Automated Geometric Theorem Proving

By Christopher Rackauckas

December 14, 2012

Abstract

Automated geometric theorem proving is the process of proving geometric theorems using algorithmic means. In this report we discuss a Groebner Basis method for automated geometric theorem proving and develop an algorithm for determining whether a geometric theorem follows from given hypotheses. After developing the algorithm, implementation and computing issues for *Mathematica* are addressed to give a more optimized algorithm and to prove that *Mathematica*'s Groebner Basis method can handle the necessary computation for any set of hypotheses polynomials $h_1, \ldots, h_s \in \mathbb{R}[x_1, \ldots x_n, u_1, \ldots, u_m]$.

Contents

Chapter 1

The Process

Automated geometric theorem proving is the process of proving geometric theorems using algorithmic means. As one could imagine, such a process has many applications in areas such as artificial intelligence. Even if one is interested in pure mathematics, having an algorithmic method for proving geometric theorems could serve as a simple way to prove many theorems that are difficult to prove using the conventional methods. For the purposes of this paper, we will be investigating how to prove geometric theorems using a process that involves polynomial equations and Groebner Bases. An alternative method of using polynomial equations is known as Wu's method and falls out the scope of this paper [5].

The idea is that we will code a problem using (x, y) coordinates. Coordinates whose values are arbitrarily chosen will be coded with some u_i value. Values which are determined by the previously declared variables will be coded as some x_i. Given coordinates which describe the figure, we then must constrain the relations between these coordinates with equations that must be satisfied in the geometric construction. Once we have our coordinates constrained, our goal is to test whether our theorem follows for all of the coordinate combinations that are allowed by the constraints.

To give a more concrete explanation of the process, we will develop the details of the steps by first showing how one would go about performing the steps on a simple problem and then generalize the process. The example that we will be looking at will be the parallelogram $ABCD$ shown in Figure 1.0.1. What we will prove using our automated methods is that the diagonals $\overline{AD}$ and $\overline{BC}$ bisect each other.

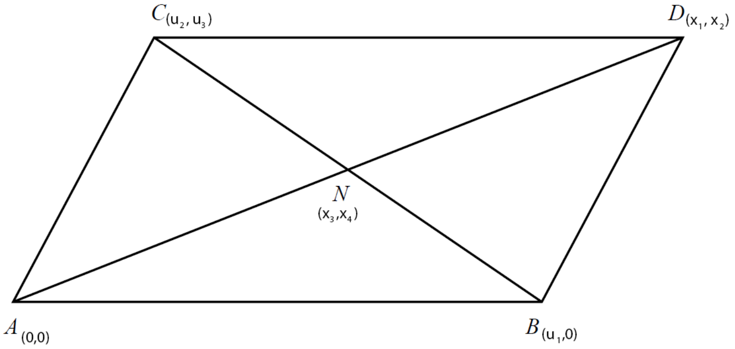


Figure 1.0.1: *Parallelogram ABCD*

1.1 Coding A Figure

To code the parallelogram $ABCD$, start by placing the origin of the coordinate grid at the point A. This implies $A = (0, 0)$. Now rotate the figure such that B is on the x-axis. Signify that B is some arbitrary distance u_1 away from A by declaring $B = (u_1, 0)$. Now denote C as some arbitrary point $C = (u_2, u_3)$. Since we have identified 3 points on our parallelogram, there exists only one place that the last corner could occupy. Therefore the point D is completely determined, and thus we define $D = (x_1, x_2)$ using x's to distinguish the fact that these values are determined. Notice the intersection of the diagonals is determined once we have picked the corners, and thus we define $N = (x_3, x_4)$.

To generalize this step, notice that we want to start by placing the origin at one of the points in our figure as this will decrease the number of non-zero coordinates and hence decrease the number of variables in the problem. Then we determine whether the values a given point are arbitrary or determined given the set of points we have already set. Note that, although it is not seen in our example, it is possible for the x coordinate to be determined while the y coordinate is not.

1.2 Coding the Hypotheses and Theorems

Now we must code the hypotheses that constrain the coordinates in our problem. For example, one hypothesis that we must consider is that, since $ABCD$ is a parallelogram, $\overline{AC} \parallel \overline{BD}$. We can describe this relation using the coordinates by noting that this is equivalent to saying $\overline{AC}$ has the same slope as $\overline{BD}$. This leads us to the equation $\frac{u_3}{u_2} = \frac{x_2}{x_1 - u_1}$. By clearing denominators and moving the variables to one side we obtain the polynomial equation $h_1 := u_3(x_1 - u_1) - x_2 u_2 = 0$. Doing the same for $\overline{AB} \parallel \overline{CD}$ we obtain the equation $h_2 := x_2 - u_3 = 0$. Another relation that we must consider is that N must be on the diagonal $\overline{AD}$ which implies that A, N, and D must be collinear. We can describe this relationship by the equivalent property that the slope of $\overline{AN}$ must be equal to the slope of $\overline{AD}$. After clearing denominators and moving the variables to one side, we obtain the equation $h_3 := x_4 x_1 - x_3 x_2 = 0$. Since N must be on the diagonal $\overline{BC}$, we use the same steps to obtain the equation $h_4 := x_4(u_2 - u_1) + u_3(x_3 - u_1) = 0$. Notice that these four equations completely define the relations between the points in our parallelogram.

Now we must code our theorems in the same manner. We wish to prove that N bisects $\overline{AD}$ which is the same as saying that the length of AN is equal to the length of ND. Thus, using the Euclidean distance formula we obtain the polynomial $t_1 := x_4^2 + x_3^2 - (x_3 - x_1)^2 - (x_4 - x_2)^2 = x_1^2 - 2x_1 x_3 - 2x_4 x_2 + x_2^2 = 0$. Doing the same for the other diagonal we obtain the equation $t_2 := 2x_3 u_1 - 2x_3 u_2 - 2x_4 u_3 - u_1^2 u_2^2 + u_3^2 = 0$.

Thus we see that to code the hypotheses we must simply find a complete set of relations that define the given problem and translate those relations into their equivalent coordinate equation form. Then we must define our theorems using their equivalent coordinate form. Table 1 shows a set translations of useful relations into their coordinate forms.

Table 1: Let $A = (x_1, y_1)$, $B = (x_2, y_2)$, $C = (x_3, y_3)$, $D = (x_4, y_4)$, $E = (x_5, y_5)$, and $F = (x_6, y_6)$.

Property	Idea	Polynomial Code
$\overline{AB} \parallel \overline{CD}$	$\text{Slope}(\overline{AB}) = \text{Slope}(\overline{CD})$	$\frac{y_2-y_1}{x_2-x_1} = \frac{y_4-y_3}{x_4-x_3}$
$\overline{AB} \perp \overline{CD}$	$\text{Slope}(\overline{AB}) = -\text{Slope}(\overline{CD})^{-1}$	$\frac{y_2-y_1}{x_2-x_1} = \frac{x_3-x_4}{y_4-y_3}$
$\text{Colinear}(A, B, C)$	$\text{Slope}(\overline{AB}) = \text{Slope}(\overline{AC})$	$\frac{y_2-y_1}{x_2-x_1} = \frac{y_3-y_1}{x_3-x_1}$
$\text{Length}(\overline{AB}) = \text{Length}(\overline{CD})$	$d(A, B) = d(C, D)$	$\sqrt{(y_2 - y_1)^2 - (x_2 - x_1)^2} = \sqrt{(y_4 - y_3)^2 - (x_4 - x_3)^2}$
C on a circle center A radius AB	$d(A, B) = d(A, C)$	$\sqrt{(y_2 - y_1)^2 - (x_2 - x_1)^2} = \sqrt{(y_3 - y_1)^2 - (x_3 - x_1)^2}$
C is the midpoint of $\overline{AB}$	$d(A, B) = d(A, C)$	$\sqrt{(y_2 - y_1)^2 - (x_2 - x_1)^2} = \sqrt{(y_3 - y_1)^2 - (x_3 - x_1)^2}$
The dot product of $\overline{AB}$ and $\overline{CD}$	$\overrightarrow{AB} = (x_2 - x_1, y_2 - y_1)$	$(x_2 - x_1)(x_4 - x_3) + (y_2 - y_1)(y_4 - y_3)$
$\angle ABC$ equals $\angle DEF$	$\overline{AB} \cdot \overline{BC} = d(A, B) d(B, C) \cos(\theta)$	$\frac{(x_2-x_1)(x_3-x_2)+(y_2-y_1)(y_3-y_2)}{\sqrt{(y_2-y_1)^2-(x_2-x_1)^2}\sqrt{(y_3-y_2)^2-(x_3-x_2)^2}} = \frac{(x_5-x_4)(x_6-x_5)+(y_5-y_4)(y_6-y_5)}{\sqrt{(y_5-y_4)^2-(x_5-x_4)^2}\sqrt{(y_6-y_5)^2-(x_6-x_5)^2}}$
A, B, C, and D lie on a circle	$\text{arcLength}(ABD) = \text{arcLength}(ACD)$	$\angle ABD$ equals $\angle ACD$
$\overline{BD}$ bisection $\angle ABC$		$\angle ABD$ equals $\angle DBC$

1.3 Geometric Proving

Once we have coded our problem, we must develop a method to show whether the theorems follow from the hypotheses. At this point, we have coded our hypotheses and theorems as equations in $\mathbb{R}[x_1, \ldots, x_n, u_1, \ldots, u_m]$. Notice that given a set of parameter vector $\underline{a} \in \mathbb{R}^{n+m}$, the hypothesis corresponding to equation h_i is true for point in parameter space if and only if $h_i(\underline{a}) = 0$. Thus the set of parameter vectors which satisfy hypotheses $h_1, \ldots h_s$ is $\boldsymbol{V}(h_1, \ldots, h_s)$. To show that a theorem follows from the hypotheses, what we want to show is that in all of the cases where the $h_1, \ldots, h_s$ are true, t_j is true. Thus we want to show that $\forall \underline{a} \in \boldsymbol{V}(h_1, \ldots, h_s)$, $t_j(\underline{a}) = 0$, or more succinctly, $t_j \in \boldsymbol{I}(\boldsymbol{V}(h_1, \ldots, h_s))$.

1.3.1 Strict Following

The above reasoning leads directly to the following definition:

Definition. The conclusion t **follows strictly** from the hypotheses $h_1, \ldots, h_s$ if $t \in \boldsymbol{I}(\boldsymbol{V}(h_1, \ldots, h_s))$.

From this definition we arrive at a useful proposition for computing whether a conclusion follows strictly from a given set of hypotheses:

Proposition. $t \in \sqrt{< h_1, \ldots, h_s >} \implies$ *t follows strictly from* $h_1, \ldots, h_s$.

We can easily prove this proposition. Assume that $t \in \sqrt{< h_1, \ldots, h_s >}$. Therefore $t^r \in < h_1, \ldots, h_s >$ for some $r \in \mathbb{Z}$. Thus we can write $t^r = \sum_i A_i h_i$ where each $A_i \in \mathbb{R}[x_1, \ldots, x_n, u_1, \ldots, u_m]$. Now take any $\underline{a} \in \boldsymbol{V}(h_1, \ldots, h_s)$. Notice $t^r(\underline{a}) = 0$ by construction. Thus $t(\underline{a}) = 0$ and since t was arbitrary we can conclude that $t \in \boldsymbol{I}(\boldsymbol{V}(h_1, \ldots, h_s))$ and thus t follows strictly from $h_1, \ldots, h_s$.

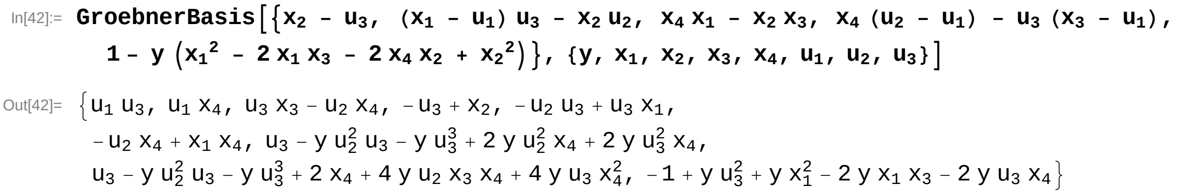
There exist common methods to solve the radical ideal membership problem. One way is to notice that

$$t \in \sqrt{< h_1, \ldots, h_s >} \iff 1 \in < h_1, \ldots, h_s, 1 - ty > \subset \mathbb{R}[y, x_1, \ldots, x_n, u_1, \ldots, u_m]$$

which is if and only if$\{1\}$ is the reduced Groebner Basis for $< h_1, \ldots, h_s, 1 - ty >$. Thus we arrive at our computational approach for proving whether a theorem follows strictly from $h_1, \ldots, h_s$:

Theorem 1. *t follows strictly from* $h_1, \ldots, h_s \iff \{1\}$ *is the reduced Groebner Basis for* $< h_1, \ldots, h_s, 1 - ty > \subset \mathbb{R}[y, x_1, \ldots, x_n, u_1, \ldots, u_m]$.

Thus we can return to our example to show how one could perform this calculation in practice. We wish to see whether theorems t_1 and t_2 follow strictly from h_1, h_2, h_3, and h_4. Thus we check to see if $\{1\}$ is the reduced Groebner Basis for $< h_1, h_2, h_3, h_4, 1 - t_1 y > \subset \mathbb{R}[y, x_1, x_2, x_3, x_4, u_1, u_2, u_3]$. We can do this using the GroebnerBasis command in *Mathematica*. By Figure 1.3.1 we see that $\{1\}$ is not the reduced Groebner Basis for $< h_1, h_2, h_3, h_4, 1 - t_1 y >$ and thus we must conclude from this method that there are cases where $\overline{AD}$ is not bisected by $\overline{BC}$. From our prior knowledge of geometry we know this cannot be the case, and thus we must investigate the problems of our method.



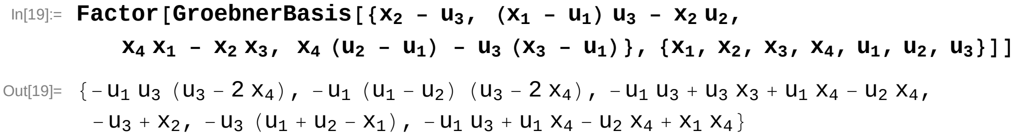
$$\text{In[42]:=}\ \textbf{GroebnerBasis}\Big[\Big\{x_2 - u_3,\ (x_1 - u_1)\, u_3 - x_2\, u_2,\ x_4\, x_1 - x_2\, x_3,\ x_4\,(u_2 - u_1) - u_3\,(x_3 - u_1),\ 1 - y\left(x_1{}^2 - 2\, x_1\, x_3 - 2\, x_4\, x_2 + x_2{}^2\right)\Big\},\ \{y, x_1, x_2, x_3, x_4, u_1, u_2, u_3\}\Big]$$

$$\text{Out[42]=}\ \Big\{u_1\, u_3,\ u_1\, x_4,\ u_3\, x_3 - u_2\, x_4,\ -u_3 + x_2,\ -u_2\, u_3 + u_3\, x_1,\ -u_2\, x_4 + x_1\, x_4,\ u_3 - y\, u_2^2\, u_3 - y\, u_3^3 + 2\, y\, u_2^2\, x_4 + 2\, y\, u_3^2\, x_4,\ u_3 - y\, u_2^2\, u_3 - y\, u_3^3 + 2\, x_4 + 4\, y\, u_2\, x_3\, x_4 + 4\, y\, u_3\, x_4^2,\ -1 + y\, u_3^2 + y\, x_1^2 - 2\, y\, x_1\, x_3 - 2\, y\, u_3\, x_4\Big\}$$

Figure 1.3.1: *Test for Strictly Following*

1.3.2 Generic Following

Let us look closer at our example to see if we can identify the error. More specifically, look at the factorization of the Groebner Basis for $< h_1, h_2, h_3, h_4 >$ shown in Figure 1.3.2. Notice that the equations are factorable. Since for polynomials $f_1, \ldots, f_s, f, g \in k[x_1, \ldots, x_n]$, $\boldsymbol{V}(f_1, \ldots, f_s, fg) = \boldsymbol{V}(f_1, \ldots, f_s, f) \cup \boldsymbol{V}(f_1, \ldots, f_s, g)$, we can factor our problem into separate varieties.



$$\text{In[19]:=}\ \textbf{Factor}[\textbf{GroebnerBasis}[\{x_2 - u_3,\ (x_1 - u_1)\, u_3 - x_2\, u_2,\ x_4\, x_1 - x_2\, x_3,\ x_4\,(u_2 - u_1) - u_3\,(x_3 - u_1)\},\ \{x_1, x_2, x_3, x_4, u_1, u_2, u_3\}]]$$

$$\text{Out[19]=}\ \{-u_1\, u_3\, (u_3 - 2\, x_4),\ -u_1\, (u_1 - u_2)\, (u_3 - 2\, x_4),\ -u_1\, u_3 + u_3\, x_3 + u_1\, x_4 - u_2\, x_4,\ -u_3 + x_2,\ -u_3\, (u_1 + u_2 - x_1),\ -u_1\, u_3 + u_1\, x_4 - u_2\, x_4 + x_1\, x_4\}$$

Figure 1.3.2: *Groebner Basis for* $< h_1, h_2, h_3, h_4 >$ *factorization*

If we repeatedly split by one polynomial and then take another Groebner Basis to factor again we see that we will decompose the variety $V = \boldsymbol{V}(h_1, h_2, h_3, h_4)$ as $V = V' \cup U_1 \cup U_2 \cup U_3$ where

$$V' = \boldsymbol{V}(x_1 - u_1 - u_2, x_2 - u_3, x_3 - \frac{u_1 + u_2}{2}, x_4 - \frac{u_3}{2})$$

$$U_1 = \boldsymbol{V}(x_2, x_4, u_3)$$

$$U_2 = \boldsymbol{V}(x_1, x_2, u_1 - u_2, u_3)$$

$$U_3 = \boldsymbol{V}(x_1 - u_2, x_2 - u_3, x_3u_3 - x_4u_2, u_1)$$

Notice that each of these varieties represents a different "case". V' is the case we wish to consider. U_1, U_2, and U_3 are cases that the algorithm found that we did not even consider. For example, U_1 and U_2 are the cases where $u_3 = 0$ which implies that $C = (u_2, 0)$ or that A, B, and C (and thus also D by the parallel hypothesis) must be collinear. Thus in these cases, N, the intersection of $\overline{AD}$ and $\overline{BC}$ is not uniquely defined! So our theorem falls apart in this case. Also, U_3 is the case where $u_1 = 0$ which implies $A = B$. Thus in this case we actually have a triangle and so the intersection of $\overline{AD}$ and $\overline{BC}$ is $A = B = (0, 0)$ which is at the endpoint of both lines and thus neither line is bisected.

Notice that these cases have the property that some u_i is determined. Here we simply have equations determining either u_1 or u_3 to be 0. We can generalize this with the following proposition:

Proposition. *Take $f \in \mathbb{R}[u_1, \ldots u_m]$ such that the highest power of u_m in f is $N > 0$. Consider $f = g$ where $g \in \mathbb{R}(u_1, \ldots, u_{m-1})[u_m]$. There exist at most N values of u_m such that $g = 0$.*

This is a corollary to the Fundamental Theorem of Algebra which states that g has N roots in $\mathbb{C}$. Thus g has at most N roots in $\mathbb{R}$. But this result states that for any function in just the $u_1, \ldots, u_m$, if you think of all but one of the variables as already chosen, there are a finite number of values that the last variable can be. Thus you can always think of such functions as determining the value of one of the u_i variables. But remember, the u_i variables are supposed to be arbitrary and thus not determined. Additionally, the problem we were having in the example was that the algorithm had found ways to determine some of the u_i variables to distort the problem into a case we were not expecting! Thus to avoid this problem, we would like to define a new way of saying a theorem follows from a given set of hypotheses. Our new definition must have include a condition that ensures none of the u_i variables are determined. This leads us to the following definition:

Definition. Let V be an irreducible variety in affine space $\mathbb{R}^{m+n}$ with coordinates $u_1, \ldots, u_m, x_1, \ldots, x_n$. The variables $u_1, \ldots, u_m$ are **algebraically independent on** V if no nonzero polynomial in the u_i alone vanishes identically on V.

Thus a variety is not algebraically independent on V if one of the defining equations for the variety is some polynomial in the u_i alone. Any variety where the u_i are not algebraically independent on V is a variety that has an equation that determines one of the u_i. Thus we wish to eliminate all cases where the u_i are not algebraically independent on V when considering our geometric theorem. Therefore we define our new definition of following:

Definition. The conclusion t **follows generically** from the hypotheses $h_1, \ldots, h_s$ if $g \in \boldsymbol{I}(V') \subset \mathbb{R}[u_1, \ldots, u_m, x_1, \ldots, x_n]$ where V' is the union of the components of $V = \boldsymbol{V}(h_1, \ldots, h_s)$ on which $u_1, \ldots, u_m$ are algebraically independent.

Notice that this definition is the same as the definition for follows strictly except we eliminate all of the cases where the u_i are not algebraically independent. The following proposition is necessary for developing the associated algorithm:

Proposition. *$\exists c(u_1, \ldots, u_m) \in \mathbb{R}[u_1, \ldots u_m]$ s.t. $ct \in \sqrt{< h_1, \ldots, h_s >}$, $h, \ldots, h_s \in \mathbb{R}[u_1, \ldots, u_m, x_1, \ldots, x_n] \implies t$ follows generically from $h_1, \ldots h_s$*

Assume there exists a c following the requirements above. Thus ct vanishes on $V = \boldsymbol{V}(h_1, \ldots, h_s)$ using the same proof as the earlier proposition for strict following. Thus if we look at any V' irreducible component algebraically independent component of V, $V' \subset V$ and thus $\boldsymbol{I}(\boldsymbol{V}(h_1, \ldots, h_s)) \subset \boldsymbol{I}(V')$. Thus $ct \in \boldsymbol{I}(V')$. But since V' was an irreducible variety, $\boldsymbol{I}(V')$ is a prime ideal and thus $c \in \boldsymbol{I}(V')$ or $t \in \boldsymbol{I}(V')$. But since V' is algebraically independent on the u_i and c is a function of $u_1, \ldots, u_m$, c does not vanish identically on V' and thus $c \notin \boldsymbol{I}(V')$. Thus $t \in \boldsymbol{I}(V')$ and thus t follows generically from $h_1, \ldots h_s$. From this we get the following corollary:

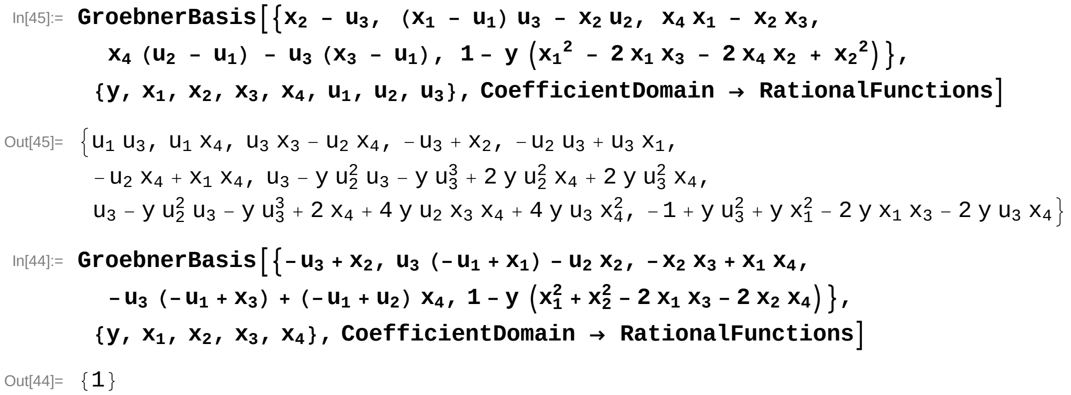
Corollary. *$t \in \sqrt{< h_1, \ldots, h_s >}$ where $h_1, \ldots, h_s \in \mathbb{R}(u_1, \ldots, u_m)[x_1, \ldots, x_n] \implies t$ follows generically from $h_1, \ldots h_s$.*

Notice this follows due to the fact that we can now think of the $c(u_1, \ldots, u_m)$ as some coefficient for t. Thus we see that we can solve the follows generically problem using the radical ideal membership algorithm.

Theorem 2. *t follows generically from $h_1, \ldots h_m \iff \{1\}$ is the reduced Groebner Basis of $< h_1, \ldots, h_s, 1 - ty > \subset \mathbb{R}(u_1, \ldots, u_m)[y, x_1, \ldots, x_n]$.*

Notice that this theorem simply implies that what we needed to do instead was treat the $u_1, \ldots, u_m$ variables as constants and solve for the Groebner Basis in just the $x_1, \ldots, x_n$ and y.

Figure 1.3.3 shows the results of this method using *Mathematica* on our parallelogram example. Notice that the first result shows that t_1, the theorem that $d(A, N) = d(N, D)$ does not follow strictly, but by declaring the u_i variables as constants (by not declaring them as variables in the GroebnerBasis command) and by telling *Mathematica* that the coefficient domain includes rational functions in the u_i variables we see that the reduced Groebner Basis is $\{1\}$ and thus t_1 follows generically from h_1, h_2, h_3, and h_4. We can do the same to see that t_2 follows generically from the h_1, h_2, h_3, and h_4. Thus our full theorem, the diagonals of a parallelogram bisect each other, follows generically from the hypotheses. We will accept t to be true if t follows generically from $h_1, \ldots, h_s$. Thus we have proven that the diagonals of a parallelogram bisect each other.



```
In[45]:= GroebnerBasis[{x2 - u3, (x1 - u1) u3 - x2 u2, x4 x1 - x2 x3,
           x4 (u2 - u1) - u3 (x3 - u1), 1 - y (x1^2 - 2 x1 x3 - 2 x4 x2 + x2^2)},
          {y, x1, x2, x3, x4, u1, u2, u3}, CoefficientDomain → RationalFunctions]

Out[45]= {u1 u3, u1 x4, u3 x3 - u2 x4, -u3 + x2, -u2 u3 + u3 x1,
          -u2 x4 + x1 x4, u3 - y u2^2 u3 - y u3^3 + 2 y u2^2 x4 + 2 y u3^2 x4,
          u3 - y u2^2 u3 - y u3^3 + 2 x4 + 4 y u2 x3 x4 + 4 y u3 x4^2, -1 + y u3^2 + y x1^2 - 2 y x1 x3 - 2 y u3 x4}

In[44]:= GroebnerBasis[{-u3 + x2, u3 (-u1 + x1) - u2 x2, -x2 x3 + x1 x4,
           -u3 (-u1 + x3) + (-u1 + u2) x4, 1 - y (x1^2 + x2^2 - 2 x1 x3 - 2 x2 x4)},
          {y, x1, x2, x3, x4}, CoefficientDomain → RationalFunctions]

Out[44]= {1}
```

Figure 1.3.3: *Test for Generic Following*

Chapter 2

Computational Concerns

2.1 Computational Efficiency

Now that we have a general algorithm for automated geometric theorem proving, let's address some computational concerns that one may come across when using this technique. To illustrate these concerns, let's use our technique to solve a problem that is slightly more computationally taxing. Take $\triangle ABC$. Recall that the altitude from A is the line from A that is perpendicular to $\overline{BC}$. Let's try to prove that the three altitudes of the triangle meet at a single point called the orthocenter of the triangle. Figure 2.1.1 gives a drawing of how we will code the problem. Notice that in the drawing we have three distinct intersections of altitudes. This is done to illustrate how we will code the proof but you should note that, given what we wish to prove, if the lines were truly perpendicular in the picture then these points would not be distinct.

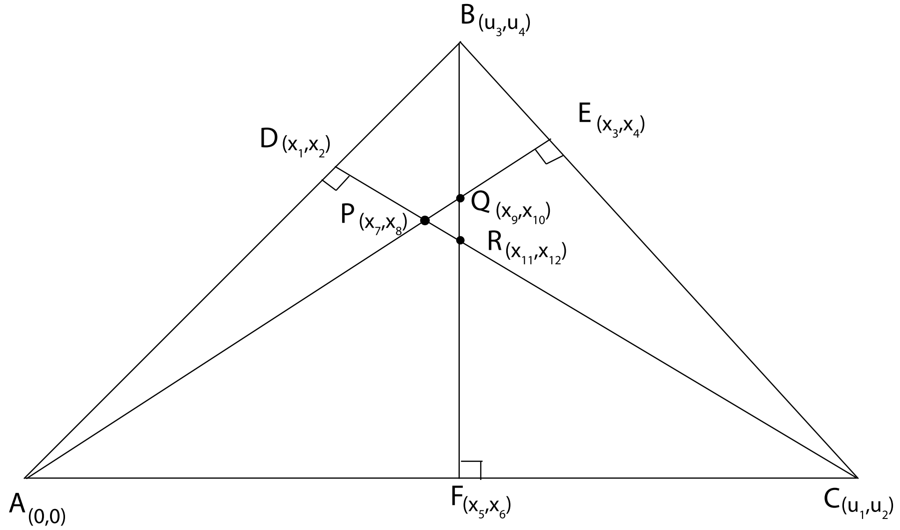


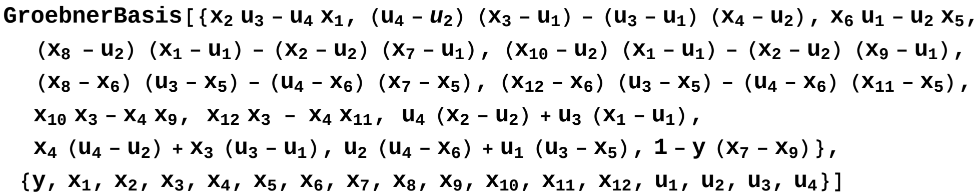
Figure 2.1.1: $\triangle ABC$

To set up this problem we require the following hypotheses: three hypotheses saying that the point of intersection between the altitude and the points defining the side of the triangle must be collinear, three hypotheses saying that the altitudes are perpendicular to the sides, and for each intersection point P, Q, and R we will have two collinear constraints to say that the intersection point is on both lines. The full set of hypotheses is found in Table 2. Our theorem is that $P = Q = R$. For example, one of the theorem equations is $t := x_7 - x_9$ saying that the x-coordinates of P and Q are the same.

Table 2: Hypotheses for the orthocenter problem.

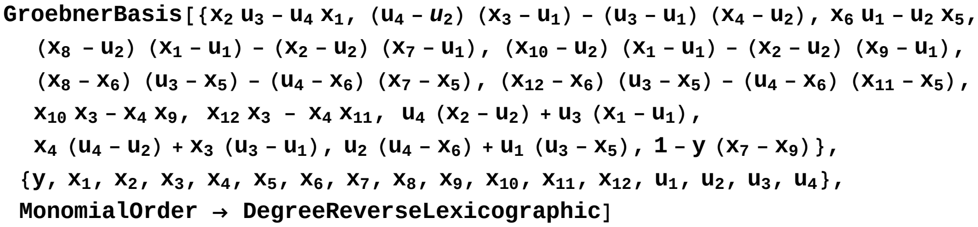
Hypothesis	Equation
$Colinear(A, D, B)$	$h_1 := x_2u_3 - u_4x_1$
$Colinear(B, E, C)$	$h_2 := (u_4 - u_2)(x_3 - u_1) - (u_3 - u_1)(x_4 - u_2)$
$Colinear(A, F, C)$	$h_3 := x_6u_1 - u_2x_5$
$Colinear(C, P, D)$	$h_4 := (x_8 - u_2)(x_1 - u_1) - (x_2 - u_2)(x_7 - u_1)$
$Colinear(C, Q, D)$	$h_5 := (x_{10} - u_2)(x_1 - u_1) - (x_2 - u_2)(x_9 - u_1)$
$Colinear(B, P, F)$	$h_6 := (x_8 - x_6)(u_3 - x_5) - (u_4 - x_6)(x_7 - x_5)$
$Colinear(B, R, F)$	$h_7 := (x_{12} - x_6)(u_3 - x_5) - (u_4 - x_6)(x_{11} - x_5)$
$Colinear(AQE)$	$h_8 := x_{10}x_3 - x_4x_9$
$Colinear(ARE)$	$h_9 := x_{12}x_3 - x_4x_{11}$
$\overline{DC} \perp \overline{AB}$	$h_{10} := u_4(x_2 - u_2) + u_3(x_1 - u_1)$
$\overline{AE} \perp \overline{BC}$	$h_{11} := x_4(u_4 - u_2) + x_3(u_3 - u_1)$
$\overline{BF} \perp \overline{AC}$	$h_{12} := u_2(u_4 - x_6) + u_1(u_3 - x_5)$

If we were to naively see if t strictly follows from the h_i, we may try using the *Mathematica* command from Figure 2.1.2.



```
GroebnerBasis[{x2 u3 - u4 x1, (u4 - u2) (x3 - u1) - (u3 - u1) (x4 - u2), x6 u1 - u2 x5,
  (x8 - u2) (x1 - u1) - (x2 - u2) (x7 - u1), (x10 - u2) (x1 - u1) - (x2 - u2) (x9 - u1),
  (x8 - x6) (u3 - x5) - (u4 - x6) (x7 - x5), (x12 - x6) (u3 - x5) - (u4 - x6) (x11 - x5),
  x10 x3 - x4 x9, x12 x3 - x4 x11, u4 (x2 - u2) + u3 (x1 - u1),
  x4 (u4 - u2) + x3 (u3 - u1), u2 (u4 - x6) + u1 (u3 - x5), 1 - y (x7 - x9)},
 {y, x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, x12, u1, u2, u3, u4}]
```

Figure 2.1.2: *Original Groebner Basis Command*



```
GroebnerBasis[{x2 u3 - u4 x1, (u4 - u2) (x3 - u1) - (u3 - u1) (x4 - u2), x6 u1 - u2 x5,
  (x8 - u2) (x1 - u1) - (x2 - u2) (x7 - u1), (x10 - u2) (x1 - u1) - (x2 - u2) (x9 - u1),
  (x8 - x6) (u3 - x5) - (u4 - x6) (x7 - x5), (x12 - x6) (u3 - x5) - (u4 - x6) (x11 - x5),
  x10 x3 - x4 x9, x12 x3 - x4 x11, u4 (x2 - u2) + u3 (x1 - u1),
  x4 (u4 - u2) + x3 (u3 - u1), u2 (u4 - x6) + u1 (u3 - x5), 1 - y (x7 - x9)},
 {y, x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, x12, u1, u2, u3, u4},
 MonomialOrder → DegreeReverseLexicographic]
```

Figure 2.1.3: *Groebner Basis Command with Grevlex*

However, after much waiting we will see that our computer will practically never give an answer to the problem. What went wrong? The large number of variables in this problem makes solving a Groebner Basis computationally hard. If one runs into this problem, one could try changing the monomial order to grevlex. Grevlex results in smaller Groebner bases and thus results in faster computations [2]. Therefore, if one instead tries the command from Figure 2.1.3, one will see that the same problem will be able to be solved (on a quad-core computer this takes $\approx$ 12 seconds).

Note that the reduced Groebner Basis for $< h_1, \ldots, h_s, 1 - ty > \subset \mathbb{R}(u_1, \ldots, u_m)[y, x_1, \ldots, x_n]$ has fewer variables than the reduced Groebner Basis for $< h_1, \ldots, h_s, 1 - ty > \subset \mathbb{R}[y, x_1, \ldots, x_n, u_1, \ldots, u_m]$. Thus the algorithm for seeing if t follows strictly from $h_1, \ldots, h_s$ is algorithmically harder than the algorithm for seeing if t follows generically from $h_1, \ldots, h_s$. However, the same principle applies for when one wants to see if t follows generically from $h_1, \ldots, h_s$: one may need to try setting the monomial order to grevlex for hard computations.

2.2 Utilizing Coefficients in $\mathbb{R}$

Setting the coefficient domain in *Mathematica* to RationalFunctions technically sets the field to all rational numbers and rational functions of all variables not in the variable list of the Groebner Basis command. Thus the command used to solve the example problem in *Mathematica* technically solves to see if $\{1\}$ is the reduced Groebner Basis of $< h_1, \ldots, h_s, 1-ty > \subset \mathbb{Q}(u_1, \ldots, u_m)[y, x_1, \ldots, x_n]$ and thus solves the problem when $h_1, \ldots, h_s \in \mathbb{Q}[x_1, \ldots x_n, u_1, \ldots, u_m]$. What we want to show is that this same command could be used to solve the problem for $h_1, \ldots, h_s \in \mathbb{R}[x_1, \ldots x_n, u_1, \ldots, u_m]$. First we will define the space we wish to prove our theorems in:

Definition. Take the ideal $I \neq \{0\}$ and let $I =< f_1, \ldots, f_s > \subset \mathbb{R}[x_1, \ldots, x_n]$. Denote the coefficients of each f_i by $c_{i_1}, \ldots, c_{i_{r_i}}$ for the r_i nonzero terms in f_i $i = 1, \ldots, s$. We define the Rational Coefficient Space of $f_1, \ldots, f_s$ to be $\mathbb{Q}(c_{1_1}, \ldots, c_{1_{r_1}}, \ldots, c_{s_1}, \ldots, c_{s_{r_s}})[x_1, \ldots, x_n]$, the set of polynomials in the $x_1, \ldots, x_n$ with coefficients in $\mathbb{Q}(c_{1_1}, \ldots, c_{1_{r_1}}, \ldots, c_{s_1}, \ldots, c_{s_{r_s}})$. We will abbreviate this space as $K(f_1, \ldots, f_s)$.

Notice that it is clear by the definition that for every generating polynomial f_i, $f_i \in K(f_1, \ldots, f_s)$. We make use of this definition to then prove our lemmas and our theorem.

Lemma. ***Colley's Lemma***. *Take the ideal $I \neq \{0\}$ and let $I =< f_1, \ldots, f_s > \subset \mathbb{R}[x_1, \ldots, x_n]$ where the coefficients of f_i are defined to be $c_{i_1}, \ldots, c_{i_{r_i}}$ where for the r_i nonzero terms in f_i $i = 1, \ldots, s$. Take $f, g \in I$ as nonzero polynomials and let $S(f, g)$ be the S-polynomial*

$$S(f,g) = \frac{x^\gamma}{LT(f)} \cdot f - \frac{x^\gamma}{LT(g)} \cdot g$$

where $x^\gamma = LCM(LM(f), LM(g))$. Then $S(f, g) \in K(f_1, \ldots, f_s)$.

Take $f, g \in I$. Look at the terms from $\frac{x^\gamma}{LT(f)} \cdot f$. Let $LC(f) = c$. Notice $c \in \mathbb{Q}(c_{1_1}, \ldots, c_{1_{r_1}}, \ldots, c_{s_1}, \ldots, c_{s_{r_s}})$ and thus we see that every term will be some coefficient from f divided by c. Thus the coefficient for every term of $\frac{x^\gamma}{LT(f)} f$ is in $\mathbb{Q}(c_{1_1}, \ldots, c_{1_{r_1}}, \ldots, c_{s_1}, \ldots, c_{s_{r_s}})$. The same analysis on $\frac{x^\gamma}{LT(g)} g$ gives the every term of the S polynomial has a coefficient in $\mathbb{Q}(c_{1_1}, \ldots, c_{1_{r_1}}, \ldots, c_{s_1}, \ldots, c_{s_{r_s}})$. Since the $multidegree(LCM(LM(f), LM(g)) \geq multidegree(LM(f))$ and the same for g, we see that $S(f, g) \in K(f_1, \ldots, f_s)$ which completes our proof.

Lemma. ***Navarrete's Lemma.*** *Take $I =< f_1, \ldots, f_s > \subset \mathbb{R}[x_1, \ldots, x_n]$. Fix a monomial order $>$ on $\mathbb{Z}^n_{\geq 0}$ and let $G = (g_1, \ldots, g_s)$ be an ordered s-tuple of polynomials in $K(f_1, \ldots, f_s)$. Take $g \in K(f_1, \ldots, f_s)$. The remainder of g on division by G, r, is such that $r \in K(f_1, \ldots, f_s)$.*

We will prove this by induction on the iteration i in the division algorithm. We want to show that at every iteration, the current polynomial, p, and the current remainder, r, are always in $K(f_1, \ldots, f_s)$. When $i = 0$, $r = 0$ and $p = g$ by the definition of the multivariable division algorithm and thus $g, r \in K(f_1, \ldots, f_s)$. Now assume that at iteration l, $g, r \in K(f_1, \ldots, f_s)$. In the next iteration, either a division occurs or a division does not occur. If a division occurs, r remains unchanged and thus r for the next iteration is in $K(f_1, \ldots, f_s)$. We note that p for the next iteration, p', is found by $p' = p - \frac{LT(p)}{LT(f_i)} \cdot f_i$ for some $i \in \{1, \ldots, s\}$. By the definition of the algorithm we know $LT(p)|LT(f_i)$. Thus $\frac{LT(p)}{LT(f_i)}$ is a monomial with multidegree $\geq$0. The coefficient of $\frac{LT(p)}{LT(f_i)}$ is in $\mathbb{Q}(c_{1_1}, \ldots, c_{1_{r_1}}, \ldots, c_{s_1}, \ldots, c_{s_{r_s}})$ and thus every term of $\frac{LT(p)}{LT(f_i)} \cdot f_i$ is in $K(f_1, \ldots, f_s)$. Thus $p' \in K(f_1, \ldots, f_s)$. If a division does not occur, then r for the next iteration, r' is found by $r' = r + LT(p)$. Since $p \in K(f_1, \ldots, f_s)$, $LT(p) \in K(f_1, \ldots, f_s)$ and thus $r' \in K(f_1, \ldots, f_s)$. Noting that in this case $p' = p - LT(p)$ we see that $p' \in K(f_1, \ldots, f_s)$. Thus at iteration $l + 1$ the current polynomial and the current remainder are in $K(f_1, \ldots, f_s)$. Thus at every iteration these two polynomials are in $K(f_1, \ldots, f_s)$ and thus they must be when the algorithm terminates. Thus the remainder of the division of g by F, the final r, is such that $r \in K(f_1, \ldots, f_s)$

This leads us to our theorem.

Theorem. ***The Replacement Theorem.*** *Take the ideal $I \neq \{0\}$ and let $I =< f_1, \ldots, f_s > \subset \mathbb{R}[x_1, \ldots, x_n]$. Let $G = \{g_1, \ldots, g_l\}$ be the reduced Groebner Basis for I. $G \subset K(f_1, \ldots, f_s)$.*

Take I. First I want to show there is a Groebner Basis G' for I s.t. $G' \subset K(f_1, \ldots f_s)$. Let's use Buchberger's Algorithm to generate the Groebner Basis G'. I want to prove by induction that at each iteration i in Buchberger's Algorithm, the current basis F is a subset of $K(f_1, \ldots, f_s)$. At the first iteration, $F = \{f_1, \ldots, f_s\}$ and by construction each $f_i \in K(f_1, \ldots f_s)$ and thus $F \subset K(f_1,\ldots,f_s)$. Now assume that at the start of iteration k the current basis F is a subset of $K(f_1,\ldots,f_s)$. Denote the basis for the iteration directly after iteration k as F'. Notice that every $g \in F'$ is either one of the original generating polynomials f_i or $\overline{S(f,g)}^{G'}$ where $f, g \in I$. By construction each $f_i \in K(f_1, \ldots f_s)$ for every generating polynomial f_i. By Colley's lemma we know that $S(f,g) \in K(f_1,\ldots,f_s)$ for any $f, g \in I$. By Navarrete's Lemma we know $\overline{S(f,g)}^{G'} \in K(f_1,\ldots,f_s)$. Thus every $g \in F'$ is such that $g \in K(f_1,\ldots,f_s)$. Thus $F' \subset K(f_1,\ldots,f_s)$. Therefore by induction we know that at the end of the last iteration we will receive a basis $F' \subset K(f_1,\ldots,f_s)$. Due to the properties of the Buchberger's Algorithm, we know that F' is a Groebner Basis for I. Thus we have found a Groebner Basis for I that is a subset of $K(f_1, \ldots f_s)$. Let's call this Groebner Basis G'.

Next, minimize the Groebner Basis G' to G'' using the fact that if $LT(g) \in < LT(G - \{g\}) >$, then $G - \{g\}$ is also a Groebner Basis for I. To do so, simply eliminate polynomials from the basis that satisfy this condition. Note that we must also divide each polynomial by its leading coefficient. Thus notice we have generated a minimal Groebner Basis G'' s.t $G'' \subset K(f_1,\ldots,f_s)$.

Now we can solve for the reduced Groebner Basis. Repeatedly replace $g \in G''$ with $g' = \overline{g}^{G-\{g\}}$ to construct the reduced Groebner Basis. Notice by Navarrete's Lemma that each $g' \in K(f_1,\ldots,f_s)$ and thus we have constructed a reduced Groebner Basis which consists of only g and g' type polynomials, each of which are in $K(f_1,\ldots,f_s)$. Thus we have found a reduced Groebner Basis for I which is a subset of $K(f_1,\ldots,f_s)$. Since reduced Groebner Bases are unique, the reduced Groebner Basis G for I is a subset of $K(f_1,\ldots,f_s)$. Thus our theorem is proven.

From this we easily see how to use the RationalFunction parameter to solve any Groebner Basis problem in $\mathbb{R}$ with the following corollary

Corollary. *Take the ideal $I \neq \{0\}$. Let $I =< f_1,\ldots,f_s >\subset \mathbb{R}[x_1,\ldots,x_n]$ and $I' =< f_1,\ldots,f_s >\subset K(f_1,\ldots f_S)$. The reduced Groebner Basis of I is equal to the reduced Groebner Basis of I'.*

To see this, simply look at the proof of the Replacement Theorem and realize that every step of our construction can be mimicked to make the reduced Groebner Basis of I' leading to the same reduced Groebner Basis as I. Thus we see that $\{1\}$ is the reduced Groebner Basis for the functions in $\mathbb{R}[x_1,\ldots,x_n]$ if and only if $\{1\}$ is the reduced Groebner Basis for the functions in $K(f_1,\ldots f_S)$. Thus we have an algorithm for solving the problem with any real coefficients using a rational function space. Simply define all of the coefficients of the generating functions to be some constant. Let *Mathematica* work in the field of rational numbers and these constants. The reduced Groebner Basis that we get from *Mathematica* will be reduced Groebner Basis will be in $K(f_1,\ldots,f_s)$ which will be identical to the reduced Groebner Basis when looking at the problem in $\mathbb{R}[x_1,\ldots,x_n]$. Thus the *Mathematica*-based method shown earlier for automated geometric theorem proving can be made to work for any set of polynomials that are a subset of $\mathbb{R}[x_1,\ldots,x_n]$ (notice that the same proof will work for $\mathbb{C}[x_1,\ldots,x_n]$).

Bibliography

[1] Cox, David A., John B. Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra.* 3rd ed. New York: Springer, 2007. Print.

[2] Lichtblau, Daniel. "Gröbner Bases in *Mathematica* 3.0." *The Mathematica Journal* (1996): 81-88. Wolfram Library Archive. Wolfram Research Inc. Web. 12 Dec. 2012. <http://140.177.205.65/infocenter/Articles/2179/TMJ_GroebnerBasis.pdf>.

[3] Miller, Conrad T. "Automated Theorem Proving in Plane Geometry." University of North Texas, 15 Nov. 2006. Web. 12 Dec. 2012. <http://people.unt.edu/ctm0055/Paper2.pdf>.

[4] Buchberger, Bruno, and Franz Winkler. "Groebner Bases Applied to Geometric Theorem Proving." *Groebner Bases and Applications.* By D. Wang. Cambridge: Cambridge UP, 2001. 281-92. Print.

[5] "GroebnerBasis." Wolfram Mathematica 9 Documentation. Wolfram Research Inc., n.d. Web. 11 Dec. 2012. <http://reference.wolfram.com/mathematica/ref/GroebnerBasis.html>.