Trends in Mathematics
Information Center for Mathematical Sciences
Volume 2, December 1999, Pages 199–207

ON ALGEBRAIC ANALYSIS I

SUN T. SOH

ABSTRACT. In this article, one considers two main aspects of mathematics: axiomatic and computational. After discussing the concept of Algebraic Analysis, one examines a promising role of computers as a new tool in doing mathematics with some examples of computer-based computations.

1. TOOLS IN MATHEMATICS

The whole body of Mathematics consists of many different kinds of knowledge or results and procedures which produce such results, most of which are verified through the process of the so-called "proof". Those proved and somewhat useful or important are usually called **theorems**. Unproven ones are formally called **conjectures**. The place for most mathematicians should be between these two words, because we are born to establish reasonable conjectures and then prove them. When one tries to prove something, something itself must be first recognized as a reasonable mathematical object; discovering such an object could be quite personal.

In proving reasonable conjectures one generally uses two tools, a special type of logics, called **mathematical logic**, and **computation,** definitely guided by his own mathematical intuitions or experiences. Since they are applied as tools in the process of the so-called **proof**, they cannot be called mathematics itself. But, the kinds of the method employed certainly puts some unavoidable limitations on the character of the consequences. These two tools, nevertheless, do not hurt each other. In contrast, they help each other, and these two are the most powerful tools we, mathematicians, have. Depending on the degree of putting more emphasis on any of these two, one usually has two basic tendencies in doing mathematics, axiomatic or computational.

1.1. **Axiomatic Nature.** Although it is very difficult to figure out what constitutes mathematics, one may describe its axiomatic nature as follows. Once a reasonable mathematical conjecture is recognized, one accepts, without questioning their existence or validity, a finite number of undefined terminologies and that of postulates (i.e., axioms or principles) which describe some relations between the undefined terminologies, and one applies a finite number of steps of mathematical logic directly to them or to some of their consequences until one finally reaches the desired positive or negative conclusions. As a demonstrative science started even

Received by the editors July 24, 1999.
1991 *Mathematics Subject Classification.* 01-08, 01A67.
Key words and phrases. Key words and phrases: symbolic computation, approximation.
This work was supported by Korea Research Foundation (1998-015-D00011).

before Greek period, mathematics has shown this axiomatic nature for a long time. One can find such an evidence in, for instance, Euclid's Elements [5] and other works of Greek mathematicians and philosophers. In fact, one can even trace back to their works to see that most of the formal terminologies used in mathematics, something like definitions, propositions, corollaries, etc., were already used in his Elements and other Greek literatures. The meaning and usage of these terminologies and other metamathematical concerns are also well explained in the introduction of A. Whitehead and B. Russell's famous work, *Principia Mathematica* [20], which was originally published in 1910.

1.2. **Computational Nature.** When computations in general are concerned, we know that there are basically two different kinds, namely, exact computation and approximation. **Approximation** is done through the limiting process mostly in terms of inequalities, where as **exact computation or symbolic computation** is done through the so-called four binary operations: addition, subtraction, multiplication and acceptable division. These operations are applied not only to numbers but also to a mixture of numbers and **symbols**.

Computations are very important to us at least for the following three reasons. First, they usually show up in the middle of the process called proof. This is almost unavoidable to every mathematician. Secondly, a decent amount of computational experiments quite often reveals to us what's going on in a very clean manner, while simple-minded 'what if ...' ways of logical questioning cannot. Thirdly, we, mathematicians, can communicate with other branch of sciences very efficiently and successfully, through computational results and their usefulness. Considering that mathematics has so far grown up to this much within the accumulation of our civilization, this point of view is also very important, because isolation only hurts oneself till one's starvation.

Up to the end of the19-th century or the beginning of the 20-th century, some important mathematical works discovered and proved were based on individual mathematician's heavy hand-computational experiments. But, as was the case in Invariant Theory of the 19-th century, mathematicians were practically forced out to give up heavy computations when the number of unknowns and the size of numbers are substantially increased so that hand-calculation obviously becomes terrible. Any attempt based on computational experiments may be called a **computational approach**.

2. Strong Emphasis on Axiomatic Approach in the 20-th Century

On the other hand, the revival of the so-called **axiomatic approach** after the Greek period (namely, Euclid's Elements) is clearly evident in the Lecture Notes of E. Artin and E. Noether, written up by B.L. van der Waerden in 1930's [19]. This approach, especially well-supported by precise notations adapted from the subject, Set Theory, was so good in pursuing a particular-minded concept or property, that it rapidly pushed out the old tradition of computational approaches, which was quite common up to the first quarter of 20-th century, from the mathematical world. And some eminent mathematicians, like Banach in Analysis or later E. Spanier in Algebraic Topology [18], tried to figure out and establish sufficiently small number of axioms but still good enough to explain successfully the rest of the theory they know. In order to fullfil such tasks, they examined many computed examples of different kinds and pulled out some common underlying axioms which replace too much involved terrible computations. The success of this attempt is very much evident in almost every part of current mathematics in both research articles and textbooks. It is also around this period that most mathematicians officially but still reluctantly began to acknowledge usefulness of the non-traditional mathematical axiom, i.e., the axiom of choice, or its equivalence, the Zorn's Lemma.

The point of views expressed in this article could be quite personal until it gets more acceptance from general audiences. But, the readers will not be hurt by the author's point of views on mathematics expressed in this article but it will supply a fresh taste or sense to our main meal, mathematics.

2.1. **Danger of Too Much Emphasis on Axiomatic Approach.** One should note that those mathematicians, who introduced an appropriate set of axioms for the knowledge they already know, were also very good at computation, since they were trained as, called from now on, the "first" generation mathematicians, by their teaches most of whom were under the strong influence of computational approach of the 19-th century. But, the "second" generation mathematicians, i.e., those who were mathematically trained by the first generation mathematicians, tended to start their mathematics not from heavy computational experiments but from a small amount of axioms, and they were more and more away from computational experiences. The lack of computational experiences is more significant in the "third" generation mathematicians, i.e., those trained by the second generation mathematicians.

Nevertheless, the strength of axiomatic approach had been fully demonstrated by the monumental works of a single mathematician, A. Grothendieck, on the Foundation of Algebraic Geometry, [6], FGA [7], EGA's [8] and [9], and SGA's [10] during 1950's up until the beginning of 1970's. His works in turn simply recall the works of a group of mathematicians, called Bourbaki [3] and [2], as elementary results. Anyone who are very much interested in the scope of axiomatic approach in mathematics explored during the 20-th century are strongly encouraged to study and digest the above mentioned Grothendieck's works, of which the basic concepts and results are described in terms of categorical languages, called the **language of schemes**.

In particular, the tendency of the results obtained under a strong influence of axiomatic approach is very much conceptual, and hence becomes more and more **soft** and hence fragile, as time goes. In this case, a good-minded mathematicians should worry about whether his and/or her results will still survive after, for instance, $1,000$ years from now. An example is Euclid's method of computing the g.c.d. of two numbers. His method is still known to us because it is about computation of certain mathematical objects in an efficient manner. Moreover, lack of computability only speeds up the isolation of Mathematics from other branches of Sciences. This is in no way acceptable to us.

For an elegant axiomatic treatment but still with a strong emphasis on computational point of views, of Algebraic Geometry, one may refer to Prof. George R. Kempf's two books [11] and [12].

3. A New Computational Tool: Computers

3.1. **Personal Computers.** Starting at around the beginning of the fourth quarter of the 20-th century, computer industry has grown up with an unexpected rate of success; this nowadays allows us to compute very big numbers using a Personal

Computer or even a Notebook Computer. As a tool for actual computation, computers are already playing a significant role and such a role will surely change the mathematical world of the 21-st century. Here are some examples.

Example 1. *Here, $p(n)$ denotes, by definition, the number of ways of writing a given natural number n as a sum of others (called components) ignoring the order among the components. One starts with a very famous theorem:*

Theorem 1. *(Rademacher [1]) Given a natural number n, one has*

$$p(n) = \frac{1}{\pi\sqrt{2}} \sum_{k=1}^{\infty} A_k(n)\sqrt{k} \frac{d}{dn} \frac{\sinh((\pi/k)\sqrt{(2/3)(n-\frac{1}{24})})}{\sqrt{n-\frac{1}{24}}}, \tag{3.1}$$

where

$$A_k(n) = \sum_{\substack{0 \le h < k \\ (h,k)=1}} \varpi_{h,k} \exp(-2\pi i n h/k),$$

where

$$\varpi_{h,k} = \exp(\pi i s(h,k)),$$

is a $24k$-th root of unity and where

$$s(h,k) = \sum_{\mu=1}^{k-1} \left(\frac{\mu}{k} - \left[\frac{\mu}{k}\right] - \frac{1}{2}\right)\left(\frac{h\mu}{k} - \left[\frac{h\mu}{k}\right] - \frac{1}{2}\right)$$

is the so-called Dedekind sum. ∎

It turned out that the value of $p(20,000,000)$, when computed it using this theorem, requires more than $4,900$ number of decimal digits (cf. [15]). Thus the final result is really a super-ultra big number, which is one point. Moreover, another point is that it only took about 17 Minutes with already-done pre-calculations of the so-called Dedekind sums, under the same hardware and software limitations as in the next example. These two points clearly show us the computing capability of current PC's.

Remark 1. *One should note that computation of such a super-ultra number eventually raises a serious concern whether the computed result is absolutely correct. This is certainly a new kind of headaches to mathematicians.*

On the other hand, the author recently discovered an exact algebraic formula for a graded partition function $p_{\mathbf{M}}(n)$ over a multiset $\mathbf{M}$ and an associated quasi-recursive formula for $p_{\mathbf{M}}(n)$, both of which never *require approximation at all (cf. [16]). One should note that it is neither quite possible to obtain an analytically exact formula for $p_{\mathbf{M}}(n)$ nor to obtain a recursive formula for it, something like either Rademacher's formula or the well-known Euler's recursive formula for $p(n)$, unless there is a significant symmetry among the elements of $\mathbf{M}$. But, one version of the author's formulas computes all the values of $p_{\mathbf{M}}(i)$ for $1 \le i \le n$ with time efficiency of $\mathrm{O}(n^2)$ for a general multiset $\mathbf{M}$ (cf. [17]).*

Example 2. *In his recent computational experiments, the author has figured out an algorithm for computing the* complete *symmetric polynomials*

$$h_m(x_1, ..., x_n) = \sum_{i_1 \leq i_2 \leq \cdots \leq i_m} x_{i_1} x_{i_2} \cdots x_{i_m}, \quad 1 \leq m \leq n,$$

in the polynomial ring $k[x_1, x_2, ..., x_n]$ *in* n *variables over a field* k*, with time efficiency of* $\mathrm{O}(2^n)$ *from his works on the divisorial complexes. Since symbols are involved, exponential time efficiency in such computation is unavoidable. Note that they are different from the so-called elementary symmetric polynomials*

$$\sigma_m(x_1, ..., x_n) = \sum_{i_1 < i_2 < \cdots < i_m} x_{i_1} x_{i_2} \cdots x_{i_m}, \quad 1 \leq m \leq n.$$

An efficient computation of $h_i(x_k, ..., x_n)$*'s for various* $1 \leq i, k \leq n$ *for a given* n*, is essential since one has a well known result, in conjunction with Theorem 3 below, that*

Theorem 2. *Fix a lex order on* $k[x_1, ..., x_n]$ *with* $x_1 > \cdots > x_n > y_1 > \cdots > y_n$*. Then the polynomials*

$$g_k = \sum_{i=0}^{k} (-1)^k h_{k-i}(x_k, ..., x_n) y_i, \quad 1 \leq k \leq n,$$

where $y_0 = 1$ *by definition, form the unique reduced Groebner basis for the ideal* $(\sigma_1 - y_1, \sigma_2 - y_2, ..., \sigma_n - y_n)$ *with respect to such a monomial order.*

Theorem 3. *(*[4]*) In the ring* $k[x_1, x_2, ..., x_n, y_1, y_2, ..., y_n]$*, fix a monomial order where any monomial involving one of* $x_1, ..., x_n$ *is greater than all monomials in* $k[y_1, y_2, ..., y_n]$*. Let* G *be a Groebner basis of the ideal in* $k[x_1, ..., x_n, y_1, ..., y_n]$ *generated by* $\sigma_1 - y_1, \sigma_2 - y_2, ..., \sigma_n - y_n$*. Given a polynomial* $f \in k[x_1, x_2, ..., x_n]$*, let* $g = \bar{f}^G$ *be the remainder of* f *on division by* G*. Then:*

1. f *is symmetric if and only if* $g \in k[y_1, y_2, ..., y_n]$*;*
2. *if* f *is symmetric, then* $f = g(\sigma_1, \sigma_2, ..., \sigma_n)$ *is the unique expression of* f *as a polynomial in the elementary symmetric polynomials* $\sigma_1, \sigma_2, ..., \sigma_n$*.*

*For definitions and terminologies, one may refer to author's next article, "*On Algebraic Analysis II.*"*

The author does not know how many of us actually computed all the $h_i(x_k, ..., x_n)$*'s for various* $1 \leq i, k \leq n$ *, for a given* $n = 2, 3, 4, ...$*, but when he did it with his own algorithm, he was able to compute them at least up to* $n = 20$*. For smaller* n*'s it is not a problem at all, but when* n *gets bigger the situation is quite different. For instance, when* $n = 20,$ *it took about* 37 *Minutes to finish the whole computation and it required about* 65.6 *Mbytes of Hard Disk space to store them. This computation was done with my PC running at* 200 *MHz and the amount of main memory assigned to it was* 128 *Mbytes. The Computer Algebra System, REDUCE under Microsoft Windows* 3.1*, produced such a result. Of course, no Hard Disk swap was allowed.*

The readers are strongly encouraged to compute them using any methods to see if one can accomplish the job when $n = 20$ *under the same hardware and software limitations as the author had in the above.*

One should note that the number 20 *is a very small number, but computation of various complete symmetric functions* $h_i(x_k, ..., x_{20})$ *in various* $k[x_k, ..., x_{20}]$ *for*

$1 \leq i, k \leq 20$ *is not that easy. Namely, it is quite often the case that something computable does not really mean that it can be easily computed even with help of computers. To my regret, most mathematicians do not understand what this really means.*

3.2. **Computer Algebra Systems.** Of course, having just a very fast computing machine of which hardware is very reliable is not enough. One also needs a reliable software which guarantees several facilities. Among them, the most urgent requirements are, for instance, ability to handle numbers of any sizes and arbitrary precision in representing fractions, each of which absolute reliability is definitely required in any computation with computers.

Example 3. *The following Table shows the actual computation of the value of the classical partition function* $p(n)$, *for* $n = 200$, *using the Rademacher's formula in Eq (3.1). Here, by adding the first* 8 ***non-zero*** *terms of fractional real numbers, one gets a fractional real number,* 3972999029388.001346332, *which is very close to the true integer value* $p(200) = 3972999029388$.

$t(1)$	=	$+3.97299899318589575787 2E+12$
$t(2)$	=	$+36282.9784854411847056$
$t(3)$	=	-87.58391287399
$t(4)$	=	$+5.146534584$
$t(5)$	=	$+1.424217569$
$t(6)$	=	$+0.07102143214$
$t(7)$	=	0
$t(8)$	=	$+0.04357655366$
$t(9)$	=	$+0.0256657539$
$[t(10)$	=	$+0.008603641042]$
Up to 9-th Partial Sum	=	$+3972999029388.001346332$

When it is about computation of an exact integer value in this way of approximation, there should not be any approximation-related errors in all of the intermediate steps. Do not expect that an error-free computation of such a kind is quite easy (cf. [15]*), unless one uses very reliable Computer Algebra Systems.*

Any software package lacking these two capabilities should not be our choice for a tool for computational experiments. In addition to them, it is also required to have an ability to solve a given system of polynomial equations, i.e.,e, non-linear equations over, say, the field of complex numbers. This means that it must have an ability to handle symbols as well as numbers. The theory for such computation was not fully understood until Buchberger discovered his famous algorithm for Groebner bases in 1965. Any reliable software package at least with these three facilities could be called a **Computer Algebra System**.

There are two types of Computer Algebra Systems, commercially available or not. Some of the commercially available ones are, MAPLE, MACSYMA, REDUCE, etc. Non-commercially available ones are, MACAULAY II, CoCoA, etc. There should be more than these.

3.3. **Behind Computational Experiments.** One first notes that there are two kinds of computations one can do with computers, **exact computation** and **approximation**. Exact computation only needs addition, subtraction, multiplication, and allowed division depending on given conditions, on a given mixture of known symbols, unknown symbols, and numbers. In contrast to this, approximation allows us to apply non-exact division and limiting process, as well as addition, subtraction, multiplication and division on given numbers only. Thus, exact computation allows us to manipulate not only numbers but also **symbols**, whereas approximation only allows us to manipulate numbers. This tells us why exact computation is also called **Algebraic Analysis** or **Symbolic Analysis**, while approximation is called **Numeric Analysis**. Examples of Numeric Analysis are numerous. For instance, FEM (Finite Element Method) is just one of them. Also, see Example 3 above.

These two approaches in computational methods do not hurt each other. In fact, they help each other. The only general trouble is that Algebraic Analysis does require some amount of elementary Algebra, mostly about polynomial rings in more than one unknowns, as well as some amount of Modern Algebraic Geometry, developed by late Prof. Oscar Zariski [21] and J. P. Serre [14], and a few others (For beginners, this amount of accounts on the subject may be good enough).

A proper sequence of computational approach is probably first

1. to apply appropriate exact methods in Algebraic Analysis until all available ones are exhausted, and then, as a final step,
2. to apply appropriate approximation methods in Numeric Analysis.

The subject of Numeric Analysis has been well known to us, for instance, after Sir Isaac Newton, at least. But, Algebraic Analysis is very new to us. The most important theoretical background for recent revival of computational approach is the Buchberger's astonishing discovery of the algorithm for Groebner bases for polynomial ideals in more than one indeterminates. He figured out the method while he was working on his Ph. D. thesis; was first recognized by mathematicians oriented to computer science somewhat before or around 1975. I myself learned this remarkable algorithm in mid 1980's.

3.4. **Danger of Too Much Emphasis on Computational Approach.** If one puts too much stress on computational approach, he will soon finds that he does not know why the method works and eventually he will be got lost, because the arithmetic operations one applies never tell us what's going on, unless one knows *in advance* what's really going on, but which is obviously impossible most of the time especially when one works on some mathematical phenomena not yet well-understood to him. This means that one really needs a well-developed theory for exact computation or Algebraic Analysis.

There were no general theories for exact computation with computers until Buchberger discovered a very important algorithm. The mixture of his algorithm with the division algorithm produces very important algorithms for polynomial ideals in more than one variables. It is very urgent to understand this theory for hard but honest and diligent computations. In my next article, "*On Algebraic Analysis II*", the most elementary results of the Theory of Algebraic Analysis, that are sufficiently good enough for the beginners, are summarized in a systematic manner. The readers of this article should not underestimate the strength and usefulness of this theory.

4. Expected Trends of the 21-st Century Mathematics

4.1. **Computer Ages.** It is well known that there were two types of computers, analogue and digital, in the past. But, nowadays computers usually mean **digital computers** or their variants for better performance, since performance of digital computers (for instance, signal to noise ratio) had already been proven to be a lot better than analogue types. It is also well known that the theoretical creation of digital computers was done by a famous mathematical logician, von Neumann, who invented Class Theory to answer, for instance, the question like whether the set of all sets is a set or not. After theoretically creating digital computers, he immediately developed a good deal of approximation theory for computing the values of non-polynomial functions with his theoretical machines.

Although Electric Engineering, Computer Engineering, and Computer Science are respectively needed to establish hardwares, architectures, and operating systems and softwares, the final combination of them is a system, called, computer, and the main reason for developing such a system is to compute or calculate something in any possible senses. The point we, mathematicians, should not miss is that **computation belongs to mathematics**. Thanks to computers, paperless computation of huge numbers, for instance, is already available to those who know how to use some of the commercially or non-commercially available Computer Algebra Systems, so that the scope of Algebra and Algebraic Geometry in both pure and applied directions are already very much expanded.

4.2. **Computers Are Created for Computation Related Human Activities.** Since computation solely belongs to mathematics, it is mathematician's due obligation to examine and develope the needed mathematical algorithms in order to obtain from the machine the desired results with absolute correctness. In addition to this, our second obligation is to make it as efficient as possible. If it takes too much computation time, then such a computational approach will never buy other people's approval at all. To my regret, these two can't be easily accomplished by those who were not trained as mathematicians.

Unlike any other centuries, existence of so many mathematical objects have been proved during the 20-th century, mostly thanks to active application of axiomatic methods influenced by Set Theory or Category Theory. These results are all inputs for further investigation that which of them are computationally constructible applying the methods of Algebraic Analysis.

I should mention one dangerous aspect of the Zorn's Lemma or its equivalent axioms. If one used the Zorn's Lemma in some place of his proof of the existence of certain mathematical object, then his proof is computationally useless in the sense that it never really tells us how to construct such an object only using the four arithmetic operations and the limiting process. This is really a disappointing aspect of the Lemma, but is, as long as **algorithmic construction** is concerned, still important enough to keep in mind in order for us not using it in one's proof as possible as one could.

Computers have already become a convenient and very capable tool for computations in any part of mathematics. We do not necessarily need Supercomputers, Servers, or even Workstations for computational experiments. We only needs Personal Computers for most computational experiments. Since Personal Computers are nowadays available to everyone and since they are already too fast and too capable of, most computational experiments could be done with them without spending too much time and money. Moreover, the hardware-related performance of these machines are nowadays very much reliable. The author strongly believes that such a very reliable computational environment, already available to everybody, sooner or later change the whole atmosphere of mathematics of the 21-st century. For instance, the so-called **internet computing** will soon become very common.

One of the main ingredients of mathematics is perhaps its **exactness** in the process of both **proof** and **computation.** The point is that both of them require quite different types of exactness. This may be also the main reason why no other branch of Sciences even tries to compete with mathematics but only attempts to rely on the results produced by mathematicians; consequently mathematics has its own everlasting life. We should not give up any of these two powerful tools, proof and computation, to other branches of Sciences. On the other hand, since we have been silently forced to ignore the significance of computational aspects for at least a century, mostly because of our inability in exactly hand-computing mixtures of big numbers and many parameters and also partly because of strong influences of axiomatic approach of the 20-th century, isn't it a time for us to revive computation related mathematical activities with a new tool, computers, to balance ourselves between them?

References

[1] George E. Andrews, *The Theory of Partitions,* Reading, 2nd printing,1981, Addison-Wesley
[2] N. Bourbaki, *Algebra,* Part I and II (in English), 1970, 1981, Springer-Verlag
[3] ——————, *Commutative Algebra* (in English), 1985, Springer-Verlag
[4] D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms, An Introduction to Computational Algebraic Geometry and Commutative Algebra,* 1992, Springer-Verlag
[5] Euclid, *The Elements,* Vol. 1 - 3 (English translation), 2nd Ed., 1956, Dover
[6] A. Grothendieck, *Algébre Homologique,* Secrétariat mathématique, 11 rue Pierre Curie, PARIS 5e, 1958
[7] ——————, *Fondements de la Géométrie Algébrique* (Extracted from Bourbaki Seminar 1957-1962), Secrétariat mathématique, 11 rue Pierre Curie, PARIS 5e, 1962
[8] A. Grothendieck and J.A. Dieudonné, *Eléments de Géométrie Algébrique,* Vol. I-IV, 1960-1967, IHES
[9] ——————, *Eléments de Géométrie Algébrique I,* 1970, Springer-Verlag
[10] A. Grothendieck, *Séminaire de Géométrie Algébrique,* Vol. 1-7, 1960-1962, IHES
[11] George R. Kempf, *Algebraic Varieties*, London Math. Soc. LNS 172, 1993, Cambridge Univ. Press
[12] ——————, *Algebraic Structures,* 1995, Friedrich Vieweg & Sohn,
[13] D. Mumford, *Algebraic Geometry I, Complex Projective Varieties,* 1976, Springer-Verlag
[14] J. P. Serre, *Faisceaux Algébriques Cohérents*, Annals of Mathematics, Vol 61, No. 2, March, 1955
[15] Sun T. Soh, *A Reliable Implementation of the Formula of Rademacher for the Partition Function* $p(n)$, submitted
[16] ——————, *A Diophantine Construction of an Exact Algebraic Formula for Graded Partition Functions*, J. Korean Math. Soc., Vol 36 (1999), No. 2, pp. 267-298
[17] ——————, *Efficient (Quasi-) Recursive Formulas for Graded Partition Functions,* preprint
[18] E.H. Spanier, *Algebraic Topology,* 1966, Springer-Verlag
[19] B.L. van der Waerden, *Algebra,* Vol. 1 and 2, 1970, Frederick Ungar Pub.
[20] A.N. Whitehead and B. Russell, *Principia Mathematica*, 1910, Cambridge Univ. Press.
[21] Oscar Zariski, *Collected Papers,* Vol. 1 - 4., 1972, MIT Press
[22] Oscar Zariski and P. Samuel, *Commutative Algebra,* Vol.1 and 2, 1958, Springer-Verlag

Current address: Department of Mathematics, Myong Ji University, Yong In, Rep. of Korea
E-mail address: sunsoh@wh.myongji.ac.kr