# Interrogating witnesses for Geometric Constraint Solving

Dominique Michelucci
Laboratoire Electronique, Informatique et Image
LE2I, UMR CNRS 5158, Dijon, France
Dominique.Michelucci@u-bourgogne.fr

Sebti Foufou
Laboratoire Electronique, Informatique et Image
LE2I, UMR CNRS 5158, Dijon, France
Sebti.Foufou@u-bourgogne.fr

## ABSTRACT
Classically, geometric constraint solvers use graph-based methods to analyze systems of geometric constraints. These methods have intrinsic limitations, which the witness method overcomes. This paper details the computation of a basis of the vector space of the free infinitesimal motions of a typical witness, and explains how to use this basis to interrogate the witness for detecting *all* dependencies between constraints: structural dependencies already detectable by graph-based methods, and also non-structural dependencies, due to known or unknown geometric theorems, which are undetectable with graph-based methods. The paper also discusses how to decide about the rigidity of a witness.

## Categories and Subject Descriptors
I.3.5 [**Applications**]: Computational Geometry and Object Modeling—*Geometric Constrait Solving*; G.1.3 [**Numerical Analysis**]: Numerical Linear Algebra

## Keywords
Geometric Constraint Solving, dependency Detection

## 1. INTRODUCTION
Shape modelling based on geometric constraints enables the designer to specify shapes as sets of geometric entities with their constraints and relationships. Geometric constraints are specifications of distances, angles, incidences, tangencies, parallelisms, orthogonalities, etc. between geometric elements such as points, lines, planes, conics, quadrics, or higher degree algebraic curves and surfaces. Examples of big clients of geometric constraints are: robotics (*e.g.,* generalized Stewart platform), molecular chemistry (*e.g.,* the molecule problem which consists in finding the configurations of a molecule from interatomic distances), geometric modelling for CAD-CAM (dimensioning mechanical parts), and virtual reality (*e.g.,* blending surfaces) [14, 1, 11, 5].

Graph-based methods work well for correct systems of constraints, and they indeed make possible the solving of systems which are intractable otherwise. These methods are even able to detect simple mistakes in systems of constraints, namely structural dependencies, which may usually occur when a subset of unknowns is constrained by too much constraints, non-structural dependencies, due to geometric theorems, can not be detected with pure graph-based methods in polynomial time. Missing to detect such dependencies makes the solver fail to solve the system, and to give a relevant explanation to the designer [13]. This is a serious drawback as the probability of existence of such dependencies increases with the size of the system to solve.

For CAD-CAM problems, assuming a witness configuration [3, 13] is available, it is possible to check the independency between the geometric constraints, or to check that a decomposition proposed by any other method is correct [6]. Systems of geometric constraints reduce to solving systems of equations of the form $F(U, X) = 0$ where $U$ is a vector of parameters composed of geometric (*e.g.,* distances, angles) and non-geometric (*e.g.,* weights, forces, costs) entities.

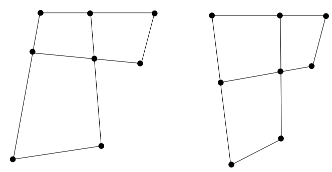

**Figure 1: A target and a witness configurations, in 2D. Constraints are collinearities, and some edges lengths or some angles.**

The target $(U_T, X_T)$ is the system to solve: $U_T$ are the values of parameters $U$, and $X_T$ is the corresponding root, so that $F(U_T, X_T) = 0$. A witness is a numeric couple $(U_W, X_W)$ such that $F(U_W, X_W) = 0$, and $U_W$ and $U_T$ are generally different. Fig. 1 shows a target configuration and a possible witness configuration. A witness $(U_W, X_W)$ is typical of the target $(U_T, X_T)$ if they share the same combinatorial properties. When no witness is available, a witness can be computed by considering $U$, the vector of parameters, as unknowns and solving the under-constrained $F(Y, X) = 0$ system with the solver in [4].

This paper introduces the vector space of free infinitesimal motions of the witness, how to compute a basis of this vector space, and how to use this basis to detect constraint dependencies. The paper is structured as follows. §2 presents the principle of the witness method. §3 presents the notion of free infinitesimal motions and how they are computed with rank considerations. §4 explains the interrogation of the witness method for testing flexibility and rigidity. For conciseness, §3 and §4 assume that the witness is typical. §5 gives the conclusion and some ideas for future improvements of this work.

## 2. THE WITNESS METHOD AND GCS

The goal of the witness approach is to help the designer build correct systems of constraints [13]. When a system is correct, the numeric solver in use can reliably solve it in a numerically stable way.

A numerical solver can reliably compute a root in $\mathbb{R}^N$ as the intersection point between the $N$ hypersurfaces described by the $N$ equations of the system only when the hypersurfaces intersect *transversely* [12], *i.e.,* when the tangent hyperplanes at the root intersect *transversely.* This means that the $N$ normal vectors to the tangent hyperplanes are linearly independent, *i.e.,* the Jacobian has full rank at the solution point.

The witness method takes into account the complications due to the intrinsic under-constrainedness of the systems of geometric constraints which have less equations than unknowns. For instance, a triangle in the plane is well constrained by three constraints (*e.g.,* three lengths), but there are six unknown coordinates $x_i, y_i,\ i = 1, 2, 3$. For conciseness, we assume that the available witness is typical of the target: they share the same combinatorial properties. The witness method basically computes the Jacobian structure at the witness; it detects subsets of hypersurfaces which do not intersect transversely, *i.e.,* subsets of equations having dependent gradient vectors. We think that transversality is definitively the good criterium. It has the convenient features listed below.

- Transversality of the witness (thus of the target) is decidable in polynomial time; it requires only standard tools from linear algebra.

- Transversality guarantees the convergence of the numerical solver in some neighborhood of the root (for the witness, and thus for target). Then classical methods from interval analysis compute such a neighborhood (a box, that is a vector of intervals), and provide guarantees.

- Transversality also guarantees that the root (the witness, or the target) is stable against small perturbations of the values of parameters $U$ in the system $F(U, X) = 0$, more precisely it guarantees that the root is locally an implicit, continuous, and differentiable function of parameters $U$; interval analysis is able to compute and guarantee such a neighborhood for $U$ and $X$.

- When the equations are transversal at the witness, but there is no root for the parameters values of the target, there is something wrong with these target parameters (*e.g.,* they violate a triangular inequality).

- There are numerous ways to translate constraints into equations, and to choose unknowns. But, the mapping between two distinct formulations is locally (*i.e.,* in some neighborhood of the witness, and thus in some neighborhood of the target) a diffeomorphism, that is a bijective, continuous and differentiable mapping. Transversality is preserved by diffeomorphisms, thus it is preserved through variations in the translation of constraints into equations.

## 3. FREE INFINITESIMAL MOTIONS

In geometric constraint-based modelling, the constraints control the shape of the configuration, and then the only permitted actions are motions that can be applied to transform the modelled configuration without violating the constraints. These infinitesimal **motions** are usually classified in two types: (i) infinitesimal **displacements**, namely translations, rotations and their compositions, which never deform the configuration, and which apply to all geometric elements in the configuration; (ii) infinitesimal **flexions** (sometimes called deformations). Clearly, the witness is flexible if it admits an infinitesimal flexion, *i.e.,* the system does not determine completely the geometric configuration.

Flexions can be displacements which apply to only a strict part of the configuration: *e.g.,* a subpart can rotate, while the rest of the configuration is fixed. Generic flexions deform the configuration. Degenerate flexions do not; they occur with atypical witnesses; an example of an atypical witness is a triangle with collinear vertices in the witness, though the collinearity is not due to the constraints. The set of atypical witnesses has measure zero, in the set of possible witnesses.

Assume a witness $(U_W, X_W)$ is known, *i.e.,* $F(U_W, X_W) = 0$, the main idea of the witness method is to compute the vector space of the free infinitesimal motions $\dot{X}$ of the witness, such that the perturbed witness $X_W + \epsilon\dot{X}$, where $\epsilon$ is an infinitesimally small number, still fulfils the specified constraints: $F(U_W, X_W + \epsilon\dot{X}) = 0$. Taylor expansion gives $F(U_W, X_W+\epsilon\dot{X}) = F(U_W, X_W)+\epsilon F'(U_W, X_W)\dot{X}^t+O(\epsilon^2)$. Thus, for $F(U_W, X_W+\epsilon\dot{X})$ to be $O(\epsilon^2)$, infinitesimally small in front of the perturbation $\epsilon$, the term $F'(U_W, X_W)\dot{X}^t$ must vanish: the vector space of the free motions is the kernel of the Jacobian matrix $F'(U_W, X_W)$ at the witness.

A basis of the infinitesimal displacements is computable *a priori*: it does not depend on the constraints, but only on the variables. Such a basis is provided below in §3.1. The following conventions are used to describe the unknowns. In 2D, a point has coordinates $(x, y)$; a line with equation $ax + by + c = 0$ is represented by a tuple $(a, b, c)$; a vector is represented by its coordinates $(u, v)$; this distinction between points and vectors is because a translation (including an infinitesimal translation) modifies the $(x, y)$ of points, but it does not modify the $(u, v)$ of vectors; similarly translations do not modify the $a, b$ coefficients of lines, but they modify the $c$ coefficient. Under displacements, the variables $u, v$ and $a, b$ behave in the same way. Other geometric unknowns (barycentric coordinates, scalar products, distances, squared distances, angle cosines or squared cosines, or other trigonometric functions, areas, volumes) are unchanged by infinitesimal displacements, so the corresponding entries in all vectors of the basis are 0. This holds for all non-geometric unknowns (weights, costs, densities, temperatures…).

| | $\dot{x}_i$ | $\dot{y}_i$ | $\dot{a}_l$ | $\dot{b}_l$ | $\dot{c}_l$ | $\dot{u}_k$ | $\dot{v}_k$ |
|---|---|---|---|---|---|---|---|
| $t_x$ | 1 | 0 | 0 | 0 | $-a_l$ | 0 | 0 |
| $t_y$ | 0 | 1 | 0 | 0 | $-b_l$ | 0 | 0 |
| $r_{xy}$ | $-y_i$ | $x_i$ | $-b_l$ | $a_l$ | 0 | $-v_k$ | $u_k$ |

| | $\dot{x}_i$ | $\dot{y}_i$ | $\dot{z}_i$ | $\dot{a}_h$ | $\dot{b}_h$ | $\dot{c}_h$ | $\dot{d}_h$ | $\dot{u}_k$ | $\dot{v}_k$ | $\dot{w}_k$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $t_x$ | 1 | 0 | 0 | 0 | 0 | 0 | $-a_h$ | 1 | 0 | 0 |
| $t_y$ | 0 | 1 | 0 | 0 | 0 | 0 | $-b_h$ | 0 | 1 | 0 |
| $t_z$ | 0 | 0 | 1 | 0 | 0 | 0 | $-c_h$ | 0 | 0 | 1 |
| $r_{xy}$ | $-y_i$ | $x_i$ | 0 | $-b_h$ | $a_h$ | 0 | 0 | $-v_k$ | $u_k$ | 0 |
| $r_{xz}$ | $-z_i$ | 0 | $x_i$ | $-c_h$ | 0 | $a_h$ | 0 | $-w_k$ | 0 | $u_k$ |
| $r_{yz}$ | 0 | $-z_i$ | $y_i$ | 0 | $-c_h$ | $b_h$ | 0 | 0 | $-w_k$ | $v_k$ |

**Table 1: Left: a basis for the free displacements in 2D for points, lines, and vectors. Right: a basis for the free displacements in 3D for points, planes, and vectors.**

## 3.1 Basis of infinitesimal displacements

It is possible to compute an *a priori* basis of the infinitesimal displacements. Tab. 1 shows such a basis, in the 2D case, composed of $t_x$ a translation in the $x$ direction, $t_y$ a translation in the $y$ direction, and $r_{xy}$ a rotation around the origin. $(x_i, y_i)$ are coordinates of a point, $(a_l, b_l, c_l)$ are coordinates of a line (*i.e.,* the line has equation: $a_l x + b_l y + c_l = 0$), and $(u_k, v_k)$ are coordinates of a vector (the difference between two points). Dotted variables $\dot{x}_i, \dot{y}_i, \dot{a}_l, \dot{b}_l, \dot{c}_l, \dot{u}_k$, and $\dot{v}_k$ are used to denote the values of the corresponding coordinates in the basis of infinitesimal displacements, *e.g.,* the couple $(\dot{x}_i, \dot{y}_i)$ represents the infinitesimal translation $t_x$ along the $x$ axis of a point $(x_i, y_i)$, it is equal to $(1, 0)$ as this is shown on the first row of Tab. 1).

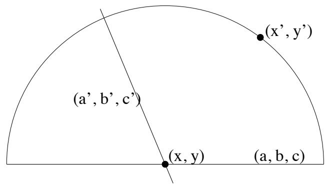


**Figure 2: A 2D under-constrained system of geometric constraints.**

Vectors $(u, v)$ are differences between two points, and thus $(\dot{u}, \dot{v})$ straightforwardly follows for all infinitesimal displacements. Note that the infinitesimal displacements for a point $(x, y)$, for a normal $(a, b)$ to a line, and for a vector $(u, v)$ are different; *e.g.,* translating a point modify it, but translating a vector or a normal does not. In 3D, a basis of the infinitesimal displacements is $t_x$, $t_y$, $t_z$, $r_{xy}$, $r_{xz}$, $r_{yz}$, where $t_z$ is a translation along $z$, $r_{yz}$, $r_{xz}$, $r_{xy}$ are rotations around the $x$, $y$, and $z$ axes. Corresponding coordinates are given in Tab. 1 Right, which has as many columns as unknowns.

## 3.2 A structurally under-constrained example

A simple structurally under-constrained example in 2D is the system of six equations shown in formula (1) and Fig. 2, with generic parameters $\delta$ (a distance) and $\lambda$ (a cosine). Point $(x, y)$ lies on two lines $(a, b, c)$ and $(a', b', c')$, with a specified angle between them. Moreover the distance between points $(x, y)$ and $(x', y')$ is specified. Tab. 2 shows the Jacobian and a basis for a set of infinitesimal motions composed of three displacements and one flexion: the point $(x', y')$ can rotate around the point $(x, y)$ (note that lines stay unchanged: $\dot{a} = \dot{b} = \dot{c} = \dot{a'} = \dot{b'} = \dot{c'} = 0$, thus this transform is not a rotation, but indeed a flexion). The reader can check that vectors of infinitesimal motions are orthogonal to the gradient vectors (the derivatives) $e'_1, \ldots e'_6$. A possible witness of this system is $(x = y = 0, x' = 3, y' = 4, \delta = 5, a = 1, b = 0, a' = 12/13, b' = 5/13, \text{and } \lambda = 12/13)$. To perform computations, the witness method replaces all variables $(x, y, x', y', a, b, c, a', b', c')$ with their numerical values at the witness in Tab. 2.

$$
\begin{array}{ll}
e_1: & ax + by + c = 0 \\
e_2: & a'x + b'y + c' = 0 \\
e_3: & (x - x')^2 + (y - y')^2 - \delta^2 = 0 \\
e_4: & a^2 + b^2 - 1 = 0 \\
e_5: & a'^2 + b'^2 - 1 = 0 \\
e_6: & aa' + bb' - \lambda = 0
\end{array}
\tag{1}
$$

## 3.3 Degrees of Displacements (DoD)

In an attempt to make graph-based methods more robust against dependencies between constraints Jermann *et al.* define degrees of rigidity [10]. We prefer the name: degrees of displacements. The degree of displacements of a rigid configuration is the number of equations needed to fix it in a cartesian coordinate system. The DoD is difficult to compute with pure graph-based methods. Jermann *et al.* mainly suggest formulas for big enough configurations and a tabulation for a finite set of small configurations; moreover the configurations need to be generic: incidence degeneracies (*e.g.,* collinearities, coplanarities) due to geometric theorems are forbidden.

The witness method computes straightforwardly the DoD, by interrogating the typical witness, and requires no genericity hypothesis at all: for instance, the typical witness can contain three collinear points; due to typicality, this collinearity is a consequence of the system of constraints, and holds for the target. The witness method can determine which infinitesimal displacements are dependent. Let $Y$ be a subset of $X$, the set of variables which describe the configuration, and $D$ be a basis of the infinitesimal displacements at the witness. The DoD of $Y$ is the rank of $D[Y]$, the subset of $D$ that is relevant to $Y$. Let us consider the computation of the DoD in the following cases:

- For a line $(a, b, c)$ in 2D, $D[Y] = D[a, b, c]$ is shown on Tab. 3. It is extracted from Tab. 1 by keeping only relevant variables. $D[Y]$ has rank 2: we can even see that the two dependent translations are $t_x$ and $t_y$, which is correct as a translation along a line leaves it unchanged.

- For a segment $Y = (x, y, z, x', y', z')$ in 3D, we just consider $D[Y]$ in the witness as it is shown on Tab. 3. In this case

| | $x$ | $y$ | $x'$ | $y'$ | $a$ | $b$ | $c$ | $a'$ | $b'$ | $c'$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $e'_1$ | $a$ | $b$ | 0 | 0 | $x$ | $y$ | 1 | 0 | 0 | 0 |
| $e'_2$ | $a'$ | $b'$ | 0 | 0 | 0 | 0 | 0 | $x$ | $y$ | 1 |
| $e'_3$ | $2(x-x')$ | $2(y-y')$ | $2(x'-x)$ | $2(y'-y)$ | 0 | 0 | 0 | 0 | 0 | 0 |
| $e'_4$ | 0 | 0 | 0 | 0 | $2a$ | $2b$ | 0 | 0 | 0 | 0 |
| $e'_5$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $2a'$ | $2b'$ | 0 |
| $e'_6$ | 0 | 0 | 0 | 0 | $a'$ | $b'$ | 0 | $a$ | $b$ | 0 |
| | $\dot{x}$ | $\dot{y}$ | $\dot{x'}$ | $\dot{y'}$ | $\dot{a}$ | $\dot{b}$ | $\dot{c}$ | $\dot{a'}$ | $\dot{b'}$ | $\dot{c'}$ |
| $t_x$ | 1 | 0 | 1 | 0 | 0 | 0 | $-a$ | 0 | 0 | $-a'$ |
| $t_y$ | 0 | 1 | 0 | 1 | 0 | 0 | $-b$ | 0 | 0 | $-b'$ |
| $r_{xy}$ | $-y$ | $x$ | $-y'$ | $x'$ | $-b$ | $a$ | 0 | $-b'$ | $a'$ | 0 |
| flx | 0 | 0 | $y-y'$ | $x'-x$ | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 2: The Jacobian and a basis of infinitesimal motions: three displacements and a flexion for the system given in (1). All variables are replaced by the numerical values of the witness.**

| | $\dot{a}$ | $\dot{b}$ | $\dot{c}$ |
|---|---|---|---|
| $t_x$ | 0 | 0 | $-a$ |
| $t_y$ | 0 | 0 | $-b$ |
| $r_{xy}$ | $-b$ | $a$ | 0 |

| | $\dot{x}$ | $\dot{y}$ | $\dot{z}$ | $\dot{x'}$ | $\dot{y'}$ | $\dot{z'}$ |
|---|---|---|---|---|---|---|
| $t_x$ | 1 | 0 | 0 | 1 | 0 | 0 |
| $t_y$ | 0 | 1 | 0 | 0 | 1 | 0 |
| $t_z$ | 0 | 0 | 1 | 0 | 0 | 1 |
| $r_{xy}$ | $-y$ | $x$ | 0 | $-y'$ | $x'$ | 0 |
| $r_{xz}$ | $-z$ | 0 | $x$ | $-z'$ | 0 | $x'$ |
| $r_{yz}$ | 0 | $-z$ | $y$ | 0 | $-z'$ | $y'$ |

| | $\dot{a}$ | $\dot{b}$ | $\dot{c}$ | $\dot{d}$ | $\dot{a}$ | $\dot{b}$ | $\dot{c}$ | $\dot{d}$ |
|---|---|---|---|---|---|---|---|---|
| $t_x$ | 0 | 0 | 0 | $-a$ | 0 | 0 | 0 | $-a'$ |
| $t_y$ | 0 | 0 | 0 | $-b$ | 0 | 0 | 0 | $-b'$ |
| $t_y$ | 0 | 0 | 0 | $-c$ | 0 | 0 | 0 | $-c'$ |
| $r_{xy}$ | $-b$ | $a$ | 0 | 0 | $-b'$ | $a'$ | 0 | 0 |
| $r_{xz}$ | $-c$ | 0 | $a$ | 0 | $-c'$ | 0 | $a'$ | 0 |
| $r_{yz}$ | 0 | $-c$ | $b$ | 0 | 0 | $-c'$ | $b'$ | 0 |

**Table 3: Left: a basis of infinitesimal displacements for a line. Middle: a basis of infinitesimal displacements for a segment. Right: a basis of infinitesimal displacements for two secant or parallel planes. Variables are replaced with their values at the witness.**

$D[Y]$ has rank 5; the three translations are independent; the three rotations are dependent, they have rank 2; this is correct as the rotation around a line supporting the segment leaves it unchanged.

- For two secant planes $Y = (a, b, c, d, a', b', c', d')$ in 3D, we also consider $D[Y]$ at the witness as it is shown on Tab. 3. It has rank 5; more precisely, the three translations have rank 2, the three rotations are independent. In the same way, we can compute the DoD of two parallel planes, which is four: the three translations have rank 2. So, the DoD of two planes depends on the planes configuration (are they secant or parallel). The DoD can not be computed reliably with graph-based methods, which have no way to decide correctly if the two planes are secant or parallel.

- Similarly, in 3D, the DoD of three collinear points is five (as for a segment), but the DoD of three non-collinear points is six. Again, the interrogation of the typical witness gives the correct answer, while graph-based methods have no way to decide if the three points are collinear or not. Remark that the three points may be collinear, not because of an explicit collinearity constraint (which some graph-based methods can account for, at least in theory), but because of a geometric theorem. Indeed, a lot of geometric theorems imply collinearities, *e.g.,* Desargues, Pappus. When a part has DoD three in 2D and six in 3D, we say that it has full DoD.

# 4. INTERROGATIONS OF A WITNESS

## 4.1 Are constraints coordinate-independent?

Correct geometric constraints are coordinate-independent. However, coordinate-dependent constraints such as $x_p = 0$ are sometimes needed to pin the configuration in the plane or in the 3D space, because numerical solvers expects systems with as many unknowns as equations.

A constraint is coordinate-dependent if its gradient vector is not orthogonal to at least one of the vectors in the basis of infinitesimal displacements. For instance, the constraint $x_p = 0$ is orthogonal to the vectors of the $y$ translation and of the rotation around the origin, but not to the vector of the $x$ translation. All equations can be tested this way. These tests are only numerical: the witness is a numerical vector, as the basis of infinitesimal displacements at the witness. In the remaining of this paper the equations are assumed to be coordinate-independent.

## 4.2 Are constraints dependent or independent?

Graph-based methods can detect only structural dependencies, as in the system: $f(x, y, z) = g(z) = h(z) = 0$ which over-constrains the unknown $z$. Interrogation of the witness makes possible the detection of non-structural dependencies. The constraints are dependent if the gradient vectors of the equations at the witness, *i.e.,* the Jacobian matrix at the witness, are dependent. It suffices to compute a basis of this Jacobian. It is also possible to tune pure graph-based methods to make them detect the simplest dependencies [14, 10, 8, 7]. However, the universal theorem makes intractable the problem of detecting all dependencies [13].

If the constraints are dependent, then the interrogation of the witness permits finding the smallest dependent set of constraints: this information is relevant to the user for removing the error from the system of constraints: remember that constraints can be numerous. This problem reduces to finding the minimal dependent set in a dependent set of vectors which are the gradient vectors of the equations at the witness. We assume that the rank of the dependent set is its cardinal minus one: typically, the last vector we try to add in the basis reduces to the null vector. So, the minimal dependent set is unique; to find it, just try to remove each

| | $x_O$ | $y_O$ | $x_A$ | $y_A$ | $x_B$ | $y_B$ | $x_C$ | $y_C$ |
|---|---|---|---|---|---|---|---|---|
| $e'_1$ | 2 | 0 | $-1$ | 0 | $-1$ | 0 | 0 | 0 |
| $e'_2$ | 0 | 2 | 0 | $-1$ | 0 | $-1$ | 0 | 0 |
| $e'_3$ | $2x_A - 2x_C$ | $2y_A - 2y_C$ | $2x_O - 2x_A$ | $2y_O - 2y_A$ | 0 | 0 | $2x_C - 2x_O$ | $2y_C - 2y_O$ |
| $e'_4$ | 0 | 0 | $x_B - x_A$ | $y_B - y_C$ | $x_A - x_C$ | $y_A - y_C$ | $2x_C - x_A - x_B$ | $2y_C - y_A - y_B$ |
| $e'_5$ | $2x_O - 2x_A$ | $2y_O - 2y_A$ | $2x_A - 2x_O$ | $2y_A - 2y_O$ | 0 | 0 | 0 | 0 |
| | $\dot{x_O}$ | $\dot{y_O}$ | $\dot{x_A}$ | $\dot{y_A}$ | $\dot{x_B}$ | $\dot{y_B}$ | $\dot{x_C}$ | $\dot{y_C}$ |
| $t_x$ | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| $t_y$ | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| $r_{xy}$ | $-y_O$ | $x_O$ | $-y_A$ | $x_A$ | $-y_B$ | $x_B$ | $-y_C$ | $x_C$ |
| flexion | 0 | 0 | 0 | 0 | 0 | 0 | $y_O - y_C$ | $x_C - x_O$ |

**Table 4: The Jacobian, and a basis of four free infinitesimal motions for the dependent system given in (2). The fourth motion is a flexion: point $C$ can rotate around $O$. Variables are replaced with their values at the witness.**

vector in the dependent set; if the set minus this vector is still dependent, then remove this vector. The remaining set of vectors is then the minimal dependent set. This greedy method can be proved with matroid theory [2, 9].

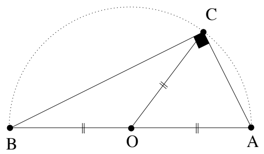


**Figure 3: Example of dependent constraints.**

## 4.3 Example of non-structural dependencies

Let us consider an example of dependency detection by witness interrogation. In 2D, suppose that we have four points $A, B, C$, and $O$ with the following constraints: (i) the distance $OA$ is specified by a parameter $u$, (ii) $O$ is the middle of the points $A$ and $B$, (iii) distances $OC$ and $OA$ are equal, and (iv) $AC$ and $BC$ are orthogonal (see Fig. 3). This last constraint results from the previous ones, this is due to a geometric theorem: if $C$ lies on the circle with diameter $AB$, then $AC$ and $BC$ are orthogonal. These constraints give the system of equations:

$$
\begin{aligned}
e_1 &: 2x_O - x_A - x_B = 0 \qquad (2)\\
e_2 &: 2y_O - y_A - y_B = 0\\
e_3 &: (x_C - x_O)^2 + (y_C - y_O)^2 - (x_A - x_O)^2 - (y_A - y_O)^2 = 0\\
e_4 &: (x_C - x_A)(x_C - x_B) + (y_C - y_A)(y_C - y_B) = 0\\
e_5 &: (x_A - x_O)^2 + (y_A - y_O)^2 - u^2 = 0
\end{aligned}
$$

A possible witness for this system of constraints is: $O = (0, 0), A = (-10, 0), B = (10, 0), C = (6, 8), u = 10$. Tab. 4 displays the Jacobian and a basis of the free infinitesimal motions: three displacements and a flexion, point $C$ can rotate around point $O$ (this is a flexion, not a rotation, because other points $A$ and $B$ are fixed). The rank of $e'_1, \dots e'_5$ computed at the witness is 4, thus equations are dependent.

Here is another 2D example, without equations for conciseness. By constraints, points $p_1, p_2, p_3$ are collinear, as well as $q_1, q_2, q_3$. Other constraints can specify lengths or angles. Due to Pappus' theorem, the three points $p_iq_j \cap p_jq_i, i \neq j$ are collinear. This collinearity holds in the target, and in all witnesses. The witness method indeed detects that the collinearity of the $p_iq_j \cap p_jq_i$ is a consequence of the constraints –though it can not say it is due to Pappus'theorem.

We underline that these non-structural dependencies are not detectable, in polynomial time, with graph-based methods.

## 4.4 Rigidity test

### 4.4.1 *Is the system flexible?*

Compute a basis of the kernel of the Jacobian at the typical witness: it is a basis of the free infinitesimal motions of the witness. If it contains vectors outside the vectorial space generated by the basis of the infinitesimal displacements, then the witness (and the target) is flexible. For instance, in the classical configuration of the double banana, the two bananas can rotate around the axis through their two common vertices [9]; the corresponding infinitesimal flexion is detected by the method. If the system is flexible, then the witness method can provide a basis of the infinitesimal flexions, and the set of maximal rigid subparts.

### 4.4.2 *Is a part rigid?*

A flexible system can contain rigid parts. A part is described by a subset $Y$ of unknowns. On table 1, each variable corresponds to a column, and a part $Y$ is thus a subset of columns. The part $Y$ is rigid if and only if the vector space $M[Y]$, the free infinitesimal motions in the columns $Y$, is equal to the vector space $D[Y]$, the free infinitesimal displacements in the columns $Y$). Vectors generating $M[Y]$ are obtained by taking only the columns $Y$ in the vectors of the basis of $M$. Similarly for $D[Y]$.

For instance, in the system defined by (1) and Tab. 2, $Y = \{x, y, a, b, c, a', b', c'\}$ is rigid, but $Y \cup \{x', y'\}$ is flexible: it does not depend on the basis chosen for $M$ and $D$. In example (2) with the Jacobian and a basis of infinitesimal motions shown on Tab. 4, the part $Y = \{x_O, y_O, x_A, y_A, x_B, y_B\}$ is rigid, while $Y \cup \{x_C\}$, $Y \cup \{y_C\}$, and $Y \cup \{x_C, y_C\}$ are not rigid. Again the rigidity is independent of the chosen basis.

### 4.4.3 *Are $A$ and $B$ relatively fixed?*

A flexible system can fix some pairs of geometric elements (two points, two lines, one point and one line, etc) relatively to each other. Actually, the previous section already provides a decision procedure. $A$ and $B$ are relatively fixed by the (possibly flexible) system if the part $Y = A \cup B$ is rigid.

## 4.5 All dependencies are detected

This section proves that the witness method detects all dependencies in algebraic systems, including non-structural dependencies due to geometric theorems, known or unknown. Structural dependencies are due to trivial theorems, they are detected as well; an example of a structural dependency is the over-constrainedness in $f(x, y, z) = g(z) = h(z) = 0$, where $(x, y, z) \in \mathbb{R}^3$.

All geometric theorems (Pappus, Pascal, Desargues, their duals, etc) relevant to geometric constraint solving are algebraically expressed by the fact that $f_1(x) = \dots f_n(x) = 0 \Rightarrow g(x) = 0$; here the $f_i(x) = 0$ express the hypothesis of the theorem, and $g(x) = 0$ is its conclusion. Algebraically, there are two possibilities for an algebraic equation $g(x) = 0$ to be a consequence of other algebraic equations $f_1(x) = \dots f_n(x) = 0$. In both cases, the method detects linear dependency in the Jacobian of the typical witness.

*First case.* In this case, $g$ is in the ideal of $(f_1, f_2, \dots f_n)$, so the gradient vector of $g$ at every common root $x$ of $f_1, f_2, \dots f_n$ is dependent on the gradient vectors $f'_1(x), \dots f'_n(x)$.

Proof. Let $F = (f_1, f_2, \dots f_n)$ be the polynomials of some algebraic system $F(x) = 0$. Let $g$ be a polynomial lying in the ideal generated by $f_1, f_2, \dots f_n$. Then, by definition, there are polynomials $\lambda_i$ such that $g = \lambda_1 f_1 + \dots + \lambda_n f_n$. Let $x$ be a root of $F$; then $x$ is also a root of $g$: $f_1(x) = \dots = f_n(x) = 0 \Rightarrow g(x) = \lambda_1(x) f_1(x) + \dots + \lambda_n(x) f_n(x) = 0$. After deriving we get $g'(x) = \lambda'_1(x) f_1(x) + \lambda_1(x) f'_1(x) + \dots \lambda'_n(x) f_n(x) + \lambda_n(x) f'_n(x) = \lambda_1(x) f'_1(x) + \dots \lambda_n(x) f'_n(x)$. The gradient vector of $g$ at $x$ lies in the vector space spanned by the gradient vectors $f'_1, \dots f'_n$ of $F$. In other words, $g'(x)$ does not intersect transversally $F'(x)$. $\Box$

*Second case.* In this case, $g$ is in the radical of $(f_1, f_2, \dots f_n)$, but not in the ideal, so the gradient vectors $f'_1(x), \dots f'_n(x)$ are linearly dependent at every common root.

Proof. The other possibility for the vanishing of $g$ to be a consequence of the vanishing of $f_1, \dots f_n$, is that $g$ lies in the radical generated by $(f_1, \dots f_n)$, *i.e.,* there is an integer $k \geq 2$ such that $g^k$ lies in the ideal generated by $(f_1, \dots f_n)$. Here, the fact that $g$ is in the radical of $(f_1, \dots f_n)$ does not imply that the gradient vector of $g$ at a root $x$ of $(f_1, \dots f_n)$ lies in the vector space spanned by the gradient vectors $F'$ of $F$ (*e.g.,* $g(x, y) = y$, $k = 2$, $f_1 = x^2 + y^2 - 1$, $f_2 = x^2 - 1$, so $g^k = f_1 - f_2$). But it implies that the gradient vectors of $f_1, \dots f_n$ at a common root $x$ are linearly dependent: deriving $-g^k + \sum \lambda_i f_i = 0$ yields $-kg^{k-1} g' + \sum \lambda'_i f_i + \sum \lambda_i f'_i = 0$. If $x$ is a common root of $(f_1, \dots f_n)$, it is also a root of $g$. Accounting for the fact that $k \geq 2$ (*i.e., g* is in the radical and not in the ideal), we obtain $\sum \lambda_i(x) f'_i(x) = 0$. Thus the gradient vectors $f'_1(x), \dots f'_n(x)$ are linearly dependent at the common root $x$, like the witness. $\Box$

# 5. CONCLUSION

Classical graph-based methods for decomposing systems of equations or constraints have intrinsic limitations: they can not detect in polynomial time all dependencies between constraints. This paper proves that the witness method does, assuming a typical witness is available. It shows how to interrogate the witness, and how all computations reduce to the polynomial time computation of the rank or a base of a set of numerical vectors. The witness method may broaden the scope of geometric constraint solving for CAD-CAM. An extension of this work is the investigation of sensitivity relatively to parameter values with applications to tolerancing.

# 6. REFERENCES


[1] B. Bruderlin and D. Roller, editors. *Geometric Constraint Solving and Applications*. Springer-Verlag, 1998.

[2] K. S. Christos H. Papadimitriou. *Combinatorial Optimization, algorithms and Complexity*. Dover Publications, 1998.

[3] S. Foufou, D. Michelucci, and J.-P. Jurzak. Numerical decomposition of geometric constraints. In *ACM Symp. on Solid and Physical Modelling*, pages 143–151, 2005.

[4] C. Fuenfzig, D. Michelucci, and S. Foufou. Nonlinear systems solver in floating-point arithmetic using lp reduction. In *Proceedings of the 2009 ACM symposium on Solid modeling*, 2009.

[5] X.-S. Gao, C. Hoffmann, and W. Yang. Solving spatial basic geometric constraint configurations with locus intersection. *Computer Aided Design*, 36(2):111–122, 2004.

[6] X.-S. Gao and G. Zhang. Geometric constraint solving via c-tree decomposition. In *ACM Solid Modelling*, pages 45–55. ACM Press, New York, 2003.

[7] C. Hoffmann, A. Lomonosov, and M. Sitharam. Decomposition plans for geometric constraint problems, Part II : New algorithms. *J. Symbolic Computation*, 31:409–427, 2001.

[8] C. Hoffmann, A. Lomonosov, and M. Sitharam. Decomposition plans for geometric constraint systems, Part I : Performance measures for cad. *J. Symbolic Computation*, 31:367–408, 2001.

[9] H. S. J. Graver, B. Servatius. *Combinatorial Rigidity. Graduate Studies in Mathematics.* American Mathematical Society, 1993.

[10] C. Jermann, B. Neveu, and G. Trombettoni. Algorithms for identifying rigid subsystems in geometric constraint systems. In *Int. Joint Conf. in Artificial Intelligence IJCAI-03*, pages 233–238, 2003.

[11] C. Jermann, G. Trombettoni, B. Neveu, and P. Mathis. Decomposition of geometric constraint systems: a survey. *Int. J. of Comp. Geometry and App. (IJCGA)*, 16(5-6):379–414, 2006.

[12] H. Lamure and D. Michelucci. Solving constraints by homotopy. In *Proc. of the Symp. on Solid Modeling Foundations and CAD/CAM Applications*, pages 263–269, May 1995.

[13] D. Michelucci and S. Foufou. Geometric constraint solving: the witness configuration method. *Computer Aided Design*, 38(4):284–299, 2006.

[14] J. Owen. Constraint on simple geometry in two and three dimensions. *Int. J. Comput. Geometry Appl.*, 6(4):421–434, 1996.