a system with redundant equations, it takes about 2 minutes to produce only 75 irreducible triangular systems. In both cases, only one irreducible system is not degenerated. The corresponding Maple files can be found in our git repository:

<https://github.com/pascalschreck/Dandelin-Gallucci-for-Issac21>.

$$\begin{cases} x_1 + u_2 - 1 & = 0 \\ x_2 + x_3 - 1 & = 0 \\ (x_{10} - u_3)x_3 + x_{11}u_3 & = 0 \\ u_2x_4 - u_1u_2 + x_6u_1 & = 0 \\ u_2x_5 + (u_2 - 1)x_6 & = 0 \\ (x_{10}u_2 + (u_2 - 1)x_{11} - u_2u_3)x_6 + x_{11}u_2u_3 & = 0 \\ x_7 + x_8 - 1 & = 0 \\ (x_{10} + x_{11})x_8 - x_{11} & = 0 \\ x_9 + x_{10}u_4 + x_{11}u_4 - u_4 & = 0 \\ (u_4u_3 - 1)x_{10} + u_3(u_4 - 1)x_{11} + (-u_4 + 1)u_3 & = 0 \\ ((u_4u_3 + u_2 - 1)u_1 - u_2u_3u_4)x_{11} - u_1u_2u_3u_4 & = 0 \\ & + u_2u_3u_4 \\ x_{12} + x_{13} - 1 & = 0 \\ (x_{15} - u_5)x_{13} + x_{16}u_5 & = 0 \\ u_2x_{14} - u_1u_2 + x_{16}u_1 & = 0 \\ u_2x_{15} + x_{16}(u_2 - 1) & = 0 \\ (u_5u_1 - 1 + (-u_5 + 1)u_2)x_{16} - u_1u_2u_5 + u_2u_5 & = 0 \end{cases}$$

This system comes with the additional conditions that the leading coefficients of the equations are not equal to zero and thus correspond to non-degenerated conditions:

$$\begin{cases} x_{10} - u_3 & \neq 0 \\ x_{10} + x_{11} & \neq 0 \\ -u_2u_3 + u_2x_{10} + u_2x_{11} - x_{11} & \neq 0 \\ x_{15} - u_5 & \neq 0 \\ u_1u_5 - u_2u_5 + u_2 - 1 & \neq 0 \\ u_1u_3u_4 - u_2u_3u_4 + u_1u_2 - u_1 & \neq 0 \\ u_2 & \neq 0 \\ u_4u_3 - 1 & \neq 0 \end{cases}$$

To find this system, we used a very simple *ad hoc* filter which eliminates each triangular system containing at least an equation reduced to one variable (parameter or unknown).

Now, performing successive pseudo-divisions of g by the equations of that system, we get the null polynomial meaning that point E, F, U and S are coplanar.

Results and discussion.

We share here some thoughts on this section by pointing out some features of 3D projective geometry. Some of them are also mentioned in [21]. In the process described above, some parts are fully automatic, for instance, computing the irreducible components and performing the successive pseudo-divisions, but the mechanization is not obvious. Several issues occur when it comes to translating statements from geometry to algebra:

- it is well known that the geometry handled by Wu's method is not ordered and implicitly considers complex coordinates. For instance, the method is unable to prove Maclane theorem 8₃ [9, 14].
- choosing the coordinates is usually guided by a geometric construction: this is well described in 2D, for instance in Chou's book [8], but it is more intricate in 3D even in the simple case of incidence geometry,
- similarly, when considering projective geometry within algebra and homogeneous coordinates, one may face an explosion of possible cases,

Thus, the decomposition into irreducible components yields hundreds of algebraic systems and all of them have to be examined.

²Using an Intel(R) Core(TM) i7-4790 CPU @3.60GHz with 16GB of memory.

Also, it may seem difficult to trust computer algebra systems. We studied this issue in the context of 2D some years ago [13], but to be exhaustive, we also had to prove the correctness of the link from synthetic geometry to algebra (this is partially done in the Boutry *et al.* paper [3] in the context of 2D). To generate a Coq proof of the whole process presented in this section, we still need to adapt the above-mentioned approaches to 3D.

4 MATROID THEORY APPLIED TO 3D INCIDENCE GEOMETRY

An alternative approach to formalize incidence geometry—and therefore to tackle the proof of Dandelin-Gallucci's theorem—is based on matroid theory [18]. The theory was introduced by Whitney in 1935 to abstractly capture the essence of dependence of the columns of matrices, but it is very general and can be found in different computer science domains such as algorithmic and graph theory [11, 22].

Matroids offer several equivalent concepts to describe and to generalize the main set properties of linear dependence in vector space. In our work, we found it convenient to use the notion of rank [17]. When combined with a finite set of points, it captures incidence, collinearity and coplanarity properties between these points. Furthermore, using ranks allows to deal only with points, thus reducing the number of objects at stake as well as the number of relations (point-line, point-plane, line-plane incidence, etc.). It makes proof automation easier because we do not handle directly lines, planes or higher dimension sub-spaces. Indeed, all the concepts encountered in incidence geometry can be expressed using only sets of points and their ranks.

Formally, an integer function rk on a finite set E is the rank function of a matroid if and only if conditions of Table 2 are satisfied. Here again, we consider a classical logic framework.

(A1) non-negative and subcardinal:

$$\forall X, 0 \leq rk(X) \leq |X|$$

(A2) non-decreasing:

$$\forall X Y, X \subseteq Y \Rightarrow rk(X) \leq rk(Y)$$

(A3) submodular:

$$\forall X Y, rk(X \cup Y) + rk(X \cap Y) \leq rk(X) + rk(Y)$$

Table 2: Rank function properties, X and Y are subsets of E

We define such a rank function on finite sets of points in the framework of projective geometry. Let M be a matroid on a finite set E with the rank function rk as above. The closure $cl(F)$ of a subset F is defined by $cl(F) = \{x \in E \mid rk(F) = rk(F \cup \{x\})\}$. A set whose closure equals itself is said to be *closed*. It is also defined as a *flat*.

A set is a flat if it is maximal for its rank, meaning that the addition of any other element to the set would increase the rank. In other words, the rank of a flat F is the cardinal of a smallest set generating F . Actually, the classic rank function corresponding to incidence geometry is defined by identifying the flats with linear sub-spaces. Some examples of sets of points and their ranks are presented in Table 3.

Using this definition, it can be shown that every projective space has a matroid structure, but the converse is not true. To capture

$rk(\{A, B\}) = 1$	$A = B$
$rk(\{A, B\}) = 2$	$A \neq B$
$rk(\{A, B, C\}) = 2$	points A, B, C are collinear with at least two of them distinct
$rk(\{A, B, C\}) \leq 2$	points A, B, C are collinear
$rk(\{A, B, C\}) = 3$	points A, B, C are not collinear
$rk(\{A, B, C, D\}) = 3$	points A, B, C, D are coplanar, not all collinear
$rk(\{A, B, C, D\}) = 4$	points A, B, C, D are not coplanar

Table 3: Ranks of some sets of points and their geometric interpretations

the geometric aspects of our framework, we need to introduce new axioms in addition to the matroid ones. Table 4 presents such axioms which allow to fully define a rank-based axiom system for projective incidence geometry.

Using the Coq proof assistant, we formally showed in [6] that this set of axioms together with the matroid axioms (A1) to (A3) is equivalent to the usual synthetic axiom system presented in Sect. 2. Axioms (A1) to (A3) enforce the matroid properties. Axioms (A4) *Rk-Singleton* and (A5) *Rk-Couple* allow to scale the range of the rank function. Axioms (A6) *Rk-Pasch*, (A7) *Rk-Three-Points*, (A8) *Rk-Lower-Dimension* and (A9) *Rk-Upper-Dimension* are straightforward translations of the usual synthetic axioms of projective geometry. Thanks to this equivalence result, we only consider the axiom system based on ranks from now on.

(A4) Rk-Singleton:

$$\forall P : \text{Point}, rk(\{P\}) = 1$$

(A5) Rk-Couple:

$$\forall P Q : \text{Point}, P \neq Q \Rightarrow rk(\{P, Q\}) = 2$$

(A6) Rk-Pasch:

$$\forall A B C D : \text{Point}, rk(\{A, B, C, D\}) \leq 3 \Rightarrow \\ \exists J : \text{Point}, rk(\{A, B, J\}) = rk(\{C, D, J\}) = 2$$

(A7) Rk-Three-Points:

$$\forall A B : \text{Point}, \exists C : \text{Point}, \\ rk(\{A, B, C\}) = rk(\{B, C\}) = rk(\{A, C\}) = 2$$

(A8) Rk-Lower-Dimension:

$$\exists A B C D : \text{Point}, rk(\{A, B, C, D\}) \geq 4$$

(A9) Rk-Upper-Dimension:

$$\forall A B C D : \text{Point}, rk(\{A, B, C, D\}) \leq 4$$

Table 4: Rank axiom system for projective space geometry

We successfully applied the approach based on ranks to prove a 2D version of Desargues property in a 3D setting [16]. The proof was performed interactively: all subsets of points involved in the proof as well as their ranks has to be determined by the user.

In the next section, we show how to do that automatically.

5 AN AUTOMATED PROVER BASED ON MATROID THEORY

Using the formal description of projective geometry using ranks presented in the previous section and the idea that the rank of a set of

- (RS1) if $X \subseteq Y$ and $rkMin(X) > rkMin(Y)$ then $rkMin(Y) \leftarrow rkMin(X)$
- (RS2) if $Y \subseteq X$ and $rkMin(Y) > rkMin(X)$ then $rkMin(X) \leftarrow rkMin(Y)$
- (RS3) if $X \subseteq Y$ and $rkMax(Y) < rkMax(X)$ then $rkMax(X) \leftarrow rkMax(Y)$
- (RS4) if $Y \subseteq X$ and $rkMax(X) < rkMax(Y)$ then $rkMax(Y) \leftarrow rkMax(X)$
- (RS5) if $rkMax(X) + rkMax(Y) - rkMin(X \cap Y) < rkMax(X \cup Y)$ then $rkMax(X \cup Y) \leftarrow (rkMax(X) + rkMax(Y) - rkMin(X \cap Y))$
- (RS6) if $rkMax(X) + rkMax(Y) - rkMin(X \cup Y) < rkMax(X \cap Y)$ then $rkMax(X \cap Y) \leftarrow (rkMax(X) + rkMax(Y) - rkMin(X \cup Y))$
- (RS7) if $rkMin(X \cap Y) + rkMin(X \cup Y) - rkMax(Y) > rkMin(X)$ then $rkMin(X) \leftarrow (rkMin(X \cap Y) + rkMin(X \cup Y) - rkMax(Y))$
- (RS8) if $rkMin(X \cap Y) + rkMin(X \cup Y) - rkMax(X) > rkMin(Y)$ then $rkMin(Y) \leftarrow (rkMin(X \cap Y) + rkMin(X \cup Y) - rkMax(X))$

Table 5: Rules used until saturation.

points is always bounded by a minimum and a maximum value, we build an automated prover based on rank interval computations. For each subset of the powerset of the geometric configuration at stake, we define the minimum rank and the maximum rank (in the worst case, when no information is known, the rank of each non-empty subset is between 1 and 4).

We translate the matroid axioms of Table 2 into the set of rewrite rules given in Table 5. Axiom (A1) is implemented as the initialization condition for each non-empty set of points (their rank is between 1 and the number of points of the set). Axioms (A2) and (A3) are implemented by the rules presented in Table 5. Each axiom leads to four rules, two for the minimum rank and two for the maximum rank. These rules can be locally used in an interval arithmetic way to incrementally reduce the difference between the maximum and the minimum ranks for each subset. To do this, a saturation algorithm is launched on a valuated graph implementing the inclusion lattice of the point powerset, labeled by the minimum and maximum rank.

This simple algorithm is efficient for small configurations—this method aims at helping mathematicians to carry out small pieces of proof automatically—but the complexity is exponential by essence.

The prover is implemented in C++, as an independent tool. It takes as input a geometric configuration (a set of points and the ranks of some of its subsets), performs the entire saturation each time it is called, and returns a Coq proof script which can then be processed and verified by Coq. More technical details of the implementation and the complexity of the algorithm are provided in David Braun's PhD thesis [5, Chap. III.1]. The current version of the prover can be found at <https://github.com/pascalschreck/MatroidIncidenceProver>.

The first step of the algorithm consists of building the initial graph by considering the set of points. The incidence constraints—which are translated into ranks according to the definition given in Table 3—are used to assign partial data to the minimum rank and the maximum rank for each subset of points. For instance, in Table 6 the statement of Dandelin-Gallucci's theorem is translated in terms of equality between ranks of some subsets; these values are used to fill the valuated graph implementing the inclusion lattice.

In a second step, we run the saturation algorithm. The algorithm proceeds as follows: several traversals of the graph are performed taking every pair of subsets into consideration and applying the rules of Table 5 in order to tighten the interval between the minimum and maximum possible values for the rank. Any time a new deduction is made, the system updates the lattice accordingly. The bounds $rkMin$ and $rkMax$ of some of the subsets involved in the deduction are updated to ensure the statements of Table 5 still hold.

This process continues until a contradiction appears, usually the value of a minimal rank becomes greater than the value of the corresponding maximal rank, or the search is successful, that is, the rank of the set corresponding to the conclusion becomes equal to its minimum rank and to its maximal rank. Apart from those two cases, the failure of the algorithm occurs when no more rules can be applied.

Throughout this process, the prover records the application of each rule. The third step of our algorithm consists in rebuilding the trace of the deductions leading to the computation of a new rank for a given set of points. The trace is then extracted as a certificate and fed to the Coq proof assistant which verifies and validates it.

Example. As an example of our approach, let us prove a very simple theorem. Having four different points A, B, C and D in the plane, prove that if A, B and C are collinear, and A, B and D are collinear, then points B, C and D are collinear too. In terms of ranks, we have as hypotheses:

- the rank of each pair of points is 2,
- the maximum rank is 3 (we consider a plane configuration),
- $rk(\{A, B, C\}) = 2$,
- $rk(\{A, B, D\}) = 2$,
- $1 \leq rk(\{A, B, C, D\}) \leq 3$,
- $1 \leq rk(\{B, C, D\}) \leq 3$

and we want to prove that $rk(\{B, C, D\}) = 2$. We derive the successive deductions:

- (1) $2 \leq rk(\{A, B, C, D\}) \leq 3$ by rule RS1
- (2) $2 \leq rk(\{A, B, C, D\}) \leq 2$ by rule RS5,
that is $rk(\{A, B, C, D\}) = 2$
- (3) $2 \leq rk(\{B, C, D\}) \leq 3$ by rule RS1
- (4) $2 \leq rk(\{B, C, D\}) \leq 2$ by rule RS3
using $rk(\{A, B, C, D\}) = 2$

Therefore $rk(\{B, C, D\}) = 2$ and the algorithm succeeds.

Several issues should be noted. First, the algorithm always ends, but its average-case complexity in time and in space is at least exponential. Clever data structures, decomposition of the matroid structure and coloration heuristics can be used to successfully reach configurations with about twenty points. But still, it is an algorithm with exponential complexity. Second, and more challenging, the axioms with existential quantification are not taken into account in

the rewrite rules. Actually all the points needed for the proof have to be present in the statement before its translation into a graph, and no construction of auxiliary points can be made by this algorithm. As it is, it cannot be used for clever proofs which need to consider new points. The algorithm simply performs a combinatorial search.

We use this prover to prove the equivalence of Dandelin-Gallucci's and Pappus' properties. Table 6 presents the lemma for the Pappus to Dandelin-Gallucci implication and Table 7 presents the reciprocal statement Dandelin-Gallucci to Pappus.³ Both statements and proofs can be found in the git repository already mentioned. Note that only the basic axioms of incidence geometry are used by our prover, and, for instance, the Pappus configuration which define points X , Y and Z is not discovered by our method. On the contrary, the user has to be smart enough to discover the appropriate configuration. The collinearity property from which the existence of point R is deduced can be proved (see the git repository mentioned above), but the addition of that point to the statement is performed interactively by applying manually an existence axiom. All these steps are then formally checked by Coq.

We have the same observation for the statement describing the direction from “Dandelin-Gallucci” to “Pappus”. The point R and its associated properties are provided outside the prover and included by hand, as the reader can see in Table 7.

Termination, correctness and validation. Our automatic prover terminates by construction. But on the one hand, as mentioned above, it can fail, even if the property to be proved is valid. This is the case, for instance if some auxiliary points needed in the proof are not provided by the user. Our prover is therefore not complete. On the other hand, when it succeeds, there is no formal guarantee that it performed a correct deduction. That is why it generates a trace which is then proof-checked using the Coq proof assistant. Correct proofs are only those accepted during the proof-checking process carried out in Coq.

Results and discussion. The Coq proof script generated by the prover for the implication Pappus' property to Dandelin-Gallucci's property is about 25,000 lines long with about 330 auxiliary lemmas. Finding a solution and generating the associated Coq script takes about 20 minutes of computations and Coq takes less than 2 minutes to check the proof.⁴ Proving the other way, Dandelin-Gallucci to Pappus, produces a 60,000 lines long Coq proof, with about 700 lemmas, in 32 minutes and it is checked by Coq in about 4 minutes.

This is the biggest example that the prover managed to handle so far. However, we envision the prover as a tool to help mathematicians proving tedious but immediate lemmas within Coq, rather than proving big theorems in one go. Compared to Wu's method, a big weakness of the prover is that the user has to provide auxiliary points explicitly. Indeed, this task is very smart and is not well-suited for automation. The main advantage of our approach is that it is highly combinatorial and neither complex computations are required nor coordinates are considered.

³Lemmas are named automatically using L, followed by the name of the points involved in the conclusion of the statement.

⁴Using an Intel(R) Core(TM) i7-4790 CPU @3.60GHz with 16GB of memory

```
Lemma LCCpSpTUV :
forall Oo A B C Ap Bp Cp X Y Z M N Sp T U V R,
  rk(Oo :: A :: nil) = 2 -> rk(Oo :: B :: nil) = 2
-> rk(A :: B :: nil) = 2 -> rk(Oo :: C :: nil) = 2
-> rk(A :: C :: nil) = 2 -> rk(B :: C :: nil) = 2
-> rk(Oo :: A :: B :: C :: nil) = 2      (* line a*)
-> rk(Oo :: Ap :: nil) = 2 -
-> rk(Oo :: A :: Ap :: nil) = 3          (* a != e*)
-> rk(Oo :: Bp :: nil) = 2
-> rk(Ap :: Bp :: nil) = 2
-> rk(Oo :: Cp :: nil) = 2
-> rk(Ap :: Cp :: nil) = 2
-> rk(Bp :: Cp :: nil) = 2 ->
  rk(Oo :: Ap :: Bp :: Cp :: nil) = 2      (* line e*)
                                      (* Pappus point X *)

-> rk(B :: Ap :: X :: nil) = 2
-> rk(A :: Bp :: X :: nil) = 2
                                      (* Pappus point Y *)

-> rk(C :: Ap :: Y :: nil) = 2
-> rk(A :: Cp :: Y :: nil) = 2
-> rk(C :: Bp :: Z :: nil) = 2      (* Pappus point Z *)
-> rk(B :: Cp :: Z :: nil) = 2
-> rk(X :: Y :: Z :: nil) = 2      (* Pappus prop. *)
-> rk(Ap :: M :: Sp :: nil) = 2 ->      (* line b *)
  rk(Oo :: A :: B :: C :: Ap :: M :: Sp :: nil) = 4
-> rk(Bp :: N :: T :: nil) = 2 ->      (* line c *)
  rk(Oo :: A :: B :: C :: Bp :: N :: T :: nil) = 4
-> rk(C :: Sp :: T :: nil) = 2 ->      (* line d *)
  rk(Oo :: C :: Ap :: Bp :: Cp :: Sp :: T :: nil) = 4
-> rk(Ap :: Bp :: M :: N :: Sp :: T :: nil) = 4
-> rk(A :: M :: U :: nil) = 2 ->      (* line f *)
  rk(Oo :: A :: Ap :: Bp :: Cp :: M :: U :: nil) = 4
-> rk(A :: C :: M :: Sp :: T :: U :: nil) = 4
-> rk(B :: N :: V :: nil) = 2 ->      (* line g *)
  rk(Oo :: B :: Ap :: Bp :: Cp :: N :: V :: nil) = 4
-> rk(B :: C :: N :: Sp :: T :: V :: nil) = 4 ->
-> rk(Cp :: U :: V :: nil) = 2      (* line h *)
-> rk(Oo :: A :: B :: C :: Cp :: U :: V :: nil) = 4
-> rk(A :: B :: M :: N :: U :: V :: nil) = 4
-> rk(Ap :: Cp :: M :: Sp :: U :: V :: nil) = 4
-> rk(Bp :: Cp :: N :: T :: U :: V :: nil) = 4
-> rk(Y :: M :: R :: nil) = 2      (* point R *)
-> rk(Z :: N :: R :: nil) = 2
->
                                      (*coplanarity*)
  rk(C :: Cp :: Sp :: T :: U :: V :: nil) = 3.
Proof.
```

Table 6: Pappus property implies Dandelin-Gallucci's property (Coq statement using ranks).

6 CONCLUSION

The main contribution of this paper is to present two new and very different methods to prove Dandelin-Gallucci's theorem, compare them and examine, for each approach, how far it can be mechanized. We first summarized the synthetic proof, following Horváth's description [15]. We then show how to automate the proof as much as possible using two different approaches: an algebraic one based on Wu's method and a combinatorial one based on a matroid representation of incidence projective geometry.


```

Lemma LXYZ : forall Oo A B C Ap Bp Cp P X
Y Z Q A1 B1 Ap1 Bp1 R ,
rk(A :: B :: nil) = 2 ->
rk(A :: C :: nil) = 2 -> rk(B :: C :: nil) = 2 ->
rk(Oo :: A :: B :: C :: nil) = 2 ->
rk(Oo :: A :: Ap :: nil) = 3 ->
rk(Oo :: B :: Bp :: nil) = 3 ->
rk(Ap :: Bp :: nil) = 2 ->
rk(Oo :: C :: Cp :: nil) = 3 ->
rk(Ap :: Cp :: nil) = 2 ->
rk(Bp :: Cp :: nil) = 2 ->
rk(Oo :: Ap :: Bp :: Cp :: nil) = 2 ->
rk(Oo :: A :: B :: C ::
  Ap :: Bp :: Cp :: P :: nil) = 4 ->
rk(B :: Ap :: X :: nil) = 2 ->
rk(A :: Bp :: X :: nil) = 2 ->
rk(C :: Ap :: Y :: nil) = 2 ->
rk(A :: Cp :: Y :: nil) = 2 ->
rk(C :: Bp :: Z :: nil) = 2 ->
rk(B :: Cp :: Z :: nil) = 2 ->
rk(P :: Q :: nil) = 2 ->
rk(X :: Q :: nil) = 2 ->
rk(P :: X :: Q :: nil) = 2 ->
rk(A :: P :: A1 :: nil) = 2 ->
rk(A :: Cp :: P :: A1 :: nil) = 3 ->
rk(B :: Q :: B1 :: nil) = 2 ->
rk(Cp :: A1 :: B1 :: nil) = 2 ->
rk(Ap :: P :: Ap1 :: nil) = 2 ->
rk(C :: Ap :: P :: Ap1 :: nil) = 3 ->
rk(Bp :: Q :: Bp1 :: nil) = 2 ->
rk(C :: Ap1 :: Bp1 :: nil) = 2 ->
rk(Cp :: A1 :: B1 :: R :: nil) = 2 ->
rk(C :: Ap1 :: Bp1 :: R :: nil) = 2
->
rk(X :: Y :: Z :: nil) = 2.    (* Pappus col. *)
Proof.

```

Table 7: Dandelin-Gallucci property implies Pappus property (Coq statement using ranks).

Few authors discussed automated theorem proving in 3D geometry, see for instance [4, 20, 21]. Their approaches are mainly based on algebra and, thus they have to translate geometric statements into algebraic systems and to choose appropriate coordinates rather than using an axiomatic approach. Thus they faced the problems that we described in section 3, but their approach to deal with them is not described. Moreover, only simple theorem are proved, for instance the 3D version of Desargues's theorem which does not involve auxiliary points and is easily proved by our combinatorial prover.

There are several directions to continue the work in both approaches. For instance, as far as we know, the mechanization of proofs of other challenging 3D geometric theorems using a purely synthetic geometric has not yet been tackled. Regarding the algebraic approach, a lot of work must be done in order to fully automate the proofs, for instance about the translation from geometry to algebra and the management of special cases. Finally, our combinatorial approach is powerful in the sense that it produces fully verified proofs. However it suffers from an exponential complexity and we keep

working on improving the proof of concept presented in this paper to make it easily usable in practice.

REFERENCES

- [1] Mohammadali Asadi, Alexander Brandt, Robert H. C. Moir, Marc Moreno Maza, and Yuzhen Xie. On the parallelization of Triangular Decompositions. In Ioannis Z. Emiris and Lihong Zhi, editors, *ISSAC '20: International Symposium on Symbolic and Algebraic Computation, Kalamata, Greece, July 20-23, 2020*, pages 22–29. ACM, 2020.
- [2] Yves Bertot and Pierre Castéran. *Interactive Theorem Proving and Program Development, Coq'Art: The Calculus of Inductive Constructions*. Springer, 2004.
- [3] Pierre Boutry, Gabriel Braun, and Julien Narboux. Formalization of the arithmetization of euclidean plane geometry and applications. *J. Symb. Comput.*, 90:149–168, 2019.
- [4] Thierry Boy de la Tour, Stéphane Fèvre, and Dongming Wang. Clifford Term Rewriting for Geometric Reasoning in 3D. In Xiao-Shan Gao, Dongming Wang, and Lu Yang, editors, *Automated Deduction in Geometry, Second International Workshop, ADG'98, Beijing, China, August 1-3, 1998, Proceedings*, volume 1669 of *Lecture Notes in Computer Science*, pages 130–155. Springer, 1998.
- [5] David Braun. *Approche combinatoire pour l'automatisation en Coq des preuves formelles en géométrie d'incidence projective*. PhD thesis, Université de Strasbourg, sept. 2019.
- [6] David Braun, Nicolas Magaud, and Pascal Schreck. Two cryptomorphic formalizations of projective incidence geometry. *Ann. Math. Artif. Intell.*, 85(2-4):193–212, 2019.
- [7] Changbo Chen and Marc Moreno Maza. Algorithms for computing triangular decompositions of polynomial systems. *Journal of Symbolic Computation*, 47(6):610–642, 2012.
- [8] Shang-Ching Chou. *Mechanical Geometry Theorem Proving*. Kluwer Academic Publishers, Norwell, MA, USA, 1987.
- [9] Pasqualina Conti and Carlo Traverso. A case of automatic theorem proving in euclidean geometry: the maclean 8₃ theorem. In Gérard D. Cohen, Marc Giusti, and Teo Mora, editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, 11th International Symposium, AAECC-11, Paris, France, July 17-22, 1995, Proceedings*, volume 948 of *Lecture Notes in Computer Science*, pages 183–193. Springer, 1995.
- [10] Coq development team. *The Coq Proof Assistant Reference Manual, Version 8.9*. LogiCal Project, 2019.
- [11] Jack Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1(1):127–136, 1971.
- [12] Joran Elias. Automated Geometric Theorem Proving: Wu's Method. *The Montana Mathematics Enthusiast*, 3(1):3–50, February 2006.
- [13] Jean-David Génevaux, Julien Narboux, and Pascal Schreck. Formalization of Wu's Simple Method in Coq. In Jean-Pierre Jouannaud and Zhong Shao, editors, *Certified Programs and Proofs, CPP 2011, Kenting, Taiwan, December 7-9, 2011, Proceedings*, volume 7086 of *LNCS*, pages 71–86. Springer, 2011.
- [14] Ahmed Guergueb, Jean Mainguené, and Marie-Françoise Roy. Examples of automatic theorem proving a real geometry. In Malcolm A. H. MacCallum, editor, *Proceedings of the International Symposium on Symbolic and Algebraic Computation, ISSAC '94, Oxford, UK, July 20-22, 1994*, pages 20–24. ACM, 1994.
- [15] Ákos G Horváth. Gallucci's axiom revisited. In *arXiv preprint arXiv:1712.04800*, 2017.
- [16] Nicolas Magaud, Julien Narboux, and Pascal Schreck. A Case Study in Formalizing Projective Geometry in Coq: Desargues Theorem. *Computational Geometry: Theory and Applications*, 45(8):406–424, 2012.
- [17] Dominique Michelucci and Pascal Schreck. Incidence Constraints: a Combinatorial Approach. *Int. Journal of Computational Geometry and Applications*, 16(5-6):443–460, 2006.
- [18] James G Oxley. *Matroid Theory*, volume 3. Oxford University Press, USA, 2006.
- [19] Gleb Pogudin and Ágnes Szántó. Irredundant triangular decomposition. In Manuel Kauers, Alexey Ovchinnikov, and Éric Schost, editors, *Proceedings of the 2018 ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC 2018, New York, NY, USA, July 16-19, 2018*, pages 311–318. ACM, 2018.
- [20] Eugenio Roanes-Lozano, Nicolas van Labeke, and Eugenio Roanes-Macías. Connecting the 3D DGS Calques3D with the CAS Maple. *Math. Comput. Simul.*, 80(6):1153–1176, 2010.
- [21] Eugenio Roanes-Macías and Eugenio Roanes-Lozano. A Maple Package for Automatic Theorem Proving and Discovery in 3D-Geometry. In Francisco Botana and Tomás Recio, editors, *Automated Deduction in Geometry, 6th International Workshop, ADG 2006, Pontevedra, Spain, Aug. 31-Sept. 2, 2006. Revised Papers*, volume 4869 of *Lecture Notes in Computer Science*, pages 171–188. Springer, 2006.
- [22] William Thomas Tutte. Matroids and graphs. *Transactions of the American Mathematical Society*, 90(3):527–552, 1959.
- [23] W.T. Wu. Basic principles of mechanical theorem proving in elementary geometries. *Journal of Symbolic Computation*, 4:207–235, 1984.