Computing Pure Conditions in Edge Coordinates

Zvi Rosen

Florida Atlantic University

Workshop on Real Algebraic Geometry
and Algorithms for Geometric Constraint Systems

The Fields Institute for Research in the Mathematical Sciences

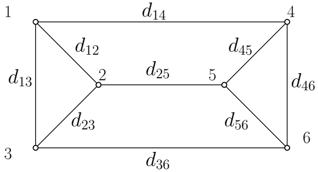
June 14, 2021

Goal of the Talk

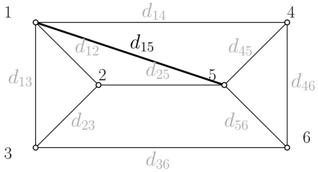
Based on joint research with Jessica Sidman, Louis Theran, and Cynthia Vinzant.

Goal– Try to compute two polynomials:

1. The "pure condition" of the triangular prism.



2. The "circuit polynomial" of the triangular prism plus one edge.



Outline of the Talk

1. Where do these polynomials come from?
2. Summarize several failed, and some partially successful, computational strategies to write down these polynomials.
   - Gröbner bases,
   - Resultants,
   - Numerical AG, and
   - More Resultants.

Why discuss failed strategies?

- Failed attempts can still be educational.
- With so many experts in the audience, someone may have relevant experience!

Frameworks in the Plane

Points	$p_1, \ldots, p_n$.
Coordinates	$(x_1, y_1), \ldots, (x_n, y_n)$.
Squared Edge lengths:	$d_{ij} = \|\|p_i - p_j\|\|^2 = (x_i - x_j)^2 + (y_i - y_j)^2$.

Cayley-Menger determinant

For points $p_1, \ldots, p_k$, the CM-determinant, denoted $\{12 \cdots k\}$, computes a scaled volume of the $(k-1)$-simplex spanned.

If the points lie in the plane, any set of four points span a degenerate simplex $\Rightarrow$ every 4-CM-determinant

$$\{ijkl\} = f(d_{ij}, d_{ik}, \ldots, d_{kl}) = 0.$$

We will also see conditions like $\{ijk\} = 0$. This implies that p_i, p_j, p_k are collinear (so the simplex degenerates).

Rigidity of a Framework

- Every framework in the plane has a 3-dimensional space of rigid motions: translations in the plane (2 d.o.f.) and rotation (1 d.o.f.). While these change the point coordinates (x_i, y_i), they do not affect the squared distances (d_{ij}).
- If these are the **only** motions, the framework is said to be *rigid*. If removing any edge from a rigid graph would introduce a motion, the graph is said to be *minimally rigid*.
- A graph is said to be *generically rigid* if an assignment of generic coordinates to the points yields a rigid graph.

Theorem (Pollaczek-Geiringer 1927, Laman 1970)

A graph $G = (V, E)$ is generically minimally rigid if $|E| = 2|V| - 3$ and for every subgraph with k vertices has at most $2k - 3$ edges.

Algebraic Interpretation

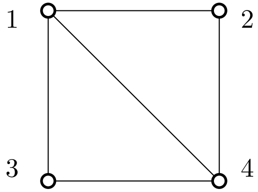
- We can interpret these as morphisms of varieties with corresponding ring homomorphsims.

$$\begin{array}{ccc} \mathbb{R}^{2n}/(\text{rigid motions}) & \longrightarrow & \mathbb{R}^{|E|} \\ (p_1, \ldots, p_n) & \mapsto & ||p_i - p_j||, (ij) \in E \end{array}$$

$$\begin{array}{ccc} \mathbb{R}[x, y]/\langle x_1 = y_1 = x_2 = 0\rangle & \longleftarrow & \mathbb{R}[d_{ij} : (ij) \in E]/\langle CM_4\rangle \\ (x_i - x_j)^2 + (y_i - y_j)^2 & \leftarrow\!\shortmid & d_{ij} \end{array}$$

- In this setting, *rigidity* means that the fibers are finite.
- The framework is non-rigid when the Jacobian of this map (also known as the rigidity matrix) drops rank.

Example: K_4 minus an edge



$$\begin{array}{c} \\ 12 \\ 13 \\ 14 \\ 24 \\ 34 \end{array}\begin{array}{c} \begin{array}{cccccccc} x_1 & x_2 & x_3 & x_4 & y_1 & y_2 & y_3 & y_4 \end{array} \\ \begin{pmatrix} x_1-x_2 & x_2-x_1 & & & y_1-y_2 & y_2-y_1 & & \\ x_1-x_3 & & x_3-x_1 & & y_1-y_3 & & y_3-y_1 & \\ x_1-x_4 & & & x_4-x_1 & y_1-y_4 & & & y_4-y_1 \\ & x_2-x_4 & & x_4-x_2 & & y_2-y_4 & & y_4-y_2 \\ & & x_3-x_4 & x_4-x_3 & y_3-y_4 & y_4-y_3 & & \end{pmatrix} \end{array}$$

Result: $x_4^2 y_2 y_3 - x_3 x_4 y_2 y_4 = x_4 y_2 (x_4 y_3 - x_3 y_4) = [124][134]$.

The Bracket Algebra

The bracket $[ijk]$ is a polynomial which indicates when the points p_i, p_j, p_k are collinear.
The formula is straightforward:

$$[ijk] = \begin{vmatrix} 1 & 1 & 1 \\ x_i & x_j & x_k \\ y_i & y_j & y_k \end{vmatrix}$$

These brackets are not independent, for example:

$$[135][245] = [145][235] + [125][345],$$

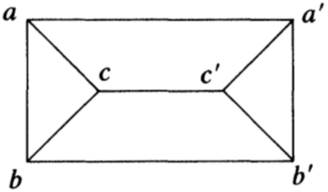
$$[135][246] = [124][356] + [146][235] + [126][345]$$

The brackets can be combined to give more complicated geometric information.

Geometric Interpretations

White & Whiteley "Algebraic Geometry of Stresses in Frameworks," 1983, combinatorial formulae along with geometric interpretations:

1.3. Triangular prism



Triangles abc, $a'b'c'$ are perspective from a line ≡ Triangle abc or $a'b'c'$ is collinear or the two triangles are perspective from a point

$$[abc][a'b'c']([abb'][a'c'c]-[a'bb'][ac'c])$$

Translating Brackets into CM-determinants

Essential connection between brackets and CM-determinants:

$$4[ijk]^2 = \{ijk\}$$

For the prism example, the pure condition in vertex coordinates was

$$[123][456]([125][346] - [146][235]).$$

- the ideal generated by [123] has preimage in $\mathbb{R}[d_{ij} : (ij) \in G]/CM_4$ generated by $\{123\}$.
- the ideal generated by [456] has preimage generated by $\{456\}$.
- the ideal generated by $[125][346] - [146][235]$ includes

$$16([125][346] - [146][235])([125][346] + [146][235])$$

$$= (4[125]^2)(4[346]^2) - (4[146]^2)(4[235]^2) \sim \{125\}\{346\} - \{146\}\{235\}.$$

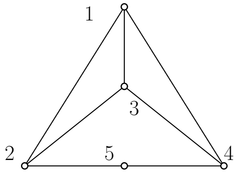
Pure Condition on Specified Support

The problem with $\{125\}\{346\} - \{146\}\{235\}$ as an edge-coordinate version of the pure condition is that its support includes edges not in the prism.

We want to find the unique up-to-scalar polynomial $\mathfrak{pc}_{prism}$ in the ideal $\langle\{125\}\{346\} - \{146\}\{235\}\rangle + \langle CM_4\rangle$ whose support is the prism.

Circuit Polynomial

The vanishing of the pure condition implies that the Jacobian drops rank. This does NOT mean that there is a true motion of the underlying framework.



Definition

A *circuit polynomial* is the unique up-to-scalar relation of minimal support in a prime ideal.

For us, given a circuit graph (a 1-overconstrained graph, minimal in this respect), there is a circuit polynomial in the Cayley-Menger ideal on that support.

The Motion Ideal

Let G be a Laman graph. Suppose that $G \cup \{e\}$ is a circuit graph with associated circuit polynomial $p_{G \cup e}$.

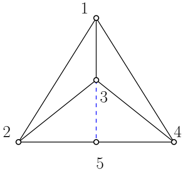
Definition

motionIdeal(G, e) is the ideal given by the coefficients of powers of d_e in the polynomial $p_{G \cup e}$.

If there exists a motion of the graph with d_e unconstrained, then all of the coefficients of $p_{G \cup e}$ with respect to d_e must be identically zero.

Example: Graph on 5 vertices

The graph G pictured is minimally rigid, but with the addition of (35), it becomes a circuit.



The circut polynomial $p_{G\cup(35)}$ has degree four in each variable, so motionIdeal(G, (35)) is generated by five polynomials.

motionIdeal(G, (35)) has six minimal primes. Only two correspond to true motions:

$$\langle d_{13}, d_{14} - d_{34}, d_{12} - d_{23}\rangle, \langle d_{23} - d_{34}, d_{12} - d_{14}, d_{25} - d_{45}\rangle$$

Want: circuit polynomial associated to prism plus edge, to understand configurations of prism with true motions.

Gröbner Bases

Most obvious approach.

- Start with $R = \mathbb{C}[x_i, y_i, d_{ij}]$.
- Let I be the ideal of R generated by $d_{ij} - (x_i - x_j)^2 - (y_i - y_j)^2$ together with $p(x_i, y_i)$, the pure condition associated to G.
- Compute a Gröbner basis under an elimination ordering prioritizing the x_i, y_i coordinates and d_{ij} for $(ij) \notin G$.
- Eventually obtain a polynomial in d_{ij} for $(ij) \in G$ corresponding to the pure condition.

Unfortunately, this is hopeless. Complexity of Buchberger's algorithm for constructing a Gröbner basis is $O(2^{2^n})$. Experimentally, these examples do **not** terminate.

Sylvester Resultant

A more direct approach to eliminating variables.

- Let $I = \langle f, g \rangle \subseteq R[x]$, where $f(x) = a_m x^m + \cdots + a_1 x + a_0$, and $g(x) = b_n x^n + \cdots + b_1 x + b_0$.
- The intersection $I \cap R$ is generated by the following determinant:

$$\operatorname{res}_x(f,g) := \begin{vmatrix} a_m & a_{m-1} & \cdots & a_0 & 0 & \cdots & & 0 \\ 0 & a_m & a_{m-1} & \cdots & a_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & a_m & a_{m-1} & \cdots & a_1 & a_0 \\ b_n & b_{n-1} & \cdots & b_0 & 0 & \cdots & & 0 \\ 0 & b_n & b_{n-1} & \cdots & b_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & b_n & b_{n-1} & \cdots & b_1 & b_0 \end{vmatrix}$$

Sylvester Resultant for Pure Condition

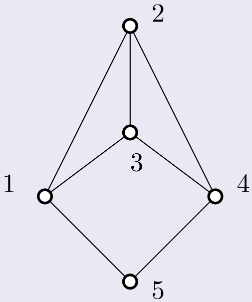
The resultant is implemented in `Macaulay2` with the command `resultant(f,g,x)`.

We can take a resultant of one form of the pure condition with an element of the Cayley-Menger ideal and eliminate a variable.

Example

The pure condition of the graph at right has a factor [145]. In edge coordinates, this maps to the CM determinant $\{145\}$.

Take $\text{res}_{14}(\{1234\}, \{145\})$ and obtain a degree-6 polynomial with 316 terms in the edges of the graph.

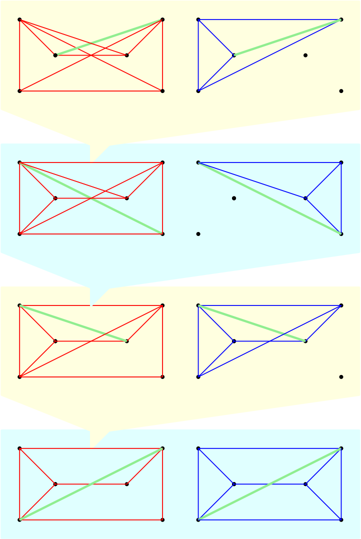


Computational Roadblock

Resultants could theoretically help us compute the pure condition of the triangular prism as well, as pictured at right.

Unfortunately, `Macaulay2` crashed on the first step of this series of resultants: $\text{res}_{24}(\{125\}\{346\} - \{136\}\{245\}), \{1234\})$.

[Malić-Streinu successfully implemented similar computations in Mathematica – more on this later!]



Numerical Algebraic Geometry

After roadblocks with symbolic computation, next plan: numerical algebraic geometry.

- Consider a system of polynomials in n variables defining an irreducible variety V of dimension k and degree d.
- Intersect the variety V with k general linear equations. The resulting system will be dimension 0 and degree d, i.e. d points.
- Use homotopy continuation to compute the solutions to the system, thus obtaining the number d.

In the case where the variety is not irreducible, monodromy allows us to check which solution points are on the same component.

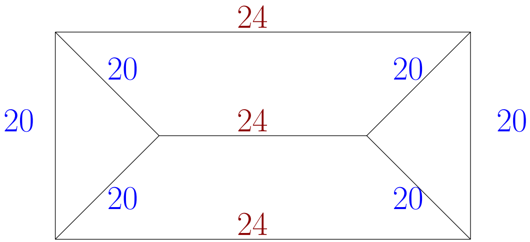
Computing degrees

- The pure condition we want should be a polynomial in the 9 edges of the triangular prism – the variety described is then a hypersurface of $\mathbb{C}^9$.
- If we take a general line in **that** $\mathbb{C}^9$, then it should intersect the variety in $(\deg \mathfrak{pc}_{prism})$ many points.
- Fix d_{ij} for some $(ij) \in G$. Choose generic values c_{kl} for all *other* variables $d_{kl} \in G$. The resulting system will have $\deg_{ij} \mathfrak{pc}$.

We use the `HomotopyContinuation.jl` package in Julia to carry out these computations. Many thanks to Paul Breiding for answering numerous questions in the process.

Degree of the Pure Condition

For our triangular prism, this method obtains $\deg \mathfrak{pc} = 40$ and the degree of each variables as labeled below:



The space of polynomials satisfying these degrees has dimension 362,557,602.

Motion Ideal by Numerical AG

To understand the motion ideal, the first step is to compute the circuit polynomial. This was not achievable with symbolic tools in `Macaulay2`. Instead, we apply Julia again.

We find that the circuit polynomial has:

- degree 20,
- with degree 8 in each of the variables of the triangular prism, and
- degree 12 in the extra edge variable.

This implies that the motion ideal is generated by 13 polynomials in 9 variables of degrees 8 through 20.

Coefficient Polynomials

In order to understand the motion ideal better, we would like to identify one or more of these coefficient polynomials.

Our general plan:

- Choose a coefficient polynomial of degree d.
- Compute the number $N(d)$ of possible monomials of degree d in 9 variables.
- Use Julia to generate $N(d) - 1$ points on the corresponding variety. Compute the $N(d)$ monomials corresponding to each of these points.
- Use linear algebra to compute a vector in the kernel of this matrix.

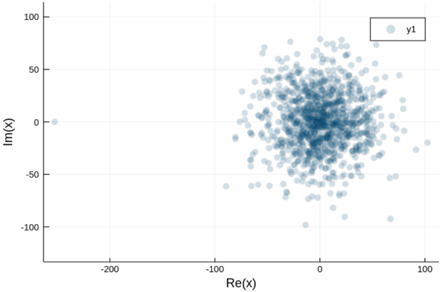
Given perfect data, the unique vector in the kernel would be the coefficients of the desired polynomial.

Which coefficient?

- The simplest coefficient polynomial to access would be the "constant" one: just intersect the variety with the hyperplane $d_e = 0$.
- However, the number of monomials of degree 20 in 9 variables with degree of each variable bounded by 8 is 2,429,487. In my implementation in Julia, the matrix computation never terminated.
- Instead, we will try to isolate the leading coefficient. Add in a new variable z satisfying $d_e z = 1$, then project away from d_e.
- The resulting surface is defined by a polynomial in z whose "constant term" is our old leading term.
- There are only 12,870 candidate monomials for this polynomial.

Fitting Coefficients to the Samples

After computing 12,869 generic points in the variety, we obtained a vector in the kernel. Its coordinates are plotted below:



Problems?

1. Computing the kernel of a matrix is highly unstable.
2. Passing from points to monomials will amplify errors.

Return to Resultants

Posted Mar 15, 2021:

Combinatorial Resultants in the Algebraic Rigidity Matroid

Goran Malić
Computer Science Department, Smith College, Northampton MA, USA
http://goranmalic.com
goranm00@gmail.com, gmalic@smith.edu

Ileana Streinu[1]
Computer Science Department, Smith College, Northampton, MA, USA
http://cs.smith.edu/-istreinu
istreinu@smith.edu, streinu@cs.umass.edu

Important take-aways for us:

1. Construction of circuit polynomials using iterated resultants.
2. Wolfram Mathematica is *much* faster at resultants than `Macaulay2`.

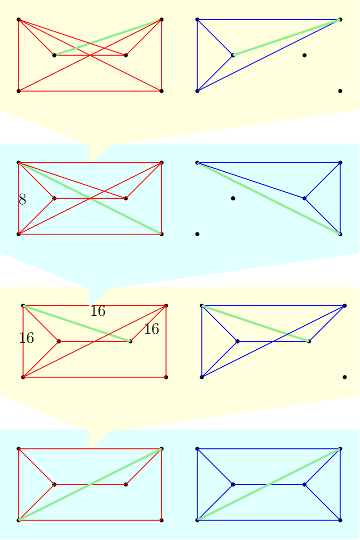
Resultants for Pure Condition Computation

- irreducible
- hom degree 4
- all variables deg 2
- 70 terms

- irreducible
- hom degree 10
- most variables deg 4
- 12,449 terms

- irreducible
- hom degree 24
- most variables deg 8
- 18,313,612 terms

- ???
- ???
- ???



Polynomial Coefficients

Degree	Degree in Each Variable	Number of Terms	Irreducible?
20	$\{8,4,8,6,4,8,4,6,8\}$	19,081	NO
19	$\{8,5,8,7,5,8,5,7,8\}$	65,184	NO
18	$\{8,6,8,8,6,8,6,8,8\}$	113,634	YES
17	$\{8,7,8,8,7,8,7,8,8\}$	130,377	YES
16	$\{8,8,8,8,8,8,8,8,8\}$	119,814	YES
15	$\{7,8,8,8,7,8,8,8,7\}$	92,763	YES
14	$\{6,8,8,8,6,8,8,8,6\}$	60,937	YES
13	$\{5,8,7,7,5,7,8,7,5\}$	33,772	YES
12	$\{4,8,6,6,4,6,8,6,4\}$	15,450	YES
11	$\{3,7,5,5,3,5,7,5,3\}$	5560	YES
10	$\{2,6,4,4,2,4,6,4,2\}$	1380	YES
9	$\{1,5,3,3,1,3,5,3,1\}$	211	NO
8	$\{0,4,2,2,0,2,4,2,0\}$	12	NO

Computation of Minimal Primes

- Macaulay2 ran for several hours before crashing.
- Singular ran for about a day before crashing.
- ~~The DynamicPolynomials package used by Julia does not accept polynomials with more than 100,000 terms.~~ (Based on a misreading of an error message)
- Bertini did not complete a single path-tracking due to the large number of terms in the polynomial.

Suggestions welcome!

Thank you for your attention!