Cylinders through five points in $\mathbb{R}^3$

Daniel Lichtblau
May 2004

Wolfram Research, Inc.
100 Trade Centre Dr.
Champaign, IL 61820
danl@wolfram.com

Abstract

Abstract. We address the following question: Given five points in $\mathbb{R}^3$, determine a right circular cylinder containing those points. We obtain algebraic equations for the axial line and radius parameters and show that these give six solutions in the generic case. An even number (0, 2, 4, or 6) will be real valued and hence correspond to actual cylinders in $\mathbb{R}^3$. Of these possibilities we will see that none is nongeneric; there are sets with nonempty interiors in the configuration space that give rise to each case. We will also investigate several computational and theoretical issues related to this problem. In particular we will show how Gröbner bases and equation solving techniques may be used to advantage in pursuit of this and related problems.

2000 Mathematics Subject Classification: 14N10, 14Q10, 51N20, 52A15, 65C05, 65H10, 68U20.

Key words and phrases: Computational geometry, enumerative geometry, Gröbner bases, nonlinear systems of equations.

1. Outline of the problem

Given five points in $\mathbb{R}^3$, we are to determine all right circular cylinders containing those points. We will do this by using the input data to set up and solve equations for the axial line and radius parameters. We will see that generically one obtains six solutions to these equations. Not surprisingly, an even number of the six must be real valued, as the complex valued ones appear in conjugate pairs. Moreover there are open regions in the real configuration space that give each of these possibilities and hence none are generically disallowed. In the sequel we frequently use the term "real cylinders" to denote real valued solutions to the cylinder equations that arise from a given configuration of five points. Sometimes we refer to arbitrary solutions as "cylinders" even if they have complex values. The meaning should be clear from context. As worded the problem appears to dwell in the realm of enumerative geometry, but note that it may be recast in a computational geometry setting: Given five points in $\mathbb{R}^3$, find the smallest r and orientation parameters such that the cylinder of radius 2 r with those parameters encloses tangentially the balls of radius r centered at the points. Several related topics we discuss will also have computational geometry intepretations.

We will discuss both approximate and exact computational methods for finding the (real or complex) cylinders. We show how certain specific configurations give rise to open sets in our domain for which we obtain the extreme cases, that is, either zero or six real cylinders. Here are a few of the related questions we will consider.

(1) Given the points and corresponding cylinder parameters, how might we display them graphically?

(2) Given the cylinder parameters, how may we obtain its implicit equation as a hypersurface in $\mathbb{R}^3$?

(3) Reversing this, how can one obtain parameters from the implicit form?

(4) Given six or more points, how do we find the coordinates of a (generically unique) cylinder in $\mathbb{R}^3$ that "best" fits those points?

(5) Given five points chosen with random uniform distribution in a cube, what is the expected probability that one lies inside the convex hull of the other four (this is related to the "no real cylinder" case).

Some of the material below was posted in response to a Usenet news group query on this topic [24]. Related techniques and results were similarly presented by Dave Rusin [29]. That one obtains six cylinders was previously demonstrated in [5], though by rather different means. Independent of this work, similar theory is developed in [13]. A related problem, finding cylinders of a given radius through four given points in $\mathbb{R}^3$, is discussed in [15] and [26]. A nice survey of computational commutative algebra methods that are applicable to nonlinear problems in computational geometry can be found in [7]. Another good general treatment of theoretical and practical aspects of using Gröbner bases in computational geometry is chapter 7 of [18]. Computations in the sequel were performed with the version of *Mathematica* [33] under development at the time of this writing (*Mathematica* (TM) is a registered trademark of Wolfram Research, Incorporated). Most were done on a 1.4 GHz Athelon machine under Linux.

The remainder of this paper is structured as follows. In section 2 we present several computational sides to the problem. These include finding and counting cylinder solutions. In section 3 we handle various associated computational geometry problems, and basics of point/cylinder visualization. Considerable attention is given to the ways in which these specialized methods intersect computational mathematics in general. Section 4 delves into the frequencies of real cylinders containing suitably random point sets. These investigations are again largely computational, though we relate some to a recent result in integral geometry. In section 5 we state and prove several theorems regarding the enumerative geometry of the problem. Following that are summary and acknowledgements.

2. Computing cylinders through five points

Finding cylinder parameters from a set of 5 points

We will assume unless otherwise stated that our points are generic. In particular, no three are collinear, no four are coplanar, cylinder axes do not lie in coordinate planes, and so forth. These assumptions allow us to avoid avoid computational complexity that would arise from parametrizing axial directions using a sphere (this gives rise to two problems: we have one extra variable, and so to eliminate it we would add an equation that normalizes the direction. Moreover we would double the size of our solution set because any direction is equivalent to its negative). Given these stipulations we proceed as follows.

With our assumptions in place, given a cylinder axis line L in $\mathbb{R}^3$ we may parametrize it as $\{ y= a \ x+ b, z= c\ x+ d\}$. Given r¿0 there is a unique circular cylinder C of radius r with center axis L. Supposing we have five points on that cylinder the following questions now arise. How do we find L and r? How do we use them to parametrize C e.g. for purposes of plotting it? Why do we even expect this will determine finitely many cylinders?

This last is answered as follows. Given a point on C we will project orthogonally onto L in order to get an equation involving the parameters we wish to find. We have five parameters to determine in the setup used above, and each point will give us an algebraic equation of the form $||\mathrm{perp}||^2=r^2$. For generic choice of points the equations should be algebraically independent, hence the dimension of the solution set would be zero. In more detail, if we take five points with indeterminate coordinates (that is, coordinates expressed as variables) then we obtain a system of five equations of the form $f_j[\ a,\ b,\ c,\ d,\ r]=0$ for which we want to solve for the cylinder parameters in terms of the coordinates. To show there are finitely many solutions it suffices by the implicit function theorem to show that the Jacobian of the map $\{ f_1, f_2, f_3, f_4, f_5\}$ has full rank for these generic coordinates. One can do this explicitly by finding the symbolic Jacobian, plugging in random values for the coordinates, and checking that the resulting matrix has full rank. We will instead show a computation in the last section that demonstrates there are generically at most nine solutions. Simple reasoning will further reduce this to eight. Following that is a more elaborate proof that the actual number is six.

Let us demonstrate how to solve for the cylinder parameters with a specific example. We will take as our parameter values a=3, b=2, c=4, d=-1,and $r=\sqrt{21}$.

The locus of points on C is obtained as sums of a vector on L plus a vector of length r perpendicular to L. All vectors perpendicular to L are spanned by any independent pair. We can obtain an orthonormal pair $\{\overrightarrow{w_1},\overrightarrow{w_2}\}$ in the standard way by finding the null space to the matrix whose one row is the vector along the axial direction, that is, $\overrightarrow{vec}=\{1,\ a,\ c\}$, and then using Gram-Schmidt to orthogonalize that pair.

We will then select five "random" points on C. We do this by selecting five values for an axial vector scale factor x and five values for an angle θ such that

$0 \le \theta \le 2\,\pi$. Our points will be of the form $\vec{v}+\vec{w}$ where

$$\vec{v} = x\overrightarrow{vec} + \overrightarrow{offset}$$

$$\vec{w} = r\mathrm{Cos}[\theta]\overrightarrow{w_1} + r\mathrm{Sin}[\theta]\overrightarrow{w_2}.$$

We will then show how to recover a set of cylinder parameters from these five points.

```
{a, b, c, d, r} = {3, 2, 4, -1, Sqrt[21]};
vec = {1, a, c}; offset = {0, b, d};
pair = NullSpace[{vec}];
{w1, w2} = GramSchmidt[pair]
```

$\{\{-\frac{4}{\sqrt{17}}, 0, \frac{1}{\sqrt{17}}\}, \{-\frac{3}{\sqrt{442}}, \sqrt{\frac{17}{26}}, -6\sqrt{\frac{2}{221}}\}\}$

```
SeedRandom[111111];
xvals = Table[Random[Real, {-10, 10}], {5}];
thetas = Table[Random[Real, {0, 2π}], {5}];
points =
Table[xvals[[j]]vec + offset + rCos[thetas[[j]]]w1+
rSin[thetas[[j]]]w2, {j, 5}];
```

Given a point on C we want to project orthogonally onto L, to get an equation involving the parameters we wish to find. This cannot be done immediately because L is not a subspace so we first translate our point by subtracting $\overrightarrow{offset}$. We then project onto the line spanned by $\overrightarrow{vec}$. Subtracting this projection from the translated vector gives us our $\overrightarrow{perp}$.

```
perp[vec1_, vec_, offset_]:=
vec1 - offset - Projection[vec1 - offset, vec]
perps = Table[perp[points[[j]], vec, offset], {j, 5}];
```

One can readily check this by taking the five points and seeing that all perps have length-squared equal to 21.

```
Map[#.#&, perps]
```

$\{21., 21., 21., 21., 21.\}$

When we use this point set to find r and the parameters describing L we will of course clear those that we set above so they become symbolic indeterminates. Using these points we will obtain the needed algebraic expressions for which we extract roots.

```
Clear[vec, offset, a, b, c, d, r]
vec = {1, a, c}; offset = {0, b, d};
perps = Map[perp[#, vec, offset]&, points];
exprs = Map[Numerator[Together[#.# - r^2]]&,
perps];
```

We can use numerical methods to find some roots. This is apparently very sensitive to initial conditions and often simply fails.

```
rt1 = FindRoot[Evaluate[Thread[exprs == 0]],
{a, 3.4}, {b, 2.8}, {c, 3.7}, {d, -1.6}, {r, 3.3},
MaxIterations → 500]
```

$\{a \to 3., b \to 2., c \to 4., d \to -1., r \to 4.58258\}$
rt2 = FindRoot[Evaluate[Thread[exprs == 0]],
{a, 2.7}, {b, 1.8}, {c, 3.2}, {d, –.7}, {r, 3},
MaxIterations → 500]
$\{a \to 1.90555, b \to 7.08893,$
$c \to 2.48119, d \to 7.02234, r \to -4.4421\}$

Here is another well known method to find numeric roots. We will sum the squares of the polynomials for which we just found zeroes and then minimize this sum. This too is sensitive to initial guesses.

esquares = Apply[Plus, Map[#^2&, exprs]];
{m3, rt3} = FindMinimum[Evaluate[esquares],
{a, 2.4}, {b, 1.8}, {c, 3.2}, {d, –.8}, {r, 3},
MaxIterations → 500,
Method → LevenbergMarquardt]
FindMinimum::fmlim :
The minimum could not be bracketed in 500 iterations.
$\{2.52309 \times 10^6, \{a \to 0.174607, b \to -0.257527,$
$c \to 0.217278, d \to -4.56064, r \to 6.93507\}\}$
{m4, rt4} = FindMinimum[Evaluate[esquares],
{a, 2.4}, {b, 1.8}, {c, 2.2}, {d, –1.8}, {r, 3},
MaxIterations → 500,
Method → LevenbergMarquardt]
$\{3.61685 \times 10^{-22}, \{a \to 1.90555, b \to 7.08893,$
$c \to 2.48119, d \to 7.02234, r \to 4.4421\}\}$

Clearly the above methods are not terribly robust. Next we will show how to find cylinder parameters using a global solver for nonlinear algebraic systems of equations.

Solving simultaneously for all roots of the cylinder parameter equations

An obvious drawback to the methods seen thus far is the need for good initial guesses. We may instead take advantage of the fact that the equations are all polynomial, and use a global solver suitable for such systems. We demonstrate below the utility of this approach. For simplicity we will only use integer coordinates. Specifically, we will take five points whose coordinates are all integers in the range {-10,10}, and show how to find our parameter values. To simplify matters we will solve for the square of the radius this time. This will avoid solutions with negative values for r as well as cut in half the number of complex valued solutions.

SeedRandom[1111];
intpoints = Table[Random[Integer, {–10, 10}],
{5}, {3}];
perps = Table[perp[intpoints[[j]], vec, offset],
{j, 5}];

exprs = Map[Numerator[Together[#.# − rsqr]]&,
perps];

Suprising though it may seem, it turns out to be computationally painless to get all solutions to this system, rather than just a particular solution. We do this with the *Mathematica* **NSolve** function, which uses a hybrid symbolic-numeric technique to efficiently find all roots. Explanations of this technology may be found in [9] [10] [25].

Timing[solns = NSolve[exprs, {*a*, *b*, *c*, *d*, rsqr}]]

$\{0.27\text{Second},$
$\{\{a \to -1.03253 + 0.760393i, b \to 6.11349 - 3.37419i,$
$c \to -0.322931 - 1.37768i, d \to -0.295427 + 6.8709i,$
$\text{rsqr} \to 344.25 + 23.8554i\},$
$\{a \to -1.03253 - 0.760393i, b \to 6.11349 + 3.37419i,$
$c \to -0.322931 + 1.37768i, d \to -0.295427 - 6.8709i,$
$\text{rsqr} \to 344.25 - 23.8554i\}, \{a \to 30.9362, b \to 93.172,$
$c \to 37.1186, d \to 92.7034, \text{rsqr} \to 198.258\},$
$\{a \to 0.151635, b \to -1.25748, c \to 1.58897,$
$d \to -6.45046, \text{rsqr} \to 83.0554\},$
$\{a \to 0.613253 - 0.359335i, b \to -4.49777 - 3.77132i,$
$c \to 0.102934 + 0.159852i, d \to -1.56979 + 2.23275i,$
$\text{rsqr} \to 57.5606 + 13.7534i\},$
$\{a \to 0.613253 + 0.359335i, b \to -4.49777 + 3.77132i,$
$c \to 0.102934 - 0.159852i, d \to -1.56979 - 2.23275i,$
$\text{rsqr} \to 57.5606 - 13.7534i\}\}\}$

Actually we can get exact solutions in the same way, albeit at greater computational cost. This illustrates a sort of cascading "hybrid" algorithm: one starts with asymbolic-numeric method to handle numeric problems, then modifies it to give exact rather than approximate results.

Timing[exactsolns =
NSolve[exprs, {*a*, *b*, *c*, *d*, rsqr},
WorkingPrecision → Infinity];]

$\{1.71\text{Second}, \text{Null}\}$

The solution set in exact form is a bit larger than merits printing:

LeafCount[exactsolns]

12139

We can improve considerably on the computational efficiency of finding cylinder parameters from five points. For one, a different formulation of the problem, to be utilized later, finds directions for which all points project onto the same circle in a plane perpendicular to the direction. Using this we can reduce the computational time by a substantial factor vs. the method shown above.

In addition to changing the formulation of the problem to one that is computationally easier, one might also change the solver method. We discuss one very efficient alternative. This is the sparse homotopy method described in [30]. Here one constructs a readily solved system using information from the Newton polytope. One then forms a homotopy to move from each solution of the first system to a solution of the new system. If we call the systems F(x) and G(x) respectively, where xdenotes a vector of variables, then one adds a new variable, t, and sets up the homotopy between solutions in each set as a relation (1-t)F(x)+t G(x)=0. At time t=0 we have a solution to the first system, and a time t=1 we have a solution to the new system. Techniques for moving along the homtopy path generally involve some sort of predictor-corrector method to increment t by a small amount and then alter the coordinates of x to maintain the relation above; a general introduction to this method is presented in [23]. For our cylinder problem there is a nice refinement that goes by the name of the "cheater's homotopy" wherein we start with known solutions for one set of points and hence can skip the first step of the general approach. In order to find cylinder parameters for each subsequent set of points we simply use a homotopy appropriate for the new set of equations.

An occasional disadvantage to the general sparse homotopy technique is that in some cases one has fewer actual solutions than are given by the starting system. When this occurs, in the process of following the homotopies some must wander off to infinity. This can pose difficulties for the software in terms of deciding when a path is diverging rather than merely wandering afar prior to converging. As we will explain later, the sparse homotopy method will predict that there are eight solutions for cylinder parameters, two more than are actually present. Hence the cheater's homotopy is all the more appealing for this class of problems. It should be noted, however, that the general sparse method is far better at approximating the correct number of solutions than any predecessor approach based on homotopies, and moreover it tends to handle systems with far more solutions than can be successfully tackled by methods such as [9] that require computation of matrix eigensystems. See [31] for further details.

The size of the solution set

In examples that use **NSolve** we get six possibly complex valued solutions. To find out what is happening we will continue with the example above. We now form a (lexicographic) Gröbner basis for *exprs*. This is a standard tactic for computational equation manipulation [1] [3] [6] [11] [17]. The idea is that it effectively triangulates the polynomial system in a manner that will become clear below. As we do not care about the ordering among the variables, we use the built in heuristic sorting algorithm (which was designed for the purpose of speed enhancement).

Timing[gb = GroebnerBasis[exprs, Sort → True];]
gb[[1]]

$\{0.88\text{Second}, \text{Null}\}$
$-36556110913245+$
$220390657872336c - 1091903566289090c^2+$
$412701818100667c^3 - 375539205226535c^4+$
$327695756151531c^5 - 8563282997415c^6$

The coefficients are large so we will only show that first polynomial. It is also instructive to see the structure of the full set of polynomials. The code fragment below serves for this purpose. We see that the first polynomial is of degree 6 in the variable c (as we already knew), and the rest are quintic in c and linear in each of the respective other variables. So now we see what was meant by triangularizing the system. To solve it one could find the six roots in c and back substitute each into the remaining equations in order to get six corresponding solutions in each of the remaining variables.

list1 = Map[Apply[List, #]&, gb]/._Integer * x_ → x;
list2 = Map[Cases[#, a_/;!IntegerQ[a]]&, list1]

$\{\{c, c^2, c^3, c^4, c^5, c^6\},$
$\{a, c, c^2, c^3, c^4, c^5\}, \{c, c^2, c^3, c^4, c^5, d\},$
$\{b, c, c^2, c^3, c^4, c^5\}, \{c, c^2, c^3, c^4, c^5, \mathrm{rsqr}\}\}$

This tells us to expect six solutions in general. While a theorem and proof will be deferred until later, two reasons to believe this are offered below.

Reason 1: In the theory of lexicographic Gröbner bases there is a fact known as the Shape Lemma [2], which may be stated as follows. As is well known, a generic zero dimensional polynomial ideal over an infinite field is radical and in general position with respect to the last variable in any ordering of the variables. In other words, the variety has no multiplicity and moreover its finitely many points do not share any coordinates. The lemma states that under these circumstances any lexicographic Gröbner basis will have exactly one polynomial with leading term a pure product in each variable, all but the one in the smallest variable will be linear, and that one in the smallest variable will have degree equal to the size of the solution set. One interpretation of the Shape Lemma is almost a matter of philosophy: one proves the above fact given a radical ideal in general position, and then asserts that generic ideals satisfy these hypotheses. In addition to the Shape Lemma there is the following result: lexicographic Gröbner bases of ideals defined over rational function fields remain Gröbner bases after generic specialization of coefficients [16] [21]. In other words, there is a Zariski-open set in the parameter space for which specializations do not alter the skeleton of the basis. We use these facts as follows: if our selection of coefficients was generic, we may conclude that the generic Gröbner basis has the same shape as that of the basis we just obtained. Moreover we may believe that our selection was generic because (i) it had the correct shape of a generic basis, and (ii) we chose the data at random from a fairly large set.

Reason 2: The simulation indicated below tested over 4000 randomly chosen configurations. Every one gave exactly six solutions, and this too strongly suggests that six is the generic value.

We will also use the results of these tests to say a bit about percentages of examples for which one obtains given numbers of real solutions.

One might ask why we do not simply compute a lexicographic basis for our system using generic coefficients. The answer is that it does not finish in finite time. Indeed, even making one coordinate a parameter leads to tremendous computational effort (nearly an hour on a *Pentium 2* 333Mh processor) and very large coefficients for the basis; polynomials in that parameter are of high degree with large integer coefficients.

Another thing to note is that Gröbner bases computations never leave their base field. In other words, if we begin with real data then the polynomials in the basis will have real coefficients. As we now know the form of the generic Gröbner basis for our systems, an immediate conclusion is that we will get complex solutions in pairs, and thus we might have zero, two, four, or six (real) cylinders in $\mathbb{R}^3$. In the case of the example above we have two.

3. Computational geometry of the solution cylinders

Finding the implicit equation of a cylinder from its parametric form

Given the parameters of a cylinder, it is natural to ask how one might obtain the implicit form. The first method we show, best described as "applied brute force", is from modern elimination theory. Some references for this technique are [1] [11] [18] [22]. We begin with equations for $\{x, y, z\}$ in terms of the five parameters and the sine and cosine of an (unrestricted) angular parameter. In more detail, we have a parametrization for the cylinder in terms of a scalar multiplier t for the direction vector $\overrightarrow{vec}$ and an angle θ to determine a unit vector in the plane orthogonal to $\overrightarrow{vec}$. To make this parametrization algebraic we can use the usual pair of trigonometric functions, abbreviated below as (algebraic variables) sin and cos. This gives one parameter more but of course we also now have the polynomial $\cos^2$+$\sin^2$-1. We use a standard Gröbner basis method for elimination of variables, which involves a term ordering that is typically efficient for partially triangularizing the polynomials. In particular it weights terms that involve any of the elimination variables higher than all other terms. We form a Gröbner basis with respect to such an ordering and remove all polynomials that contain any of the elimination variables $\{t, sin, cos\}$. What remains, a single polynomial, is the implicit relation in the variables $\{x, y, z\}$. The code snippet below uses some simplifications such as clearing of denominators.

```
Clear[a, b, c, d, t, r];
vec = {1, a, c}; offset = {0, b, d};
pair = NullSpace[{vec}];
{w1, w2} = GramSchmidt[pair];
polys =
Append[t vec + offset + r cos w1 + r sin w2 - {x, y, z},
sin^2 + cos^2 -1];
ee = Numerator[MapAll[Together, polys]];
ff = Numerator[Together[PowerExpand[ee]]];
Timing[
imp =
First[GroebnerBasis[ff, {x, y, z}, {t, sin, cos},
Sort -> True,
MonomialOrder -> EliminationOrder,
CoefficientDomain -> RationalFunctions]]]
```

$\{1.63\text{Second}, b^2 + b^2c^2 - 2abcd + d^2 + a^2d^2 - r^2 -$
$a^2r^2 - c^2r^2 + (2ab + 2cd)x + (a^2 + c^2)x^2 +$
$(-2b - 2bc^2 + 2acd)y - 2axy + (1 + c^2)y^2 +$
$(2abc - 2d - 2a^2d)z - 2cxz - 2acyz + (1 + a^2)z^2\}$

Note that, as one might expect, the implicit polynomial is a function of r^2.

We now check this implicit polynomial with the example we used above.

i3 = imp/.{a → 3, b → 2, c → 4, d → −1, r → Sqrt[21]}
i3/.Map[Thread[{x, y, z}-¿#]&, points]//
Chop

$-420 + 4x + 25x^2 - 92y - 6xy +$
$17y^2 + 68z - 8xz - 24yz + 10z^2$
$\{0, 0, 0, 0, 0\}$

Here is a more clever way to find the implicit form. Just use the formulation we gave to find the distance from a point to the axial line. This gives an equation satisfied by every point on the cylinder. Hence it will be the hypersurface expression we seek.

pp = perp[{x, y, z}, vec, offset];
imp2 = Numerator[Together[pp.pp − r^2]];
Together[imp − imp2]

0

As is so often the case, we see that brute force can be useful but it is no match for finesse. Why is the first approach of interest? Simply because it is a standard technique in computational algebraic geometry, and works when geometric intuition may not be so readily available.

Finding cylinder parameters from the implicit form

Now we look into the reverse problem of finding parameters from the implicit form. Algebraic parametrization of algebraic objects is in general difficult, when it can be done at all. That said, parametrization of quadric surfaces in $\mathbb{R}^3$ can be done and in fact is not terribly hard; general methods for this are presented in [18, chapter 5]. For the case of cylinders we will show a very simple approach. Note that we are using a trigonometric parameter to describe circle coordinates and thus it is not algebraic (though it suits our purposes quite well). If so desired, one can convert to the usual algebraic parametrization after a rotation of coordinates. We leave the details to the interested reader. We will illustrate our method for cylinders using the example above. As we know the general implicit form, it suffices merely to equate coefficients with those of the specific implicit form and solve for the parameters. Some of the coefficients are linear in the cylinder parameters so this is computationally quite easy.

generalImplicitForm = imp/.r^2 → rsqr;
thiscase = generalImplicitForm/.
{a → 3, b → 2, c → 4, d → −1, rsqr → 21};
SolveAlways[thiscase == generalImplicitForm,
{x, y, z}]

$\{\{\text{rsqr} \to 21, b \to 2, d \to -1, a \to 3, c \to 4\}\}$

Were the coefficient equations not so readily solvable we could instead do as follows. Starting with that cylinder in implicit form we generate at least five points that lie on it. To this end we might simply take values for x and ycoordinates, and solve for z. We then form equations for the parameters from the first five points and solve them. This gives candidate parameter values. Last we find the implicit equation corresponding to each set of parameters: the correct parameters will be the ones that recover the original implicit form (up to scalar multiple).

```
points =
Partition[
Flatten[
{x, y, z}/.
Table[Solve[{thiscase == 0, x == j, y == 0},
{x, y, z}], {j, 0, 2}]], 3];
perps = Table[perp[points[[k]], vec, offset], {k, 5}];
exprs = (Numerator[Together[#1.#1 - rsqr]]&)/@
perps;
Select[candidates,
NumericQ[Together[generalImplicitForm/.#1 / thiscase]]&]
{{rsqr → 21, b → 2, d → -1, c → 4, a → 3}}
```

Solving for overdetermined cylinders

A natural question to ask, regarding computations related to this problem, is what we might do to find a cylinder (approximately) containing more than five given points? The typical case is where the points all lie approximately on a cylinder and we wish to find the best fitting one (perhaps to assess tolerance). We will use **FindMinimum** for this task. We can set up an expression to minimize as follows. First form the list *perps* of orthogonal complements to projections of our points onto the axial line. Then take a sum of squares of differences between projected lengths and radius. To illustrate we reconstruct our original example, but this time we will use more points chosen at random, and we will add random noise to these. We know approximately the correct values for the parameters, and at this value our sum of squares will be near zero. An important issue is how to find good starting values. We do this by taking five points, solving for cylinder parameters, and then using other points to decide which of the six possibilities we should utilize.

```
{a, b, c, d, rspr} = {3, 2, 4, -1, 21}; numpts = 8;
vec = {1, a, c}; offset = {0, b, d};
pair = NullSpace[{vec}];
{w1, w2} = GramSchmidt[pair];
xvals = Table[Random[Real, {-10, 10}], {numpts}];
thetas = Table[Random[Real, {0, 2π}], {numpts}];
randomNoise3D[max_]:= max Table[Random[], {3}]
points =
Table[xvals[[j]]vec + offset+
```

```
√rsqr Cos[thetas[[j]]]w1+
√rsqr Sin[thetas[[j]]]w2+
randomNoise3D[0.001], {j, numpts}];
```

First we will get a set of candidate starting values.

```
Clear[a, b, c, d, rsqr]
perps = Table[perp[points[[j]], vec, offset],
{j, numpts}];
exprs =
Map[
Numerator[
Together[Rationalize[#.#, 0] − rsqr]]&,
Take[perps, 5]];
Timing[solns = NSolve[exprs, {a, b, c, d, rsqr}]]
```

$\{0.23\text{Second},$
$\{\{a \to -0.126422 - 0.717553i, b \to 15.0555 + 8.16833i,$
$c \to -0.0773551 + 0.727294i,$
$d \to 24.6363 - 0.307637i, \text{rsqr} \to 96.45 + 59.8042i\},$
$\{a \to -0.126422 + 0.717553i, b \to 15.0555 - 8.16833i,$
$c \to -0.0773551 - 0.727294i,$
$d \to 24.6363 + 0.307637i, \text{rsqr} \to 96.45 - 59.8042i\},$
$\{a \to 2.1017, b \to 9.66139, c \to 2.35744,$
$d \to 9.4369, \text{rsqr} \to 35.0918\},$
$\{a \to 3.52001 - 2.7712i, b \to 5.33062 + 23.8131i,$
$c \to 7.31272 - 2.08636i, d \to -12.7639 + 30.8396i,$
$\text{rsqr} \to 7.67878 + 13.8389i\},$
$\{a \to 3.52001 + 2.7712i, b \to 5.33062 - 23.8131i,$
$c \to 7.31272 + 2.08636i, d \to -12.7639 - 30.8396i,$
$\text{rsqr} \to 7.67878 - 13.8389i\},$
$\{a \to 3.00057, b \to 1.99012, c \to 3.99872,$
$d \to -1.01676, \text{rsqr} \to 21.0378\}\}\}$

Next we select the best candidate by calculating values of the six implicit equations at all points, summing absolute values for each equation over all points, and using the parameters that correspond to the implicit equation that yields the smallest such sum.

```
squaresums =
Apply[Plus, Abs[(generalImplicitForm/.solns)/.
Map[Thread[{x, y, z} → #]&, points]]]
```

$\{195.805, 195.805, 240.623, 10683., 10683., 2.20873\}$

Clearly the sixth solution is the one to use. This may be found programmatically as below.

```
candidate =
solns[[Position[squaresums, Min[squaresums]][[
1, 1]]]]
```

$\{a \to 3.00057, b \to 1.99012,$
$c \to 3.99872, d \to -1.01676, \text{rsqr} \to 21.0378\}$

Now we form our sum of squares and minimize it.

```
sumsquarelens =
Plus@@((√#1.#1 − r)^2&)/@perps;
startvals = (List@@#1&)/@candidate;
newstartvals = startvals/.{rsqr, v_} :→ {r, √v};
{min, ee} = FindMinimum[Evaluate[sumsquarelens],
Evaluate[Sequence@@newstartvals]]
```

$\{8.48222 \times 10^{-8}, \{a \to 2.9999, b \to 1.99926,$
$c \to 3.99985, d \to -1.00179, r \to 4.58278\}\}$

As a general remark, attempts with different optimization methods indicate that this sort of expression is quite problematic to minimize without reasonable starting points.

There is an interesting application that can make use of the this. In the industrial realm of geometric tolerancing one wishes to measure how well an object conforms to specifications. The cylinder is of course a very common object in manufacture. A good approach to metrology involving cylinders may be found in [14]. The technology discussed therein is especially effective when the object in question is small and may be readily positioned, but one might accept a cruder approach e.g. to check an underground pipeline. For this sort of task one could probe five points, obtain from them a set of approximate cylinder parameters, then probe several others and obtain parameters for a least-squares nearest cylinder as above. One can then simply check whether all probed points are within specification tolerance in actual radial measure from the computed axial line.

Visualization of the real cylinders containing a set of points

It can be a challenge to visualize how a given set of points fit on the cylinders that are computed to contain those points. Below we present a viable approach. The idea is to draw the cylinder, its axis, and the points connected by segments. We then decrease the radius of the drawn cylinder to 70% of its actual value as this frequently maintains visibility and also more readily shows how the points belong on the cylinder. Another idea would be to punch holes in the cylinder where points are obscured, so that they might be made visible, but this involves considerably more programming effort.

```
showlines[points_List, rest___]:=
Module[{plotpoints, plotlines, len = Length[points]},
plotpoints =
Table[Graphics3D[
{Blue, PointSize[0.03], Point[points[[j]]]}],
{j, 1, len}];
plotlines =
Table[Graphics3D[
{Hue[1/26 (j^2 + k − 4)],
Line[{points[[j]], points[[k]]}]}],
{k, 1, len − 1}, {j, k + 1, len}];
Show[plotpoints, plotlines, rest,
```

```
DisplayFunction → Identity, Boxed → False,
Axes → False, Shading → False,
ViewPoint → {-1/2, 2, 1},
ImageSize → {300, 480}]]
cylinderplot[rt_, pts_List, vec_, offset_,
showcyl_ : True]:=
Module[{r, vec2, lin, x, θ, w1, w2, circ,
cylplot, axis}, r = Sqrt[rsqr/.rt]; vec2 = vec/.rt;
lin = x vec + offset/.rt; pair = NullSpace[{vec2}];
{w1, w2} = GramSchmidt[pair];
circ = 0.7 r Cos[θ] w1 + 0.7 r Sin[θ] w2/.rt;
cylplot = ParametricPlot3D[Evaluate[lin + circ],
{x, -3, 3}, {θ, -0.5π, 0.85π},
Shading → True, DisplayFunction → Identity];
axis = ParametricPlot3D[lin, {x, -3.5, 3.5},
DisplayFunction → Identity];
If[showcyl, showlines[pts, axis, cylplot,
DisplayFunction → Identity],
showlines[pts, axis, DisplayFunction → Identity]]]
```

Let us look at an illustrative configuation. One notes that it is hardly generic in the sense that the points form a double regular tetrahedron (with edge length of $\sqrt{3}$). In fact it was necessary to change the direction and offset parameters to get all six solutions for this example.

```
dpoints = {{1, 0, 0}, {-1/2, Sqrt[3]/2, 0}, {-1/2, -Sqrt[3]/2, 0},
{0, 0, Sqrt[2]}, {0, 0, -Sqrt[2]}};
vec = {a, c, 1}; offset = {b, d, 0};
solveCylinders[pts_List, vec_, offset_,
prec_ : Automatic]:=
Module[{exprs, k, perps},
perps = Table[perp[pts[[k]], vec, offset], {k, 5}];
exprs = (Numerator[Together[#1.#1 - rsqr]]&)/@
perps;
solns = NSolve[exprs, {a, b, c, d, rsqr},
WorkingPrecision → prec]; rs = N[rsqr/.solns]]
solveCylinders[dpoints, vec, offset]
{0.81, 0.81, 0.81, 0.81, 0.81, 0.81}
```

We see that the radii are all equal for this configuration. In fact (as may be verified by setting the optional fourth argument to ∞) the radii are all $\frac{9}{10}$.

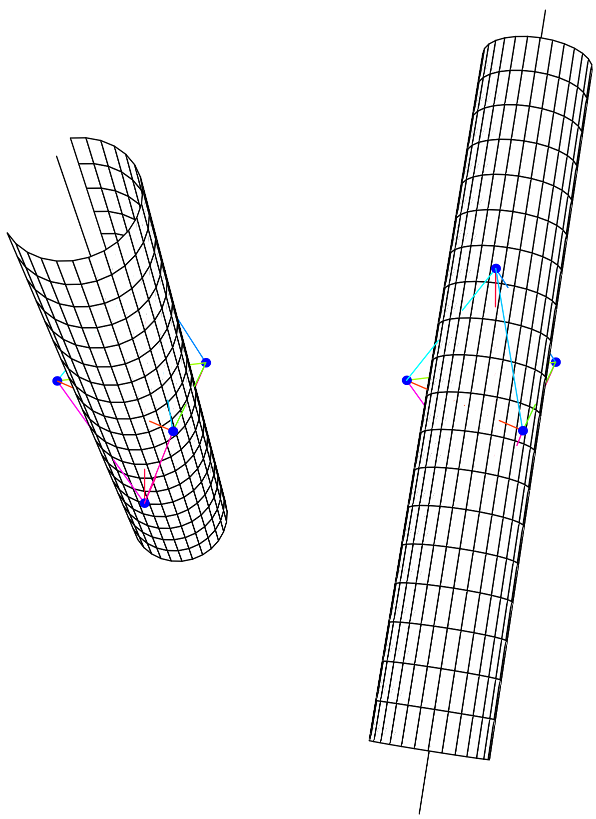
We used a global variable, *solns*, as a convenience so that we can make further use of the full solution set. While this is poor programming style, it suits our immediate purpose.

```
nsols = N[solns]
{{rsqr → 0.81, d → 0., b → 0.1,
a → 0., c → 0.816497}, {rsqr → 0.81,
d → 0., b → 0.1, a → 0., c → -0.816497},
```

$\{\mathrm{rsqr} \to 0.81, d \to 0.0866025, b \to -0.05,$
$a \to 0.707107, c \to 0.408248\},$
$\{\mathrm{rsqr} \to 0.81, d \to -0.0866025, b \to -0.05,$
$a \to -0.707107, c \to 0.408248\},$
$\{\mathrm{rsqr} \to 0.81, d \to -0.0866025, b \to -0.05,$
$a \to 0.707107, c \to -0.408248\},$
$\{\mathrm{rsqr} \to 0.81, d \to 0.0866025, b \to -0.05,$
$a \to -0.707107, c \to -0.408248\}\}$

Here are plots of two of the cylinders.

nsols = *N*[solns];
Show[GraphicsArray[
Table[cylinderplot[nsols[[*j*]], dpoints, vec,
offset, True], {*j*, 2}]]];



It is interesting to note that from this double regular tetrahedron construction one may obtain twelve real cylinders of a certain radius that intersect four particular points. Such an example was first presented in [26]; here we show

how it arises naturally from our construction above. We begin with a regular tetrahedron and this time glue four others onto it, one on each face. The vertices of the original tetrahedron will be our four points. Clearly from one of the glued on tetrahedra we get the six cylinders from above, each intersecting those four points and all having the same radius. Indeed, each glued tetrahedron gives rise to six cylinders, but they pair off so that the actual total is twelve. We remark that computational techniques essentially identical to those we have shown can also be used to solve for cylinder parameters for this problem. A different approach, using homotopy continuation methods as described e.g. in [30], was employed in chapter 4 of [15]. A generalization of finding a the cylinder of given radius through four points has been studied by [19].

4. Real cylinders: probabilities and configurations

Enumerating real cylinders

We now investigate cases in which a configuration of five points will give rise to the various possible numbers of real cylinders containing it. First we note one obvious situation for which there can be no real cylinders: if one point is inside the convex hull of the other four then no cylinder can contain all five, because right circular cylinders are convex. It would be interesting to know how frequently this arises. To this end we employ a simple simulation. More than 4000 random examples were run to get some idea of how frequently one gets zero, two, four, or six real cylinders, given a certain random distribution for the configuration parameters.

```
Clear[a, b, c, d, rsq];
vec = {1, a, c}; offset = {0, b, d};
pair = NullSpace[{vec}];
{w1, w2} = GramSchmidt[pair];
vars = {a, b, c, d, rsqr};
SeedRandom[1111];
len = 2^12;
intpoints =
Table[Table[Random[Integer, {-100, 100}], {5}, {3}],
{len}];
Timing[
rvals =
Table[
perps = Table[perp[intpoints[[j, k]], vec, offset],
{k, 5}];
exprs =
Map[Numerator[Together[#.# - rsqr]]&,
perps];
solns = NSolve[exprs, {a, b, c, d, rsqr}];
```

```
rs = N[rsqr/.solns];
{j, Cases[rs, _Real]}, {j, len}]; ]
{1113.48Second, Null}
rvals2 =
Sort[rvals, Length[#1[[2]]] ≤ Length[#2[[2]]]&];
lens = (Length[#1[[2]]]&)/@rvals2;
lenlens = Length/@Split[lens]
{931, 2206, 865, 94}
```

Roughly 23% of the cases give no real cylinders. It is natural to ask whether these are all cases in which one point is enclosed by the other four. This turns out not to be so. We first discuss the frequency of random examples for which one point is enclosed by the hull of the other four. Later we will see an open set in the configuration space for which no point lies inside the hull of the rest, and for which there are no real cylinders through all points.

Below is a simulation of the one-enclosed-by-four situation. The code below will generate random configurations and then check to see in how many cases one point lies within the convex hull of the rest.

```
plane[p1_, p2_, p3_]:=
With[{norm = Cross[p1 − p2, p1 − p3]},
{norm, norm.p1}]
sameside[{p1_, p2_, p3_}, p4_, p5_]:=
Module[{norm, d},
{norm, d} = plane[p1, p2, p3];
(norm.p4 − d) ∗ (norm.p5 − d) > 0]
encloses[pnts : {p1_, p2_, p3_, p4_}, p5_]:=
Module[{combos, j},
combos = Table[{Drop[pnts, {j}], pnts[[j]]},
{j, Length[pnts]}];
Apply[And, Map[sameside[#[[1]], #[[2]], p5]&,
combos]]]
anyenclosed[pnts : {p1_, p2_, p3_, p4_, p5_}]:=
Module[{combos, j},
combos = Table[{Drop[pnts, {j}], pnts[[j]]},
{j, Length[pnts]}];
Apply[Or, Map[encloses[#[[1]], #[[2]]]&,
combos]]]
SeedRandom[1111];
len = 2^14;
realpoints =
Table[Table[Random[Real, {−100, 100}], {5}, {3}],
{len}];
Timing[
enclosedlist =
Transpose[
{Range[len], Map[anyenclosed, realpoints]}]; ]
{140.66Second, Null}
```

```
hasenclosed = Cases[enclosedlist, {_, True}];
numenclosed = Length[hasenclosed]
N[numenclosed/len]
```
1147
0.0700073

In this simulation run between six and seven per cent of the examples had one point enclosed by the hull of the other four. Thus, for the no-real-cylinder examples, we surmise that roughly three out of four cases do not arise in this way. We will return to this when we discuss enumerative geometry aspects of this cylinder problem.

The frequency of one point being enclosed by the others is related to some old problems in integral geometry. One way to pose this is as a three dimensional version of Sylvester's problem [11]: What is the probability that five points chosen at random in a unit cube all lie on the convex hull they define? Another variant is to find the expected volume of a random tetrahedron in the unit cube (several other variations are posed in the reference). We will call this expected volume *vTet*. To see how these problems are related, we order the five random points, then ask what is the probability that the first is enclosed by the others. This is exactly that expected volume. Now observe that the expected likelihood that any one point is enclosed by the other four is 5 *vTet*, as these are each pairwise exclusive events. Indeed, by taking the average of the five cases of one-point-enclosed-by-the-rest one obtains a Monte Carlo simulation of *vTet*: it appears to be in the ballpark of $\frac{1}{70}$.

Taking this another step we might refine the estimate with a low precision quadrature. The option settings used below were obtained by trial and error, and the nondefault **MaxPoints** setting causes **NIntegrate** to do a quasi-Monte Carlo evaluation. The result is clearly in accord with that above.

```
vol[p1_, p2_, p3_, p4_]:=
Abs[(p2 − p1).Cross[p3 − p1, p4 − p1]]/6
NIntegrate[
Evaluate[vol[{x1, y1, z1}, {x2, y2, z2}, {x3, y3, z3},
{x4, y4, z4}]],
{x1, 0, 1}, {y1, 0, 1}, {z1, 0, 1}, {x2, 0, 1},
{y2, 0, 1}, {z2, 0, 1},
{x3, 0, 1}, {y3, 0, 1}, {z3, 0, 1}, {x4, 0, 1},
{y4, 0, 1}, {z4, 0, 1},
PrecisionGoal → 2, AccuracyGoal → 6,
MaxPoints → 1000000]
```
0.0136429

The problem of finding the expected volume of a tetrahedron with vertices independently and uniformly distributed inside a cube was recently solved [34] using an elaborate breakdown of the region and many multivariate integral computations. The actual value is $\frac{3977}{216000}$-$\frac{\pi^2}{2160}$, or approximately .013843.

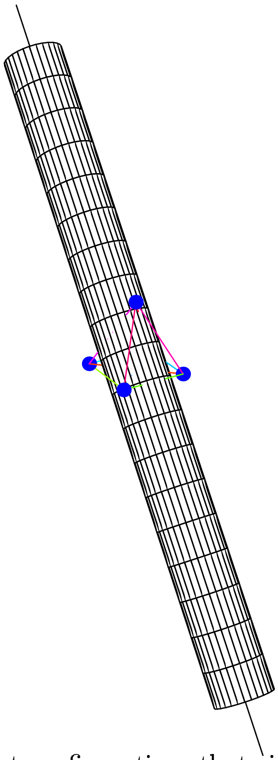
Configurations that give six real cylinders

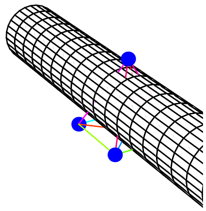
We previously obtained six real cylinders above by starting with a regular tetrahedron and gluing a copy of itself to one face to obtain five points. If the common face is in the x- y plane (so that one tetrahedron points up, the other down), then three cylinders intersect the three faces of the upper tetrahedron (and cut through the edges of the lower), while the other three do just the opposite. In fact it is quite clear by symmetry that if we have one real cylinder then we must have six: we get two "conjugates" by rotating, and three more by reflecting through the x- y plane. There is another configuration, from [29], that can be seen to give six cylinders. We have four points forming vertices of a square in the x- y plane. This is the base of a pyramid with the fifth point as its apex above the centroid of this square. We obtain two horizontal cylinders each passing through a pair of opposite triangular faces of the pyramid. The remaining four each pass through a triangular face, angled upward, and an edge of the base. Some are illustrated below. Note that we again must take care in our choice of axis direction vector and offset, because once again we have a nongeneric choice of coordinates for illustrative purposes. Accordingly the code for **showlines** has been altered to use a **ViewPoint** of {0,0,1}.

```
dpoints = {{1, 0, 0}, {−1, 0, 0}, {0, 1, 0},
{0, −1, 0}, {0, 0, 3/2}};
vec = {1, a, c}; offset = {0, b, d};
solveCylinders[dpoints, vec, offset]
nsols = N[solns];
```

{0.818182, 0.818182, 0.818182,
0.818182, 0.840278, 0.840278}

```
Show[GraphicsArray[
Map[cylinderplot[nsols[[#]], dpoints, vec,
offset, True]&, {3, 5}]]];
```





Conjecture: All five point configurations that give rise to six cylinders in $\mathbb{R}^3$ are small perturbations of one of these two configurations. This idea, admittedly difficult to quantify, is based on visual experimental evidence. It is interesting that one of these examples has a four-fold symmetry rather than exclusively two-or-three-fold symmetries.

(More) configurations that give no real cylinders

As noted earlier, we get no real cylinders whenever one point is in the convex hull of the other four. It is also clear from experiments that there are other configurations that give no real cylinders. To describe one family of such, we return to the double tetrahedron glued along a common face. If we alter either or both of the upper and lower vertices we can jump from having six cylinders through the five points to having none. In the code below we explicitly show this phenomenon by allowing the upper vertex to vary (actually there is no way to impose a positivity constraint so it could be negative. In this case we get no cylinders because either it or the other negative vertex will be in the tetrahedron hull of the remaining four vertices).

dpointslong = {{1, 0, 0}, {−1/2, Sqrt[3]/2, 0},

{−1/2, −Sqrt[3]/2, 0}, {0, 0, *z*}, {0, 0, −1}};
perps = Table[perp[dpointslong[[*k*]], vec, offset],
{*k*, 5}];
exprs = Map[Numerator[Together[#.# − rsqr]]&,
perps]
$\{a^2 + 2ab + b^2 + c^2 + b^2c^2 + 2cd -$
$2abcd + d^2 + a^2d^2 - \text{rsqr} - a^2\text{rsqr} - c^2\text{rsqr},$
$3 + 2\sqrt{3}a + a^2 - 4\sqrt{3}b - 4ab + 4b^2 + 4c^2 -$
$4\sqrt{3}bc^2 + 4b^2c^2 - 4cd + 4\sqrt{3}acd - 8abcd +$
$4d^2 + 4a^2d^2 - 4\text{rsqr} - 4a^2\text{rsqr} - 4c^2\text{rsqr},$
$3 - 2\sqrt{3}a + a^2 + 4\sqrt{3}b - 4ab + 4b^2 + 4c^2 +$
$4\sqrt{3}bc^2 + 4b^2c^2 - 4cd - 4\sqrt{3}acd - 8abcd +$
$4d^2 + 4a^2d^2 - 4\text{rsqr} - 4a^2\text{rsqr} - 4c^2\text{rsqr},$
$b^2 + b^2c^2 - 2abcd + d^2 + a^2d^2 - \text{rsqr} - a^2\text{rsqr} -$
$c^2\text{rsqr} + 2abcz - 2dz - 2a^2dz + z^2 + a^2z^2,$
$1 + a^2 + b^2 - 2abc + b^2c^2 + 2d + 2a^2d -$
$2abcd + d^2 + a^2d^2 - \text{rsqr} - a^2\text{rsqr} - c^2\text{rsqr}\}$
gb = GroebnerBasis[exprs, {rsqr, *d*, *b*, *a*, *c*},
CoefficientDomain → RationalFunctions]
$\{2 + 3c^2 - 4z + c^3(-2 + 2z), -1 - 2c^2 + 2z + a^2(-1 + 2z),$
$-a + ac(1 - z) + b(1 + 2z),$
$4 - c + c^2(2 - 2z) - 4z + d(4 + 8z),$
$-5 - 20z^2 + \text{rsqr}(4 + 16z + 16z^2)\}$

We already know from the symmetry argument that we obtain at least one real solution if and only if we obtain six real solutions. So it suffices to indicate situations where we cannot have six. For this we focus on the univariate polynomial. Dividing by the leading coefficient we have a polynomial that, for z sufficiently large, approximates c^3-2. Clearly these do not have three real roots. Hence the system cannot give rise to six real solutions, and by the symmetry argument there are in fact none.

5. Enumerative geometry of cylinders through five points

Basic theory

The above investigations indicate ways in which one might approach the problem computationally. Now we state and prove several results suggested by the computations of the previous sections.

Proposition 1: Generic configurations of five points in $\mathbb{R}^3$ lie of the surface of finitely many cylinders. Moreover an upper bound on the number of these cylinders is nine.

Proof: This is largely a computational proof which we first describe. We set up some linear algebra similar to that already seen, but now we reduce to two equations in two variables and several configuration parameters. The linear algebra is as follows. Without loss of generality we have one point at the origin, another at {1,0,0}, and a third in the x-y coordinate plane. We project these onto the set of planes through the origin, parametrized (generically) by a normal vector {a,b,1}. We then obtain equations that must be satisfied by the parameters {a,b} in order for the remaining two points to project onto the same circle in that plane as the first three; this is exactly the condition that all five lie on some cylinder. Factoring shows that they are irreducible and hence are relatively prime. So generically they have finite intersection, with upper bound given by the Bezout theorem. In fact, as each polynomial has degree three in the variables {a,b}, we see that there are at most nine solutions for the cylinder axis direction parameters, hence at most nine solutions for the set of cylinder parameters.

```
Clear[a, b];
normal = {a, b, 1};
spanners = GramSchmidt[NullSpace[{normal}]];
points = {{0, 0, 0}, {1, 0, 0}, {x2, y2, 0}, {x3, y3, z3},
{x4, y4, z4}};
projpoint[p_, span_]:= Sum[p.span[[j]]span[[j]], {j, 1, Length[span]}];
projpoints =
Table[Together[projpoint[points[[j]], spanners]],
{j, Length[points]}];
circle[{p1_, p2_, p3_}, normal_]:=
Module[{rsqr, c1, c2, c3, c, cp1, cp2, cp3, gb},
c = {c1, c2, c3};
cp1 = c - p1; cp2 = c - p2; cp3 = c - p3;
polys =
Append[
Thread[{cp1.cp1, cp2.cp2, cp3.cp3} - rsqr],
c.normal];
gb = GroebnerBasis[polys, {c1, c2, c3, rsqr},
CoefficientDomain -> RationalFunctions];
{{c1, c2, c3}, rsqr}/.
Solve[gb == 0, {c1, c2, c3, rsqr}]]
{cen, radsqr} =
First[circle[Take[projpoints, 3], normal]];
vec1 = projpoints[[4]] - cen;
vec2 = projpoints[[5]] - cen;
polys =
Numerator[
Together[{vec1.vec1 - radsqr,
vec2.vec2 - radsqr}]]
```

$\{-x_3y_2 - b^2x_3y_2 + x_3^2y_2 + b^2x_3^2y_2 + x_2y_3 +$
$b^2x_2y_3 - x_2^2y_3 - b^2x_2^2y_3 + 2abx_2y_2y_3 -$
$2abx_3y_2y_3 - y_2^2y_3 - a^2y_2^2y_3 + y_2y_3^2 + a^2y_2y_3^2 -$

$bx_2z_3 - b^3x_2z_3 + bx_2^2z_3 + b^3x_2^2z_3 + ay_2z_3 +$
$ab^2y_2z_3 - 2ab^2x_2y_2z_3 - 2ax_3y_2z_3 + by_2^2z_3 +$
$a^2by_2^2z_3 - 2by_2y_3z_3 + a^2y_2z_3^2 + b^2y_2z_3^2,$
$-x_4y_2 - b^2x_4y_2 + x_4^2y_2 + b^2x_4^2y_2 + x_2y_4 +$
$b^2x_2y_4 - x_2^2y_4 - b^2x_2^2y_4 + 2abx_2y_2y_4 -$
$2abx_4y_2y_4 - y_2^2y_4 - a^2y_2^2y_4 + y_2y_4^2 + a^2y_2y_4^2 -$
$bx_2z_4 - b^3x_2z_4 + bx_2^2z_4 + b^3x_2^2z_4 + ay_2z_4 +$
$ab^2y_2z_4 - 2ab^2x_2y_2z_4 - 2ax_4y_2z_4 + by_2^2z_4 +$
$a^2by_2^2z_4 - 2by_2y_4z_4 + a^2y_2z_4^2 + b^2y_2z_4^2\}$

One verifies the claims made above from this computation. □

Remark: We will presently give two proofs that the correct number is six. One method will use the computation shown above.

Remark: In [13] it is noted that this projected circles approach is related to the Delaunay triangulation of projections of the five points on all possible planes. Specifically, directions of projection where the triangulation changes are important, as these occur exactly when four points become collinear. This gives a direct tie between the enumerative and computational geometry aspects to the problems under scrutiny.

Proposition 2: Real valued solutions always have positive values for the square of the radius.

Remark: The significance of this proposition is that all real valued solutions do indeed give cylinders in $\mathbb{R}^3$.

Proof: Suppose we form a lexicographic Gröbner basis with the radius-square variable ordered as smallest. Then generically (shape lemma) we have a basis containing a univariate polynomial in that variable. For each of the other variables there will correspond a linear polynomial in the basis, and it will have real valued coefficients. Suppose a solution to that univariate polynomial is real valued. Then the remaining cylinder parameters, on back substitution, will also be real valued as they are given by linear polynomials over the reals. Now recall that our original equations were of the form sum-of-squares=radius-squared where the left hand side is a polynomial function of the input data and cylinder parameters. Hence all the original equations will have positive left hand sides, so the radius squared must also be positive. □

Theorem 1: Five generic points in $\mathbb{R}^3$ determine six distinct sets of cylinder parameters, of which an even number are real valued.

The proofs of this theorem are deferred to the following subsection.

Remark: One way to prove this would be to do a brute force computation of a Gröbner basis for a system with generic configuration parameters. But this is not possible with software and methods available to the author. Some evidence furthermore suggests that the problem is computationally quite difficult. Indeed, using the linear algebra setup first shown, it was quite strenuous to compute a Gröbner basis when we made a generic parameter of just one coordinate of one point.

Proposition 3: Configurations that give rise to an empty solution set or to a solution set of positive dimension lie on a variety.

Outline of proof: One obtains, in principle, a description of the generic solutions by forming a lexicographic Gröbner basis for the system with indeterminate data. The process of doing this gives rise to the generic basis because at steps along the way one is allowed to divide by polynomials in the indeterminates. All inputs that fail to give the generic basis thus must satisfy conditions among the coordinates that cause these polynomials, upon specialization, to vanish. As there are finitely many steps in forming the basis, there are finitely many such conditions. As thesevanishing conditions are defined by polynomials, their union is a variety. We may further refine it. Some configurations might fail to give the generic basis but still yield a nonzero finite solution set. If we exclude the conditions that give this situation, we are still left with a variety for which we get either zero or infinitely many (complex valued) solutions. □

Definition: Generically we have six solutions for the cylinder parameters, given five points in $\mathbb{R}^3$ (or $\mathbb{C}^3$, for that matter). Above we saw that the subset of nongeneric configurations that give either zero or infinitely many solutions comprises a variety. We refer to this as the "bad variety", denoted V_{bad}. Several results below are stated in terms of configurations that miss this variety.

Theorem 2: There is a nonempty open set in our configuration space (which is, in effect, $\mathbb{R}^{15}$) for which we obtain no cylinders in $\mathbb{R}^3$.

Proof: Suppose we have a configuration that gives no real cylinders. Again consider a Gröbner basis with a univariate polynomial in the radius-squared. Then all of its roots are complex valued. Small perturbations of this configuration will still give complex valued roots because the real and imaginary parts vary continuously in the configuration parameters. We already know there exist configurations that give no real cylinders, which establishes that the set is nonempty. □

Corollary: The maximum number of cylinders, already shown to be bounded by nine, is in fact no larger than eight.

Proof: This is a consequence of the following facts. (i) Restricting to real inputs does not move us out of the generic case because this restriction is not algebraic. (ii) Given real data, complex solutions appear in pairs. (iii) The case where one coordinate lies in the hull of the other four contains an open subset in the real part of the parameter space. Hence there is an open set in parameter space for which there are only complex solutions. So in general there must be an even number of solutions. □

Remark: This also shows that the the number cannot be seven. So we know it is either six or eight.

Theorem 3: There is a closed set with nonempty interior in $\mathbb{R}^{15}$-V_{bad} for which we obtain six cylinders real cylinders provided we count solutions to the cylinder parameter equations by multiplicity.

Outline of proof: We may count the number of real roots using the Rule of Signs [28] on the univariate polynomial in the lexicographic Gröbner basis. This gives a closed condition for the boundary of the set of configurations that yield six sets of real valued cylinder parameters. To show it has nonempty interior it suffices to demonstrate one such configuration that has no multiple solutions. But this was the case for both six-real-cylinder examples shown earlier. □

Remark: Similar argument shows that the sets in $\mathbb{R}^{15}$-V_{bad} that give rise to two or four cylinders real cylinders also have nonempty interiors.

Theorem 4: Suppose we have four non-coplanar points in $\mathbb{R}^3$. They are the vertices of some tetrahedron. Then there is an open set S containing the open tetrahedron and a dense subset of its boundary in the configuration space, such that if the fifth is chosen in S there will be no real cylinders containing all five points.

Remark: If the fifth point is inside the convex hull of the other four then we already know this result. Now take the tetrahedron formed by the four points. Through each vertex the planes containing the three coincident faces form a cone with triangular base. If the fifth point lies within that cone then it obscures that vertex, i.e. the vertex lies inside the new tetrahedron formed by the fifth point and the remaining three. Hence this case is also covered by the "one point in the hull of the others" situation. Note that in this case the fifth point need not be near in distance to the other four.

Algebraic proof: Suppose the fifth point lies on a face of the tetrahedron formed by the other four. Then the convexity argument still tells us that no cylinder in $\mathbb{R}^3$ can contain all five points. As we assumed the tetrahedron coordinates are generic, we are in one of two situations: either having the fifth coordinate lie on a face formed by three others puts the configuration in V_{bad} or it does not. We show that generically it does not, or in other words, the algebraic condition that four points are coplanar is not a condition for the bad variety. That this is so follows from the trivial observation (verified computationally) that there are configurations with four coplanar points that give rise to lexicographic Gröbner basis with the generic "shape"; were this a condition to lie in the bad variety then every configuration with four coplanar points would be in it.

The preceding argument shows that generically the fifth point is not on V_{bad}, so the set of such fifth points is dense in the set of all boundary points of the tetrahedron. By genericity we may assume that we have a univariate polynomial of degree six for one of the cylinder parameters. As there are no real cylinders containing this configuration, this polynomial has exclusively nonreal roots. These roots vary continuously with the configuration, hence the imaginary parts remain nonzero under small perturbations of the five points. Thus there is an open set around this point on the boundary for which we still obtain no real roots. As we require real roots in order to obtain cylinders in $\mathbb{R}^3$ this suffices to finish the proof. □

Outline of a geometric proof: (This line of proof was suggested by Dick Bishop [4].) In order for all five points to lie on a cylinder there must be a plane tangent to the unit sphere, and a circle in that plane, such that they all project onto that circle. Suppose the fifth point is inside or on the boundary of the tetrahedron formed by the other four. Then it is clear that the projection of the five points onto any such plane will have the projection of this last point contained in the quadrilateral formed by the projection of the other four. Hence any quadratic in the plane that contains all five projected points must be a hyperbola (because all other quadratic curves are convex). Moreover the parameters of the hyperbola equation are continuous in the locations of the five points. As the set of projection planes is compact, a small perturbation of the fifth point beyond the hull of the other four will not alter the situation that the five points project onto hyperbolas in all such planes, hence they cannot lie on any cylinder in $\mathbb{R}^3$. Hence from every boundary point on the tetrahedral hull of the four points, we may perturb outward some minimal distance (depending on that boundary point) and still have no real cylinders. As the tetrahedron boundary is compact we deduce that there is a minimum positive distance we can move outside and still not get real cylinders. □

Corollary to proof: There is an open set S containing the closed tetrahedron, such that if the fifth is chosen in S there will be no cylinders in $\mathbb{R}^3$ containing all five points. In other words, the "bad" variety in configuration space is not an issue.

Open question: Might one use similar ideas to get a geometrical description of what configurations will give rise to the various numbers of real valued cylinders? Perhaps there is some way to use more refined information about the convex hull to determine when such projections, say in directions that intersect the six faces (if origin is placed appropriately e.g. at the centroid), cannot possibly give circles?

Proof and further remarks for theorem 1

Proof: We know from the proposition that the generic solution set is finite. Moreover as the ideal formed by the cylinder equations is generically radical and in general position with respect to all variables one may apply the Shape Lemma. Hence we may assert that the generic lexicographic Gröbner basis has a univariate polynomial in one variable, say a, of the form

$$p[a] = p_n a^n + p_{n-1} a^{n-1} + ... + p_1 a + p_0$$

where each p_j is a polynomial in the configuration data, that is the coordinates of the five points. We know from the preceding theory that either n=6 or n=8 and wish to show that it must be the former. Note that each of the more than four thousand examples run in the simulation gave exactly six solutions. So we must now show that we did not choose in excess of four thousand unfortunate examples. It will turn out we need only work with one such "good" choice.

Let C_0 denote a configuration of five points that gives rise to exactly six solutions to the cylinder equations. We will show it is generic in the sense that this is the number of solutions in an open set around C_0 in the configuration space. Suppose this is not the case. Then in an open dense set around C_0-V_{bad} there are configurations that give n=8 solutions. We take a sequence of these configurations that approaches C_0. As solutions vary continuously with the configuration in the generic case (which comprises a dense open connected subset of $\mathbb{C}^{15}$) we see that some solutions for the variable a must go to infinity as the configurations approach C_0. This is so because the leading coefficient of p[a] is going to zero; if not then we would have n solutions for C_0 in contradiction to our hypotheses.

A similar argument shows that solutions for the other radial direction parameter, b, also go to infinity. Hence the axial direction for these excess solutions must lie in the x-y coordinate plane, assuming our axial direction was set up to be {a,b,1}. Now we recompute the solution set for C_0 with the axial direction taken to be {a,1,b}, and again repeat with direction vector of {1,a,b}. If we still have only six solutions in all cases then the same reasoning as above demonstrates that the axial directions of the sequence of configurations approaching C_0 must also lie in the x-z and y-z coordinate planes. As no nonzero vector lies in all three coordinate planes we obtain a contradiction.

To finish the proof it suffices to demonstrate a configuration that gives six solutions using all three forms of axial direction vector. For example, take
dpoints = {{1, 2, −4}, {−1, 1, 3}, {−1, −1, 1},
{1, 0, 1}, {6, 1, −1}};

Now simply use the procedure **solveCylinders** above to compute the number of solutions obtained using all three possibilities indicated above for direction vector; verify that six solutions are obtained in each case.

An alternative argument is as follows. Take a given configuration with six solutions, using the setup that forces axial directions going to infinity to do so in the x-y coordinate plane. Rotate it so that no axis moves into that plane, and the axis of rotation does not lie in that plane. Then a symmetry argument shows that the axial directions going to infinity can no longer lie in that plane, but a computational argument shows that they must, a contradiction. □

Remark: A similar argument can be used to show that, if we have n¿ *0* solutions at some point counting multiplicity, then n must be six (that is, it cannot be less). Note, however, that using the computational setup from previous sections one may construct examples with only four solutions. This is an artifact of how we choose coordinates. For example, by insisting that the axis direction vector have a z coordinate value of *1* (which generically is allowable) we may lose valid cylinders in examples that are not generic for that choice. As remarked at the outset, we could instead have worked in projective space, rather than in an affine open set thereof and similarly made the displacement vector generic, but at some computational expense. In particular we would need to use a pair of extra equations (e.g. normalize the direction vector to unity length and make the displacement perpendicular to the direction), and we would also have to work with doubled solutions because a given direction and its negative are equivalent in this setting.

Remark concerning computational approaches to theorem 1: One might wish to use the method of mixed volume to compute the number of solutions [30]. One finds the convex hull of the Newton polytopes of the exponent vectors for each polynomial and then computes a mixed volume. This is easy to do using the computation from the proof of proposition 1. Each of the two polynomials has the same set of power products in {a,b} and specifically the hull of the exponent vectors is given by the vertex set:

$$\{\{0,0\},\{2,0\},\{2,1\},\{1,2\},\{0,3\}\}$$

The volume of this region is *4*. The Minkowski sum of the two polytopes is just the same hull scaled to twice its size, and the mixed volume is equal to the total volume minus the sum of the volumes of each separate hull, or *16-2 *4=8*. So the generic number of solutions for equations with these sets of exponent vectors is *8* rather than *6*. Indeed, one can verify this immediately by solving a pair of random equations that use the same power products. We thus conclude that the entire family of cylinder problems is nongeneric with respect to the theory presented in [30]. A hint as to why this is so may be gleaned from the computational proof of theorem 1 presented next. This sort of nongeneric example is also noted in [20]. The related problem discussed in [15] and [26] similarly fails to be generic for the polyhedral homotopy solving method.

Second proof of theorem 1: We will look at the solutions at infinity for the polynomials shown in the proof of proposition 1. First we will homogenize and then set the homogenizing variable to zero to get the initials (that is, the degree forms).

hompolys = Expand[apolys/.{a → at, b → bt}]/.
$\boldsymbol{t^{n\_} :\to \frac{w^{4-n}}{a}}$**;**
initials = hompolys/.w → 0

$\{-b^3x_2z_3 + b^3x_2^2z_3 + ab^2y_2z_3 -$
$2ab^2x_2y_2z_3 + a^2by_2^2z_3, -b^3x_2z_4 + b^3x_2^2z_4 +$
$ab^2y_2z_4 - 2ab^2x_2y_2z_4 + a^2by_2^2z_4\}$

We now solve for {a,b}.

solns = Solve[initials == 0, {a, b}]

Solve::svars : Equations may not give solutions for all solve variables.
$\{\{a \to -\frac{b(1-x_2)}{y_2}\}, \{a \to \frac{bx_2}{y_2}\}, \{b \to 0\}\}$

We thus obtain three solutions at infinity for the homogenized system (these are simply the directions of the three lines between any pair of the first three points). The number of solutions from the Bezout theorem, nine, counts these three, and hence there are six solutions in affine space. To understand why this is not generic for the mixed volume computation, note that the initials are identical up to a constant multiple; any random perturbation (say, add b^3 to the first polynomial) will give only one solution at infinity, and therefore yield 8 finite solutions. □

Purely computational proofs of theorem 1: We can form a Gröbner basis with respect to a degree based term ordering for the polynomials we created in proposition 1. Looking at the head terms we find that there are 6 monomials in {a,b} that lie are not reducible with respect to this basis and hence 6 solutions to the system [9] [10].

Alternatively we can compute the resultant of the pair of polynomials with respect to one of the two variables. We obtain a (very large) polynomial of degree 6 in the other. This means there are at most 6 solutions. As we already know there can be that many, this suffices to show that there are generically 6 solutions. □

Remark: It may be useful to look at theorem 1 in the context of what are known as comprehensive Gröbner bases [3] [32]. This construction in effect allows one to circumvent the problem that Gröbner bases are not continuous in their input data (and indeed it seems designed in part for the purpose of addressing that defect). Such a basis contains encoded all Gröbner bases for a given ideal under all specializations of the parameters. It does so in essence by doing multiple polynomial reductions, in the sense of Gröbner basis literature, on a given polynomial in the basis. This has the desirable effect of allowing for the possibility that any (nonnumeric) leading coefficients may be zero. The upshot is that the coefficients of the comprehensive Gröbner basis vary continuously in the parameters of the configuration. The typical use of such a basis is in concrete examples when one wishes to make case distinctions based on parameter values. When a lexicographic term ordering is utilized we can say a bit about the structure of such bases in the (generic) case of finite solution sets, using insight gained from our examples.

For instance, suppose we have six distinct solutions in a situation where the Shape Lemma does not apply (we have seen this in the two examples where we obtained six real valued cylinders; in one case all radii were equal and in the other there were two distinct radii, but of course the cylinders were all distinct in both examples). Then the degree of the univariate polynomial in this variable (say, r), in a lexicographic Gröbner basis, must be less than six (otherwise we would have solutions with multiplicity). We also know from the Shape Lemma that the comprehensive Gröbner basis with respect to the same lexicographic term ordering must contain linear polynomials in each of the higher variables, with lead coefficients that generically do not vanish. But by [21] one or more such linear polynomials must vanish for the type of example under consideration (otherwise we would have fewer than six solutions). Hence there must be, in the comprehensive basis, polynomials of higher degree in those variables. Again invoking results from [21] we know that for the corresponding linear polynomials, when their coefficients do not vanish, they are satisfied at values for which their higher degree counterparts must also be satisfied. So such a linear polynomial (say, in the variable b) must have the form

$$p[\text{parameters}](b - q[r])$$

where the second factor divides any corresponding polynomial(s) of higher degree in b. In other words, when the first factor, which involves only parameters, is nonzero, then where the second is satisfied all those in higher degree must also be satisfied; moreover they cannot all vanish when the first factor vanishes, so they must be divisible by the second factor.

Nongeneric configurations

Thus far we have discussed exclusively the generic situation. It is of interest to make a few observations about the nongeneric case. This in turn sheds light on cylinder solutions for point configurations that are generic but "near" to such nongeneric ones.

Proposition 4: Sets of five coplanar points are not generic insofar as they do not give six solutions to the cylinder equations. In general they give four such solutions.

Proof: This follows from the computation performed in the proof of proposition 1. We take the pair of polynomials generated, substitute zero for the two nontrivial z parameters, and compute a Gröbner basis in terms of the cylinder normal parameters {a,b}. This is in the form specified by the shape lemma, and has a univariate polynomial of degree four with a second polynomial linear in the remaining variable. Hence for coplanar configurations there are generically four solutions rather than six. Of course there are further degeneracies that can arise. For example, if four of the points are collinear then there will be infinitely many cylinders containing all five. To finish the proof we must show that there are no cylinders parallel to the x-y coordinate plane (as we tacitly set the z coordinate of the normal vector to 1). But this is clear from the fact that such a cylinder would intersect the coordinate plane in a pair of lines, and generically the five coplanar points do not lie on any pair of lines.□

Corollary: As configurations of five points move toward a (generic) coplanar configuration, two of the six (possibly complex) cylinder solutions go to infinity.

Remark: This shows that in any comprehensive Gröbner basis for the system, using a lexicographic ordering, the univariate polynomial of generic degree six has leading and second coefficients vanish when the points are coplanar. The third coefficient will in general not vanish in this situation. Note also that the solutions set at infinity, consisting of three direction values, was computed in a way that implicitly depended on the five points not being coplanar. Hence the corollary does not conflict with our prior count of solutions at infinity.

We give a converse to proposition 4. We state it as a conjecture because the proof is not rigorous in all details.

Conjecture: Any nongeneric configuration of five distinct points must be coplanar.

Idea of proof: First assume no three points are collinear. We need to show that there cannot be a dimensional component of solutions. For this to happen, the algebraic curves that are solution sets to the two polynomial equations for two direction variables must share a component. That is, on a component of directions in which four points project onto a circle, the fifth must project onto that same circle. But for the first three points fixed and no three collinear, that direction curve component uniquely defines the fourth point. Hence the fifth must coincide with one of the other four. The case of three points collinear forces the direction vector to define that line, and one must look at the other cylinder parameters to deduce a contradiction. The fifth point cannot lie on the line containing the threesome (otherwise the five are coplanar). As the set of cylinders containing the first four now uniquely defines the line containing the fourth and parallel to the cylinder axis (that is, parallel to the line containing the collinear threesome), the fifth must lie on that line. But this too makes the five points coplanar. □

Remark: The above argument leads one to believe that the cases that give infinitely many solutions involve either four collinear points or three collinear with the line between the remaining two parallel to the collinear triad. Further evidence to support this may be deduced from dimensional considerations. First note that, once three noncollinear points are fixed in a plane, there are finitely many ways to combine the remaining two points such that one of the above conditions holds. For any such combination, there are two degrees of freedom in how the points are placed. Now observe that cubic equations of the form imposed by our choices above has eight degrees of freedom (general cubics have ten coefficients but ours lose one degree three term, and the cubics are only defined up to nonzero scalar multiplication, giving eight degrees of freedom). Hence pairs of cubics of that form have sixteen parameters. The set of pairs we can actually attain has eight degrees of freedom (from the eight coordinates not a priori known). In order that a pair share a component, they must factor (they cannot be identical unless a pair of points coincides, in contradiction of hypotheses). The set of pairs that share a common factor has dimension ten. Thus we expect the dimension of the set of attainable cubic pairs that share a component to be given by the intersection dimension, which is two.

We now describe what happens in the generic coplanar case.

Theorem 5: Given five coplanar points in $\mathbb{R}^3$ there are four (complex) cylinders containing them. Of those, either zero or two will be real cylinders.

Remark: It is interesting to note that the full set of four solutions cannot all be real valued.

Proof: That there are four complex cylinders was noted in the proof to proposition 4. As is well known, the five points uniquely determine a quadratic curve in the plane in which they lie. The intersection of a cylinder with a plane is likewise a quadratic in that plane, and hence any cylinder containing five coplanar points contains the entire quadratic curve they determine. If that curve is a hyperbola then no (real) cylinder can contain it. If instead it is an ellipse then there are two cylinders that contain it. These cylinders have radial axes that each go through the center of the ellipse and lie in the plane perpendicular to the ellipse minor axis. Their angle of intersection is determined by the eccentricity of the ellipse. $\square$

Corollary: Five point configurations sufficiently close to coplanar are contained by at most two real cylinders.

Remark: The example we constructed from a pair of elongated tetrahedra that gives no real cylinders is a case where a configuration approaches coplanarity.

We can use the computational construction of proposition 1 to shed light on the problem of counting the number of cylinders of a given fixed radius through four points (which, as noted in [26], is equivalent to the problem of counting the number of lines simultaneously tangent to four given spheres of equal radius). As the radius is fixed (say, to 1), we are no longer free to rescale so we would use $\{x_1,0,0\}$ for our second point. We would modify *circle*to use only two points along with the given radius. One change now is that for projections of two points onto any given plane, there are two circles of the given radius containing them. This ansatz would lead us to expect twice as many solutions for this problem as we obtained for counting cylinders through five points. That there are in fact twelve (not necessarily real valued) cylinders of given radius through four generically placed points is a theorem in [26]. In the special case that the points are coplanar, that there are eight such cylinders is a result of [27]. In contrast to theorem 5 above, all of them can be real valued.

A partial converse to theorem 4

We wish to show that all configurations that give no real cylinders arise in the setting of theorem 4. Specifically we state the following conjecture.

Conjecture: Suppose we have a configuration of five points in $\mathbb{R}^3$ for which no real cylinders exist, and moreover assume that no point lies in the hull of the others. Then one of the points can be moved anywhere inside the convex hull of the full set and still we will get no real cylinders. In particular we could move this point along a line segment from outside to inside the hull of the other four, and at no point on that path would we get real cylinders. Thus we could regard the given configuration as a perturbation of one that has one point inside the hull of the other four, effectively providing a converse to theorem 4.

We make observations of sufficient conditions for a proof, then state as a theorem a special case wherein we can fullfill the conditions. First we observe that the two curves in direction parameter space, as given by solutions to the two polynomials we constructed in proposition 1, are cubics that have one or more closed topological components in two dimensional real projective space, and because they are cubic they each have at least one real component that go to infinity. (It is well known that they are connected as complex curves; by "components" we mean the obvious thing with respect to intersections with real space.)

We regard each point on such a curve as a solution to the direction parameter equations given by the four points in configuration space lying on a cylinder with axis in that direction. In other words, each point on the solution curve in parameter space defines a cylinder through the four points in configuration space that were used to form that equation. Suppose that at solution on one such component, the fifth point lies inside the cylinder thus obtained (we are being loose with terminology but trust the meaning of "inside a cylinder" is clear). Then it must lie inside all cylinders defined by points on that affine component of the solution curve. The proof of this remark has a small complication. We next claim that the fifth point, in order to "escape" outside the cylinder containing the other four, must cross the cylinder (in contradiction to our hypothesis that there are no real cylinders containing all five points). A priori there is another way it might escape: the cylinder containing four can degenerate to a plane and subsequently reverse its "open" side, if the four points project onto a line for some solution direction. To see this does not happen for generic configurations, note that any such direction must lie in all four planes containing three of the four points. Thus the fifth point stays inside the cylinders defined by all directions on that component of the solution curve for directions of cylinders containing the other four points.

Next we observe that, were this true not just on one topological component of the direction solution curve, but on all of them, then that fifth point works in the conjecture. We see this as follows. When the fifth point lies inside all cylinders defined by a solution curve component, then it projects along the cylinder axis to a point inside the circle that intersects the projections of the other four. The same must hold for any other point in the interior of the convex hull of the five points. This is because such a point, written as a convex combination of the five, must either be in the interior of the tetrahedron defined by the first four (and thus project to the interior of the circle they define), or else have a nontrivial component of that fifth point and again project to that circle interior.

We now proceed to construct a solution on one direction curve, that is, a cylinder containing four of the points, such that the fifth is inside it. We can arrange our four so that three are in the x-y coordinate plane, and the remaining two have a segment joining them that intersects the triangle defined by the first three (such an arrangement can always be found for a configuration of five points in $\mathbb{R}^3$). We place the fourth point on the z axis beneath the origin. Projecting from the fourth point onto a plane in the direction of the segment between the fourth and fifth points gives a unique a circle containing the first three. The cylinder along that direction and containing that circle thus encloses the fourth and fifth points. Now we simply move one of the direction coordinates, forming new projections and cylinders containing the first three points, until one of the remaining points (say, the fourth) hits that cylinder. What we have done is to arrive on the first of the two solution curves for the equation for directions for cylinders containing the first four four points, as set up in proposition one. We now have a cylinder containing four points and enclosing the fifth. From the discussion above we know that this holds for all cylinders defined by this affine component of the curve of directions.

At this point we have a sufficient condition for the conjecture to hold. We simply require that each of the solution curves have only one affine component.

Theorem 6: Given five points in $\mathbb{R}^3$ for which there are no real solutions to the cylinder equations, suppose there are three such that
(i) The segment joining the remaining two lies in the triangle defined by those three.
(ii) The two curves of solution directions for cylinders containing those three and either the fourth or fifth respectively, each have only one component in real projective space.
Then either the fourth or fifth point can be moved anywhere inside the hull of the five and there will be no real cylinder containing this new point and the other four.

As remarked above, we can always order the points in such a way that the first condition holds. But then in general the second condition will not hold. We believe the conjecture to be true all the same, though we do not have a proof as yet. We also mention that extensive graphical evidence suggests that most often these curves have one component in two dimensional real projective space. This is found by taking random examples with three points in the x-y coordinate plane and the fourth and fifth above and below respectively, throwing away those that have real solutions, throwing away from the rest those for which the segment between fourth and fifth points does not go through the triangle bounded by the first three, and plotting the zero level sets for the two cylinder equations in remaining cases. In any case it would seem that this method applies to "most" of the configurations that give no real cylinder solutions.

Indeed, we can weaken the second hypothesis of theorem 6 so that the curves may have multiple components, provided the components for one are not separated by any component of the other. Graphical evidence supports the belief that this weaker requirement is always satisfied. Clearly a proof to this effect would suffice to prove the conjecture.

6. Summary

We have discussed computational methods for finding cylinders through a given set of five points in $\mathbb{R}^3$. Along the way we have covered several related problems and computational approaches thereto. We have investigated the various real valued scenarios using first simulation and then theoretical approaches. In particular we have combined geometric reasoning with Gröbner bases and several related tools from computational algebra, in order to study a rich family of problems from enumerative and computational geometry.

7. Acknowledgements

The author is pleased to acknowledge helpful conversations with Dick Bishop, John D'Angelo, Dan Grayson, and Dave Rusin (the last conducted via email). He thanks David Cox and Thorsten Theobald for bringing to his attention the earlier work of Bottema and Veldkamp. He thanks the Math Department of the University of Illinois for extending their hospitality during the academic year 1998-9 when he was a visitor there, during which time part of this work was done. In particular he thanks Jack Wetzel for inviting him to speak on this topic at the weekly geometry seminar (and for buying him lunch anyway after having sat through said talk).

8. References

1. W. Adams and P. Loustaunau. *An Introduction to Gröbner Bases.* Graduate Studies in Mathematics **3**. American Mathematical Society, 1994

2. E. Becker, M. G. Marinari, T. Mora, and C. Traverso. The shape of the Shape Lemma. In ISSAC '94, *Proceedings of the International Symposium on Symbolic and Algebraic Computation.* 129-133. ACM Press, 1994.

3. T. Becker, W. Weispfenning, and H. Kredel. *Gröbner Bases: A Computational Approach to Computer Algebra.* Graduate Texts in Mathematics **141**. Springer-Verlag, 1993.

4. R. Bishop. Private communication, 1999.

5. O. Bottema and G. Veldkamp. On the lines in space with equal distances to n given points. *Geometriae Dedicata* **6**:121-129. D. Reidel Publishing Company, 1977.

6. B. Buchberger. Gröbner bases: An algorithmic method in polynomial ideal theory. In *Multidimensional Systems Theory*, chap 6. N. K. Bose, ed. D. Reidel Publishing Company, 1985.

7. B. Buchberger. Applications of Gröbner bases in non-linear computational geometry. In *Proceedings of the International Symposium on Trends in Computer Algebra '87*, R. Janssen, ed. Lecture Notes in Computer Science **296**:52-80. Springer-Verlag, 1988.

9. R. Corless. Editor's corner: Gröbner bases and matrix eigenproblems. *ACM Sigsam Bulletin: Communications in Computer Algebra* **30**(4):26-32, 1996.

10. D. Cox. Introduction to Gröbner bases. In *Proceedings of Symposia in Applied Mathematics* **53,** D. Cox and B. Sturmfels, eds. 1-24. ACM Press, 1998.

11. D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Computer Algebra.* Undergraduate Texts in Mathematics. Springer-Verlag, 1992.

12. H. T. Croft, K. Falconer, and R. T. Guy. *Unsolved Problems in Geometry.* 54-55. Springer-Verlag, 1991.

13. O. Devillers, B. Mourrain, F. Preparata, and P. Trebuchet. On circular cylinders by four or five points in space. *Discrete and Computational Geometry* **28**:83-104, 2003.

14. O. Devillers and F. Preparata. Evaluating the cylindricity of a nominally cylindrical point set. In SODA 2000, *Symposium on Discrete Algorithms*, pages 518-527, 2000.

15. C. Durand. Symbolic and Numerical Techniques for Constraint Solving. Ph.D thesis. Purdue University, Department of Computer Science, 1998.

16. P. Gianni. Properties of Gröbner bases under specialization. In Eurocal '87, *European Converence on Computer Algebra.* J. Davenport, ed. Springer Lecture Notes in Computer Science **378**, pages 293-297. Springer-Verlag, 1987.

17. P. Gianni and T. Mora. Algebraic Solutions of systems of polynomial equations using Gröbner bases. In AAECC 5, *Applied Algebra, Algebraic Algorithms, and Error- Correcting Codes.* L. Hugeut, A. Poli, eds. Springer Lecture Notes in Computer Science **356,** pages 247-257. Springer-Verlag, 1988.

18. C. Hoffmann. *Geometric and Solid Modelling: An Introduction.* Morgan Kaufmann, 1989

19. C. Hoffmann and B. Yuan. There are 12 common tangents to four spheres, 2000. Website URL:
http://www.cs.purdue.edu/homes/cmh/distribution/SphereTangents.htm

20. B. Huber and B. Sturmfels. Bernstein's theorem in affine space. *Discrete and Computational Geometry* **17**:137-141, 1997.

21. M. Kalkbrenner. Solving systems of algebraic equations by using Gröbner bases. In Eurocal '87, *European Conference on Computer Algebra.* J. Daven-

port, ed. Springer Lecture Notes in Computer Science **378**, 282-292. Springer-Verlag, 1987.

22. M. Kalkbrenner. Implicitization of rational parametric curves and surfaces. In AAECC8, *Applied Algebra, Algebraic Algorithms, and Error- Correcting Codes.* S. Sakata, ed. Springer Lecture Notes in Computer Science **508**, pages 249-259. Springer-Verlag, 1990.

23. I. Kotsireas. Homotopies and polynomial system solving I: Basic principles. A *CM Sigsam Bulletin: Communications in Computer Algebra* **35**(1):19-32, 2001.

24. D. Lichtblau. Usenet news post to sci.math.symbolic, 1998. Google URL:
http://groups.google.com/groups?q=Lichtblau+cylinder+group:sci.math.symbolic&hl=en&lr=&ie=UTF-8&group=sci.math.symbolic&safe=off&selm=36407D25.7E3FCDAD%40wolfram.com&rnum=1

25. D. Lichtblau. Solving finite algebraic systems using numeric Gröbner bases and eigenvalues. In SCI2000, Proceedings of the World Conference on Systemics, Cybernetics, and Informatics, Volume 10. (Concepts and Applications of Systemics, Cybernetics, and Informatics). M. Torres, J. Molero, Y. Kurihara, and A. David, eds., pages 555-560. International Institute of Informatics and Systemics, 2000.

26.I. G. Macdonald, J. Pach, and T. Theobald. Common tangents to four unit balls in $\mathbb{R}^3$. *Discrete and Computational Geometry* **26**:1-17, 2001.

27. G. Megyesi. Lines tangent to four unit spheres with coplanar centres. *Discrete and Computational Geometry* **26**:493-497, 2001.

28. B. Mishra. *Algorithmic Algebra.* Springer-Verlag, 1993.

29. D. Rusin. Private communication, 1998. Website URL:
http://www.math-atlas.org/98/5pt.cyl

30. B. Sturmfels. Polynomial equations and convex polytopes. *American Mathematical Monthly* **105**:907-922, 1998.

31. J. Vershelde, P. Verlinden, and R. Cools. Homotopies exploiting Newton polytopes for solving sparse polynomial systems. *SIAM J. Numer. Anal.* **31**(3):915-930, 1994.

32. V. Weispfenning. Comprehensive Gröbner bases. *Journal of Symbolic Computation* **14**:1-29, 1992.

33. S. Wolfram. *The Mathematica Book* (5th edition). Wolfram Media, 2003.

34. A. Zinani. The expected volume of a tetrahedron whose vertices are chosen at random in the interior of a cube. *Monatshefte Math.* **139:**341-348, 2003.