Problem solving in calculus with symbolic geometry and CAS

Philip Todd
Saltire Software
<philt@saltire.com>

James Wiechmann
Tualatin High School, Tualatin Oregon, USA
<jwiechmann@ttsd.k12.or.us>

Computer algebra systems (CAS) have been around for a number of years, as has dynamic geometry. Symbolic geometry software is new (Todd, 2007). It bears a superficial similarity to dynamic geometry software, but differs in that problems may be set up involving symbolic variables and constants, and measurements are given as symbolic expressions. Mathematical expressions can be copied back and forth between a symbolic geometry system and a CAS thus making it an interesting new tool for exploring mathematics and solving problems.

The interface between the CAS and the symbolic geometry system has been engineered to be simple and intuitive. Complicated expressions are copied from the symbolic geometry system and pasted into the CAS in the same way that text is copied and pasted from a document. Simplified or derived expressions are copied from the CAS and pasted back into the symbolic geometry system for further investigation. The mathematics, then, takes place in the interface between the computer systems and the computer user. Which expressions are in need of simplification? How were they constructed, and why? Perhaps most importantly, what further avenues of study are made possible through this interaction?

We illustrate the use of a symbolic geometry system applied to two classic optimisation problems, both drawn from *100 Great Problems of Elementary Mathematics* (Dorrie, 1965). The first problem, "Regiomontanus' Optimisation Problem" is suitable for students at the upper secondary level. The second, "Fagnano's Perimeter Minimisation Problem" does involve some partial derivatives, and so would be more suitable for students at tertiary level.

Our symbolic geometry system is *Geometry Expressions* (Saltire Software, 2008) In order to illustrate the flexibility of the CAS interface, in the first problem we use *Mathematica* (Wolfram Research, 2008), and in the second problem, we use *Maple* (Maplesoft, 2008).

Regiomontanus' optimisation problem

Regiomontanus' optimisation problem can be phrased as follows: at what point on the earth's surface would a vertically suspended bar appear longest? The problem is equivalent to maximising the angle *CED* in Figure 1.

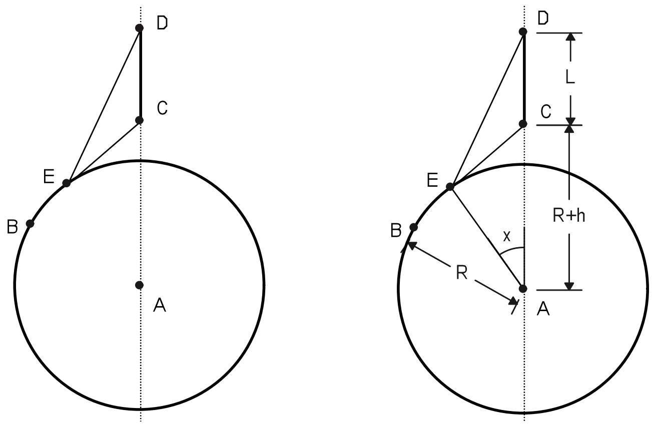


Figure 1. (a) Regiomontanus' optimisation problem.
(b) Addition of symbolic constraints on the drawing allows the problem to be expressed algebraically.

A proof strategy, for the student with a calculus background, might be to express angle *CED* as a function of some aspect of the observer's location, differentiate the function and solve to find the location where the derivative is 0. To carry out this strategy with the symbolic geometry software *Geometry Expressions* (Saltire Software, 2008), the student would need first to specify the problem further by adding some symbolic constraints to the diagram.

Symbolic constraints differ from names or labels in that they add concreteness to the drawing. *AE* refers to the length of segment *AE*, but constraining *AE* to length *R* gives it a particular length. It may be an arbitrary length, but its specification is important to the development of a solution. Thus, the choice of constraints is a critical step in solving problems with symbolic geometry. In Figure 1(b), the student could specify the radius of the earth as *R*, the length of the bar as *L*, and height of the bar above the earth as *h*. The location of the observer could be described via *x*, the measure of angle *A*.

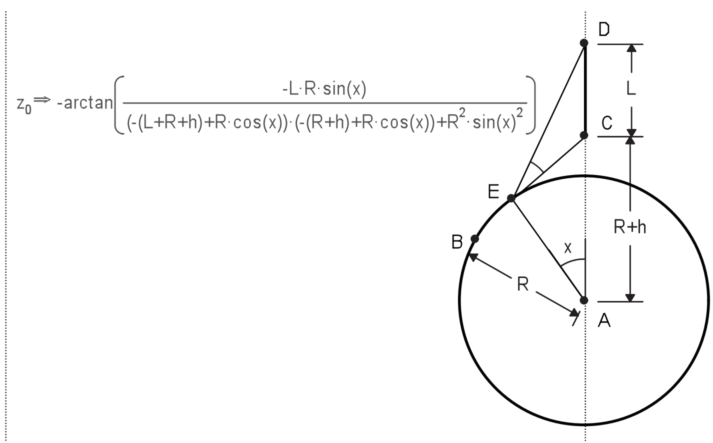


Figure 2. Trigonometric expression for the angle CED, derived by the system and given the name z_0.

Given this input, the symbolic geometry system can automatically generate an expression for angle *CED* (Figure 2) However, demystification of this expression would be appropriate in a learning environment. Clearly, the geometry system has demonstrated that angle *CED* can be described in terms as a function of *x*, but how?

Some classroom brainstorming (perhaps with teacher prompting) will lead to the addition of segment *EG*, perpendicular to the line containing segment *CD* to the drawing, as shown in Figure 3.

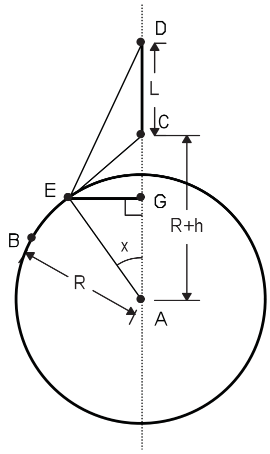


Figure 3. Adding segment EG leads to verification of the function describing angle CED.

With the length of segment EG equal to $R\sin(x)$ and the length of segment AG equal to $R\cos(x)$, students will be able to find their way to the generated expression via the tangent of a difference formula. It is, of course, up to the discretion of the teacher whether this path is pursued, alluded to, or whether the geometry system is taken at face value.

Given a function for the angle measure, it is a simple matter, in theory, to differentiate and solve. With the use of a CAS, it is a simple matter in practice too. The angle formula can be cut from *Geometry Expressions* and pasted into a CAS, the expression differentiated then solved. Figure 4 shows the steps required in carrying out this process in *Mathematica* (Wolfram Research, 2008).

In[9]:= $\mathbf{D\ ArcTan\frac{L\,R\,Sin\,x}{h\ R\ \ R\,Cos\,x\quad h\ L\ R\ \ R\,Cos\,x\quad R^2\,Sin\,x\ ^2}\ ,\ x}$

Out[9]= $\left(\dfrac{L\,R\,Sin\ x\quad 2\,R^2\,Cos\ x\ \ Sin\ x\quad R\ h\ R\ \ R\,Cos\ x\quad Sin\ x\quad R\ h\ L\ R\ \ R\,Cos\ x\quad Sin\ x}{h\ R\ \ R\,Cos\ x\quad h\ L\ R\ \ R\,Cos\ x\quad R^2\,Sin\ x\ ^{2\ 2}} \quad \dfrac{L\,R\,Cos\ x}{h\ R\ \ R\,Cos\ x\quad h\ L\ R\ \ R\,Cos\ x\quad R^2\,Sin\ x\ ^2}\right)$

$\left(1\quad \dfrac{L^2\,R^2\,Sin\ x\ ^2}{h\ R\ \ R\,Cos\ x\quad h\ L\ R\ \ R\,Cos\ x\quad R^2\,Sin\ x\ ^{2\ 2}}\right)$

In[10]:= **Solve 0, x**

Solve::ifun :
Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information

Out[10]= $x\quad ArcCos\,\dfrac{R\ 2h\ L\ 2R}{h^2\ hL\ 2hR\ LR\ 2R^2}\ ,\ x\quad ArcCos\,\dfrac{R\ 2h\ L\ 2R}{h^2\ hL\ 2hR\ LR\ 2R^2}$

Figure 4. Mathematica solution of the optimisation problem.

We notice first that there are two solutions, one of which is the negative of the other. Relating this back to the geometry model of Figure 1, we observe that this corresponds to a pair of solutions one to the left and one to the right of the line *AC*. We select the positive solution and paste it into *Geometry Expressions* for the angle *EAD*.

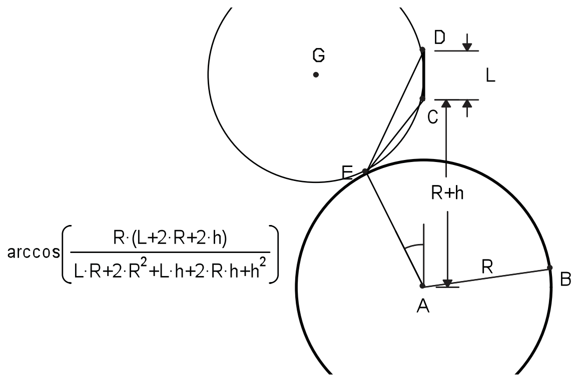


Figure 5. Mathematica solution pasted back into Geometry Expressions; we observe that the circumcircle of CDE appears to be tangential to the circle AB.

Having attained a solution through the rote application of a standard proof technique (and the aid of technology to handle the details), one can recover some geometric insight by drawing the circumcircle of *CDE* (Figure 5). While this might not occur to a student without prompting, consideration of the circumcircle is the key to unlocking a purely geometrical proof of Regiomantus' problem. The circumcircle of *CDE* is of course the locus of the points which subtend the same angle to *CD*. If this was not tangent, there would be a second point of intersection *E′* with the surface of the earth (Figure 6). Between these points the angle subtended would be greater than *CED*.

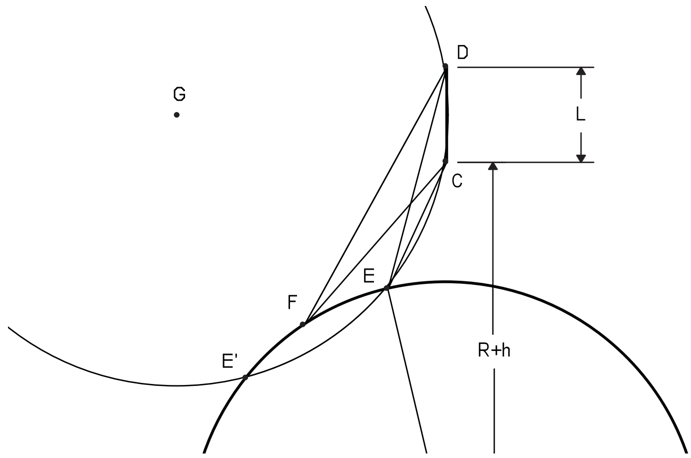
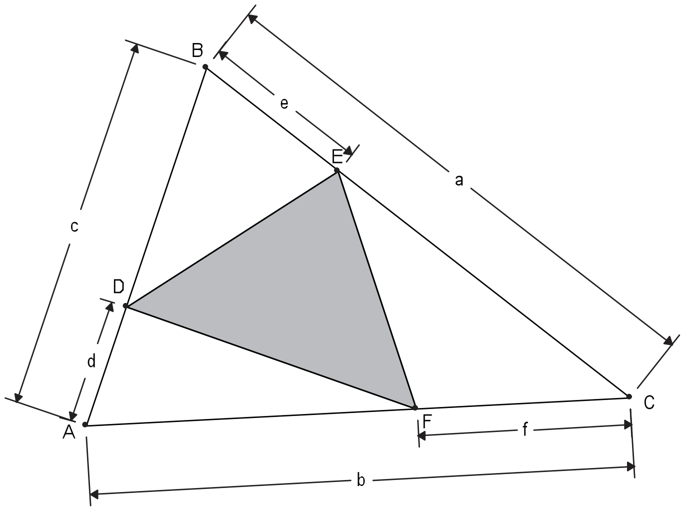


Figure 6. If the circumcircle to CDE is not tangent to the original circle, and F is a point between the two intersections E and E′, then CFD > CED. Hence if CED is maximal, the circumcircle must be tangent.

Fagnano's perimeter minimisation problem

To inscribe in a given acute angled triangle a triangle of minimal perimeter.

To model this situation, we draw a triangle *ABC* and an inscribed triangle *DEF*, where *D* lies on *AB*, *E* lies on *BC* and *F* lies on *AC*. We constrain the model by specifying fixed lengths *AB*, *BC* and *AC*. *AD*, *BE* and *CF* are constrained to specify a particular triangle inscribed within triangle *ABC* (Figure 7). The problem is solved when values for *AD*, *BE* and *CF* are found that minimise the perimeter. The perimeter of the inscribed triangle is computed by *Geometry Expressions*.



$$\Rightarrow \sqrt{a^2-2\cdot a\cdot e+e^2+f^2+f\cdot\left(-b-\frac{a^2}{b}+\frac{c^2}{b}+\frac{b\cdot e}{a}+\frac{a\cdot e}{b}-\frac{c^2\cdot e}{a\cdot b}\right)}+\sqrt{b^2-c\cdot d+\frac{a^2\cdot d}{c}-\frac{b^2\cdot d}{c}+d^2+f^2+f\cdot\left(-2\cdot b+\frac{c\cdot d}{b}+\frac{b\cdot d}{c}-\frac{a^2\cdot d}{b\cdot c}\right)}+\sqrt{c^2-2\cdot c\cdot d+d^2+e^2+e\cdot\left(-a+\frac{b^2}{a}-\frac{c^2}{a}+\frac{c\cdot d}{a}+\frac{a\cdot d}{c}-\frac{b^2\cdot d}{a\cdot c}\right)}$$

Figure 7. Perimeter of the triangle DEF computed by Geometry Expressions.

The solution is obtained through optimisation. The perimeter can be changed by manipulating any or all of the lengths *d*, *e*, and *f*. The problem is solved when the perimeter stops shrinking and begins to grow. Finding the derivative of the expression with respect to each of *d*, *e*, and *f*, setting the derivatives to zero, and then solving the system gives us our solution. The technology allows us to express our intent to a CAS, and then allow it to carry out all of the messy calculations. Here we copy the expression into Maple (Maplesoft 2008), differentiate with respect to *d*, *e* and *f*, and solve:

$$P := \sqrt{a^2-2ea+e^2+f^2+\left(-b-\frac{a^2}{b}+\frac{c^2}{b}+\frac{eb}{a}+\frac{ea}{b}-\frac{ec^2}{ba}\right)}$$
$$+\sqrt{b^2-dc+\frac{da^2}{c}-\frac{db^2}{c}+d^2+f^2+\left(-2b+\frac{dc}{b}+\frac{db}{c}-\frac{da^2}{cb}\right)f}$$
$$+\sqrt{c^2-2dc+d^2+e^2+\left(-a+\frac{b2}{a}-\frac{c^2}{a}+\frac{dc}{a}+\frac{da}{c}-\frac{db^2}{ca}\right)e}$$

\> solve({diff(P,d)=0,diff(P,e)=0,diff(P,f)=0},{d,e,f});

Warning, solutions may have been lost

$$\left\{d=\frac{c^2+b^2-a^2}{2c}, f=\frac{b^2-c^2+a^2}{2b}, e=\frac{b^2-a^2-c^2}{2a}\right\},$$

$$\left\{d=-\frac{\left(-b+f+\frac{afb}{fa-ba+eb}-\frac{af^2}{fa-ba+eb}\right)c}{b}, f=f, e=e\right\}$$

$$\left\{d=-\frac{\left(-b+f-\frac{a^2fb}{fb^2-fc^2-a^2b+bea}+\frac{a^2f^2}{fb^2-fc^2-a^2b+bea}\right)c}{b}, f=f, e=e\right\}$$

Working out these solutions by hand is tedious, daunting, and subject to error. Using the CAS allows us to stay focused on the problem. Handling a complex expression through cutting and pasting certainly makes it less formidable, and in turn, the student less intimidated.

The first solution looks promising. The symmetry in the expressions for *d*, *e* and *f* seem to ring true, since there is no notion of order or precedence among the sides of the triangle. We can copy this back into Geometry Expressions, replacing the parameters *d*, *e* and *f* with their solution values:

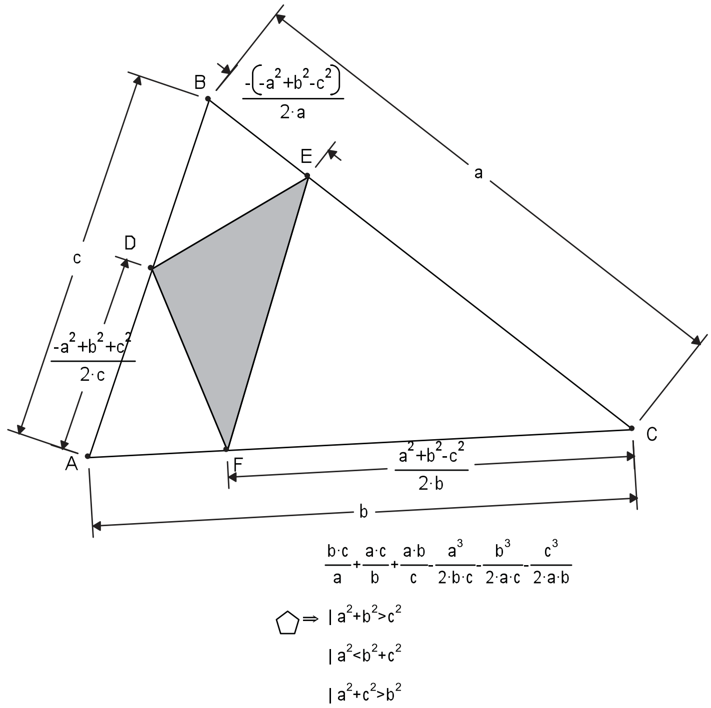


Figure 8. Maple solution expressions for d, e, f copied into the Geometry Expressions model.

We have the solution and the minimal perimeter. However, we reached the solution without a lot of effort or thought. Perhaps we can find something geometrically significant about our solution that will lead us back to a more elegant proof.

Querying *Geometry Expressions* reveals that *D*, *E* and *F* are the feet of the altitudes of the triangle.

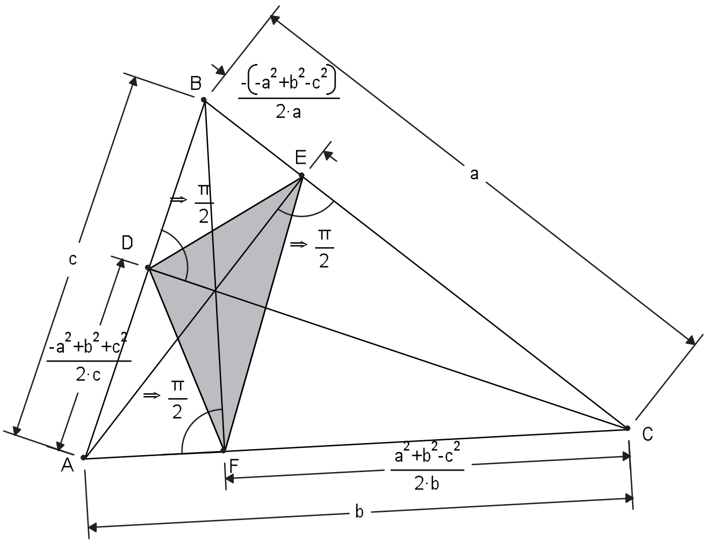


Figure 9. Geometry Expressions reports that in the solution configuration AE is perpendicular to BC, CD is perpendicular to AB and BF is perpendicular to AC.

Some additional reflection and insight brings this fact into focus: the shortest distance between a point and a line is along the perpendicular, and the altitude of a triangle is just such a distance. We should be able to work that fact into the beginnings of a more purely geometric proof.

Select a position for vertex *F*, and reflect it across segments *AB* and *BC*. The line containing both reflections of *F* also contains the other two vertices of the inscribed triangle. Then, since the triangle made by the reflections of *F* and by *B* is isosceles, length *DF* is minimised by minimising *BF'*. *BF'* is just a reflection of *BF*, and the minimum length for *BF* is the altitude (Dorrie, 1965).

The thing to note here is that geometrical insight came after a relatively routine approach actually solved the problem. Students are often asked to prove theorems that are completely stated. In this case, the theorem's consequent is not immediately known. The students are able to form the conjecture themselves before setting out to prove the result.

By the way, what are the other solutions from *Maple*? One of them is shown in Figure 10 (over page).

Discussion

In the above example, the technology facilitates the application of standard calculus techniques to a problem of moderate complexity. Without technology, students often become lost in the magnitude of the mathematical

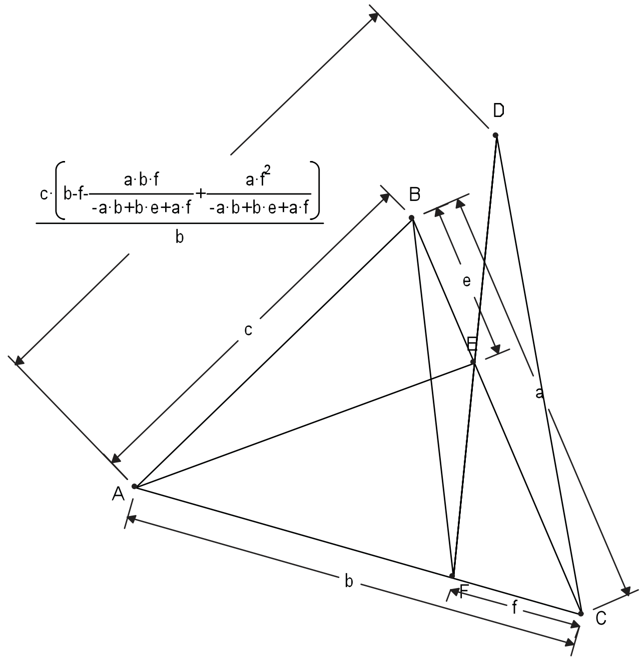


Figure 10. Degenerate solution generated by Maple. Is this a local maximum, or minimum?

expressions that need manipulating, or misled by trivial calculation errors. Symbolic geometry and CAS can raise the students above this noise level, allowing them to see the problem as a whole. Alternatively, technology can allow students to arrive at solutions without giving the problem much consideration all. The student has the responsibility of framing the problem in a meaningful way: he has to decide what lengths and angles should be specified, and what the appropriate objective function should be. He has to direct the application of the standard technique of calculus optimisation. Having attained a solution in this fashion, it can be instructive to re-examine the problem from a purely geometric perspective. In this way the student can be exposed to multiple representations generating multiple quite independent methods of proof. The calculus method, while general, uses a sophisticated mathematical bag of tricks. The geometrical technique on the other hand, requires more mathematical insight, but a lower level of mathematical knowledge.

Reference

H. Dorrie (1965). *100 Great Problems of Elementary Mathematics.* New York: Dover.

Maplesoft (2008). *Maple* [computer software]. Ontario, Canada: Waterloo.

Saltire Software (2008). *Geometry Expressions* [computer software]. Tigard OR, USA: author.

P. Todd (2007). Geometry expressions: A constraint based interactive symbolic geometry system. In F. Botana & T. Recio (Eds), *Automated Deduction in Geometry* (pp. 189–212). Springer.

Wolfram Research Inc. (2008). *Mathematica* [computer software]. Champaign, IL, USA: author.