

Using Symbolic Computation to Inductively Prove Geometric Theorems And Its Implication to The Study of General Relativity

Arief Hermanto

Physics Department
Gadjah Mada University
Yogyakarta, Indonesia
arief_hermanto@ugm.ac.id

Abstract—One can reasonably say that we always prove geometric theorems using deductive method. The deductive method is too often used such that we get the impression that there is no other alternative appropriate method. In this paper we will use inductive method. We need many special cases to prove so that we will use the computer algebra system to assist us. We use symbolic computation (using CAS and Fortran) to compute with rational number. First we prove geometric theorems about triangles and conic sections. In those cases we only need linear algebra so that using rational numbers will eliminate rounding error. Every theorem can be proved with special cases using rational numbers. Then we move on to non-euclidean geometry which is one of the most important topics in general relativity. We need calculus so that we use real number with high precision to prove special cases. The main goal of this research is to make mathematics closer to physics and by doing that we get a deeper understanding of general relativity.

Keywords-computation; inductive; geometry; relativity

I. INTRODUCTION

Theoretical Physics. Physics can be roughly divided into theoretical and experimental physics. Theoretical physics is built up by statements (or propositions in formal term) which are usually written as mathematical equations. Each proposition can (in principle) be derived from other propositions, except for those topmost propositions which are called postulates. In principle we can derive (using logic and mathematics) all propositions of a theory from a set of postulates. On the long road from the postulates to a targeted proposition there are propositions that are important to be noted. Those important propositions are called theorems (usually in mathematics) or laws (in physics). They are worthy of note because of the frequent use in the derivations in the theory.

To make the discussion simpler let us take an example of a fundamental theory, namely Special Relativity. There are two famous postulates of Special Relativity. One of the postulate is the statement that the speed of light (or photon) in vacuum is an invariant in all inertial reference frames. From the set of postulates we can derive (in principle) whatever proposition in Relativity Theory. As mentioned above there are important propositions which are called theorems or laws. One of the important laws is the law which

is usually called the Lorentz Transformation. This theorem or law can be used to derive other laws, for example the famous laws which are usually called Time Dilation and Length Contraction.

This process of derivation from the top (postulates) to the bottom (theorems, laws, special propositions) is also called deductive proof or proving deductively. This style of proof and the resulting theoretical structure is following the paradigmatic Euclidean Geometry. The strength of this way of proving is that if we really are sure of the truth of the postulates then all the derived propositions can be guaranteed to be true also. All mathematical theories can be included into this category. No wonder that if physics is designed to have theories in these form then physics will be more and more mathematical in the future.

Experimental Physics. We can make a metaphor of physics like a coin. One face of the coin is the theoretical physics which is like mathematics. There is another face of the coin which makes the difference of physics to mathematics, namely experimental physics. In theoretical physics we represent a physical quantity by a mathematical symbol and we can manipulate the quantities using their symbols without (practically) bothering their actual values.

In experimental physics we construct an experimental setup in which each physical quantity has a certain value which is ascertained by measurement. So in experimental physics we have special cases at the outset; we are at the bottom. Let us take a paradigmatically simple example, namely the setup to discover the law of the pendulum. In the first place we have to ascertain the length of the cord, the mass of the pendulum and the height from the ground. Then we measure the period of the pendulum. After experimenting with so many number of cases (infinite number in principle) we can take the conclusion that there is no influence from the mass and the length is proportional to the square of the period.

Then we boldly write the equation relating the relevant quantities. The equation has the intention of stating the validity for all cases whereas the actual experimental work has only verified the equation for finite number of cases. This process of going from special cases into general cases ending up in an equation (or a law, theorem) is called inductive proof.

Here we come into an important insight. In physics the truth of a law or theorem is proved by deductive and inductive methods. Both methods act as complementaries, we cannot choose just one method and leave the other one; both have to be considered proportionately.

The aim of this paper. In this paper we will prove geometric theorems by inductive method. Instead of experimental setup and measurement we will use computer and its number crunching capabilities. In a sense this can be called the computer simulation of experiment to explore the nature of geometry. We will use CAS but also the venerable Fortran 77 [1] to show that symbolic computation can have wider meaning including computation using rational numbers and Fortran can have something to contribute in this case.

II. PLANE ANALYTIC GEOMETRY

Triangle. Let us take a specific case of proving the existence of the centroid of a triangle. A triangle consists of three points and three sides (straight lines). With each point is associated two sides adjacent and one side in front of it. The centroid is the point of the intersection of the three line connecting a point with the midpoint of the side in front of that point. We have to prove that the three lines intersect at one point.

In accordance with the inductive method we consider a triangle with specific points. Let us call the points as A, B and C with the following Cartesian coordinates : A(1,1), B(4,5) and C(3,15). We determine the midpoints of the sides as D (for the line AB), E(for the line BC) and F(for the line CA). Then we can determine the equations of the lines AE, BF and CD and finally we determine the three points of intersection of two lines out of the three lines. We can see at the result of the computation that the three points are in fact coincident. For this specific case we will find the coordinate of the line as T(8/3,7).

We can repeat the computation with other coordinates of the points A, B and C with the number of values as many as we wish. We can even tell the computer to repeat the case a certain large number of times, randomly or systematically. This is the essence of inductive proof. We do not find any anomaly but only for finite (although very very large) number of cases but never reach the infinite number of cases dictated by the truth of the equation.

Computational error. There are two kinds of computational errors, namely truncation and rounding error. In this case we do not need calculus (only linear algebra) so that there is no truncation error. The rounding error can be eliminated by using rational number rather than floating point real number. For each specific case we get exact result. For example when comparing the coordinates of the three intersection points, what we get are the exact rational numbers 8/3 and 7/1 so that we have no doubt that those three points have really the same coordinate values.

CAS and Fortran. Computation using rational number is the default of symbolic computation although we know that the main characteristic of symbolic computation is the manipulation of mathematical expression. It is the reason

that rational number computation can also be carried out using a programming language specifically designed to do numerical computation such as Fortran.

We have constructed the programs both in CAS Derive and Fortran. Because Fortran is not specifically designed for rational number computation, the Fortran program needs longer lines of code.

```

subroutine jum(a,b,c)
integer a(2),b(2),c(2)
c(1)=a(1)*b(2)+a(2)*b(1)
c(2)=a(2)*b(2)
ngc=igcd(c(1),c(2))
c(1)=c(1)/ngc
c(2)=c(2)/ngc
end
subroutine kur(a,b,c)
integer a(2),b(2),c(2)
c(1)=a(1)*b(2)-a(2)*b(1)
c(2)=a(2)*b(2)
ngc=igcd(c(1),c(2))
c(1)=c(1)/ngc
c(2)=c(2)/ngc
end

```

The program in CAS Derive is only about 25 lines long. Although shorter in program listing it is lower in speed, so that if we want to make a large number of automatic repetition, Fortran will make a better choice.

```

garis(a, b, p, q, r) :=
  Prog
  " Defining linear equation      px+qy+r=0"
#1:
  p := a42 - b42
  q := (-1) * (a41 - b41)
  r := (a41 - b41) * a42 + (-1) * (a42 - b42) * a41
  [p, q, r]

titpot(g1, g2, ma, mx, my, deter, x, y) :=
  Prog
  " Calculating the intersection  "
#2:
  ma := [g141, g142, g241, g242]
  mx := [-g143, g142, -g243, g242]
  my := [g141, -g143, g241, -g243]
  deter := DET(ma)
  x := DET(mx)/deter
  y := DET(my)/deter
  [x, y]

```

Conic Sections. To use rational numbers we have to be able to choose rational values of the coordinates of the points on the conic section (for instance an ellipsis). This is possible through the use of Pythagorean numbers which can be readily constructed. For example we can prove inductively that a ray from a focus of an ellipsis will be reflected by the point on the ellipsis straight to the other focus. For other theorems so long as we can bypass the explicit computations of distance and angle then we can get exact result for each specific case and the inductive proof will become very convincing.

III. SPHERICAL SURFACE AS AN EXAMPLE OF CURVED SPACE

Spherical Triangle. The next step is the exploration of the geometry of curved space. This study will be very important to the understanding of general relativity because of its dependence on curved space with the accompanying non-euclidean geometry [2,3,4].

To make the discussion more transparent and to avoid the unnecessary complication we will consider spherical surface

as an example of curved (two) dimensional space. To make the discussion continuous with the previous one we will discuss spherical triangle. Let us look at a specific case of three points A, B and C on spherical surface. The sides of the triangle connecting those three points are the arcs of the great circle passing through the points. Note that a great circle passing two points is a circle lying on a plane which passes through the two specific points and the center of the sphere.

The angle subtended by the two sides on a point of a triangle, for instance the angle A, is exactly the same as the angle between the normal vectors of the two planes containing the great circles AB and AC. The normal vector of the plane containing circle AB can be determined from the three points A, B and the center of the sphere. Up to this step we can use rational number, but we cannot use rational numbers to represent the angle between the two normal vectors. We have to use real number.

After getting the values of the three angles A, B and C we will see that the sum is greater than 180 degrees. We can repeat as many as we wish and we will come to the inductive proof that the sum of the angles in spherical triangle is more than 180 degrees. Note the significant difference that here we cannot avoid using floating point real number with the consequence of the emergence of rounding error. Using CAS programming language can help making the rounding error very small, but with the price of the lower execution speed. Here again we have a trade off between precision and speed in the comparison of CAS Derive (as the representative of CAS programming language) and Fortran as the representative of numerical programming language.

IV. DISCUSSION AND CONCLUSION

The study of the geometry of curved space apparently can shed some light about general relativity. We have seen previously that if we can avoid explicit computation of angle and distance then we can use rational number and thus avoiding the rounding error. This is possible if we can restrict the mathematical machinery to linear algebra. That situation can be realized in plain space or Euclidean space.

In curved space we apparently have to use real number. That make the important insight that general relativity need the continuum of real numbers [5]. Space cannot be discrete for general relativity. We sincerely feel this is the implication to the study of general relativity. This feeling will be valuable to share in the introductory teaching of general relativity.

As the official conclusion we can say that we have constructed the programs in CAS Derive and Fortran that can inductively prove the theorems of plane analytic geometry and also inductively prove (in a weaker sense) the theorems of spherical geometry. There is an unavoidable trade off between precision and speed in the comparison of those two programming languages.

REFERENCES

- [1] M. Kupferschmid, Classical Fortran, Marcel Dekker, 2002.
- [2] R. N. Henriksen, Practical Relativity, Wiley, 2011.
- [3] O. Fritzsche, The Curvature of Spacetime, Columbia University Press, 1996.
- [4] O. Gron and A. Naess, Einstein's Theory, Springer, 2011
- [5] E.v. Groesen and J. Molenaar, Continuum Modelling in The Physical Sciences, SIAM, 2007.