Beyond Calculus: Modernizing System Modeling and Computational Methods

Pavel Loskot
ZJU-UIUC Institute
Haining, China
e-mail: pavelloskot@intl.zju.edu.cn

***Abstract*—Mathematical modeling has been used for centuries to create understanding of natural phenomena, and recently, to enable constructing man-made systems. Computational models in engineering are mostly based on calculus and linear algebra, and they are now well-established. However, the rapidly rising complexity of modern intelligent and autonomous systems calls for rethinking the present modeling strategies to allow beyond calculus modeling with different levels of abstractions, semantics, and information granularity. This vision paper offers a non-mathematical outline of selected advanced mathematical topics, which can be adopted not only to enhance modeling methods in engineering, but also to modernize engineering curricula by bridging the gap between engineering and mathematics. The key idea is to allow algebraic manipulations of mathematical objects, and assigning numerical values to these objects, so they can be used as natural data structures within mathematical models.**

Keywords-abstract algebra; algebraic geometry; algebraic topology; calculus; curriculum; mathematical object; modeling.

I. INTRODUCTION

The interest into studying scientific problems evolved since the age of Enlightenment. The first attempts were focused on creating simple physical models of measurable quantities. This then evolved into investigating the problems of disorganized complexity assuming more complex systems and phenomena, but with well-defined behaviors. The scientific progress following the World War 2 (WW2) changed the focus on problems of organized complexity that embrace all factors influencing whole systems.

The scientific methods were initially mainly experimental. The strong interest into theoretical sciences including mathematical modeling and analysis emerged already during the WW2. The advent of microprocessors leading to affordable personal computers and software tools in the 1990's gave rise to a widespread adoption of computer simulations to design and analyze complex engineering systems. The main advantage of computer simulations is that they are universal, require fewer assumptions, and can be used to evaluate complex mathematical models. The disadvantage is that they can be rather slow when they are used iteratively.

The most recent breakthrough in scientific methods emerged in the mid 2000's when not only the computing capabilities started to exponentially improve, but also the vast volumes of data started to be collected. The new era of big data and large computing models enabled many new applications including knowledge mining, and analyzing, predicting and controlling highly complex systems. This encouraged developments of machine learning models, which may be considered to be calibrated computer simulations having the real-time observations as their inputs. The universality of large-scale (deep) machine learning models made them very popular, and widely used in many applications across different scientific and engineering domains. However, the main drawbacks of large-scale machine learning models are their excessive consumption of resources, slow training, and difficulty, or even impossibility of validating the correctness of their outputs. The deep learning models are inherently probabilistic, and their outputs are only valid with a certain probability within a given context.

The current developments of deep (machine) learning models, and more generally, of intelligent systems are purely experimental. There is a substantial cost incurred for their training and testing, and many trials-and-errors are required for selecting their hyper-parameters, and fine-tuning. Furthermore, these models are neither explainable, nor is their optimality guaranteed. Hence, there is a great need to develop design guidelines for deep learning models, which can serve the same purpose as the known laws for designing electrical circuits, and electromagnetic systems, to name some examples.

The persisting lack of design guidelines may be caused by the excessive and constantly rising complexity of today's mathematical models. These models must be considered across many different scales and hierarchies. In addition to quantitative evaluations, the model semantics can play a crucial role in designing many intelligent and autonomous systems. A viable strategy could be to consider the methods and strategies that were adopted for taming the complexity in software development. At a basic level, these strategies include the model granularization and modularization. More advance strategies can be inspired by object-oriented programming, and include the model encapsulation, inheritance and polymorphism, which allow high-level abstractions and reusability.

The aim of this paper is to explore advanced mathematical structures, which can enrich traditional modeling methods. The traditional methods mostly rely on computations involving basic operations and functions of vectors and matrices [1]. The main advantage of these methods is that they are optimized for software and hardware implementations. The complexity of today's models dictates using more abstract and more flexible modeling strategies that can capture sophisticated relationships and semantic representations of multiple sub-models at different scales and levels of hierarchy.

The rest of this paper is organized as follows. Section II reviews traditional modeling methods that have been used in designing and analyzing engineering systems for many decades. Section III outlines mathematical structures, and related topics, which could be considered for advanced mathematical modeling.

Section IV discusses opportunities and challenges in using using advanced mathematical objects in building models not only in engineering, and mention possible strategies. Finally, Section V concludes the paper.

II. Traditional Methods

Traditional modeling methods in engineering are well-established. They rely on basic mathematical concepts that are normally covered in typical engineering curricula. The corresponding mathematical courses include calculus, linear algebra, and discrete mathematical structures [2]. In later years, the students usually progress to using mathematical models in various engineering applications.

A. Calculus and Differential Equations

Calculus at engineering schools is mainly about continuous functions of real and complex variables. Differential and integral calculus is useful, for example, in computing the areas and volumes of geometric objects, and characterizing scalar and vector fields along or within defined surfaces and lines in two or three dimensions. The solutions of ordinary and partial differential equations of small order can be obtained directly, or by using various transforms. For higher orders, and when computing high-dimensional integrals, numerical methods must be used. These techniques are sufficient for modeling physical and other phenomena that are encountered when designing engineering systems including electromagnetic radiation, electrical circuits, chemical reactions, and others.

B. Linear Algebra

Multiple variables can be arranged in regular shapes as vectors and matrices. These objects exist in vector spaces, which may be endowed with additional properties, such as inner products and other more general metrics. The two-dimensional matrices are important in describing systems of linear equations, as well as (locally) linear transformations. The spectral properties of matrices are related to their eigenvalues and eigenvectors. The matrices can be diagonalized and factored in various ways, which is important in many signal and data processing applications. The standard matrix product is only defined in two dimensions, but it can be extended to higher dimensions using the methods of multi-linear algebra.

C. Discrete Structures

In addition to matrices in different number of dimensions, the basic discrete objects are sets and graphs. These structures are normally encountered in undergraduate courses on data structures and algorithms, and on designing logical circuits. Graphs can represent not only systems of interacting entities, but they also capture the pairwise relationships between such entities. In graph signal processing, the graph edges define linear or non-linear transformations between neighboring variables, which are assigned to graph vertices. Graphs can also represent time-evolution of states in dynamical systems, for example, when assuming recurrences. Social network analysis is mostly concerned with graph invariants, which are scalar metrics representing the graph structure. Graph theory provides many useful properties and methods, for example, how to enumerate and count graphs of certain properties.

D. Signals and Systems

The courses on signals and systems are intended to provide the students with fundamental skills to be able to work with various models in engineering applications. The signals are time-varying variables, and every system must have at least one input or output. The signals can be deterministic or random, and they can be studied not only in time, but also in other domains. The time and frequency domains can be considered to be either continuous, or discrete. The signal properties are usually defined to be scalar invariants. In many applications, it is desirable to derive the properties of output signals given a system and its inputs. For example, the systems can be designed, so that their outputs have desired properties given specific input signals. Alternatively, the task is to define the input signals to obtain the desired outputs given a specific system. The inverse problems infer the latent properties from the input and/or output observations.

The system is said to be observable, if the hidden state or parameter value can be recovered from available observations. The system is said to be controllable, if it can be driven to a desired state from any given state in finite time.

Unlike universal modeling methods that are based on deep learning, the specialized models of signals and systems are numerically much more efficient, interpretable, and provably optimum. There can be a family of models of the same phenomena, which have varying degree of faithfulness. The current trend is to reduce the complexity of universal models by incorporating well-defined specialized models whenever they are available. The model complexity can be also reduced by analytically solving some parts of the model, which leads to efficient semi-numerical modeling strategies.

E. Optimizations

The optimizations are useful in maximizing the performance while minimizing the use of resources. The optimization objective is often constrained by equality and inequality constraints, which restrict the feasible set of possible solutions. There are also problems involving multiple objectives, which can be conflicting, and lead to Pareto trade-offs. Convex optimization problems guarantee the existence of a single global minimum. The problems involving continuous optimization variables can be solved numerically using the methods of gradient descent. Combinatorial optimization problems involve at least one optimization variable that is discrete, The challenge is how to find practical methods for online optimizations, which must update the solution iteratively as the new observations arrive.

In practice, the solutions of primary problems are often obtained as the optimization problems. This is the case, for example, of many parameter estimation methods.

III. ADVANCED METHODS

There are many important results in mathematics, which may likely be useful for designing engineering systems. Even though these results may be considered to be common knowledge in mathematics, they are likely completely unknown in engineering. In this paper, our focus is specifically on exploring advanced mathematical objects, which could be adopted in building enhanced models of engineering systems.

Computational analysis of engineering models generally involves manipulating basic objects, such as numbers and the corresponding arithmetic operations, and functions. The key observation is that, at more abstract level, the binary arithmetic operations can be defined for combining mathematical objects beyond numbers. For example, one can define how to sum and subtract graphs, or multiply and divide high-dimensional tensors. Specifically, a binary operation, $\square$, and a unary operation, $\triangle$, respectively, defined for mathematical objects, O_i, from a certain class of such objects are:

$$O_3 = O_1 \,\square\, O_2\,, \quad O_2 = \triangle\, O_1. \tag{1}$$

It should be emphasized that the operations, $\square$, and, $\triangle$, are neither arbitrary nor ad-hoc. These operations can be precisely defined using fundamental results, for example, in Abstract Algebra, and Category Theory. Likewise, it may be possible to introduce basic arithmetic functions, F_i, such as exp, log, sin, and cos to define the maps:

$$O_2 = F_i(O_1). \tag{2}$$

Furthermore, in addition to object transformations (1) and (2), which create new objects from the existing ones, abstract mathematical objects can be assigned numerical values, i.e., numerical objects. For example, the graph vertices and edges can be assigned scalar variables or parameters. These values are then manipulated using standard arithmetic operations and functions. Note that mathematical objects that hold numerical values are usually referred to as data structures in implementing the data processing algorithms. In general, defining signal and data processing over given mathematical structures containing numerical values is largely an open research problem. The challenge is how to account for the structural constraints imposed by the mathematical structures considered. For example, graph signal processing is concerned with signals that are defined over graphs. A transformation, T_O, of numerical values, $\{v_i\}$, which are constrained by mathematical structure, O, can be formally written as,

$$\{v'_i\} = T_O\left[\{v_i\}\right] \tag{3}$$

where v'_i is the transformed value of the initial value, v_i.

In the sequel, a few specific mathematical objects are outlined, which may be useful in defining the binary and unary transformations, $\square$, and, $\triangle$, and the functions, F_i, of these objects. However, the precise definitions and analyses are beyond the scope of this paper including investigating practical applications. Moreover, although mathematical literature on these topics is plentiful, not all mathematical resources are accessible by researchers having pure engineering backgrounds.

A. Tensors and Manifolds

Tensors have been introduced to describe multi-dimensional linear transformations [3–7]. This turned out to be very useful, for example, in modeling physical phenomena independently from the observer, and a chosen coordinate system (i.e., a frame of reference). It should be noted that multi-dimensional matrices in machine learning models are often called tensors; for example, tensors are often mentioned as low-rank approximations. However, this is a misnomer, since tensors are primarily linear transformations, which must satisfy certain properties, and not simple data structures.

Tensors exist in a vector space, V, which itself is a mathematical object. There is an isomorphism between the vector space and its dual. Linear forms map the vectors to scalars, i.e., $V \mapsto \mathbb{R}$. General tensors are of type, (p,q), where q is the covariant order, and p is the contravariant order. The order of tensor is equal to the number of upper and lower indices required to address its components.

The covariant and contravariant components differ whether they change along or against the change of the basis. They correspond to the orthogonal and parallel projections, respectively. A simple linear transformation, $V \mapsto V$, is a $(1,1)$-tensor. The $(0,1)$ tensor is a simple linear form or covector, whereas $(1,0)$ tensor is a contravariant vector. The multi-linear forms combine multiple vectors, and produce a scalar; the vector dot-product is a special case of a bilinear form. The dot-product (more generally, an inner product) induces a metric structure on the vector space. The tensor product, which is generally non-commutative, is a product of constituent multi-linear forms. The purpose of the tensor product is to linearize multi-linear forms. For example, a bilinear form, $V \times V \mapsto \mathbb{R}$, is a covariant 2-tensor of type $(0,2)$. The matrix determinant is an example of a multi-linear form. The multi-linear forms can be also considered for complete vector spaces.

Manifolds are another important structure that can be assumed for advanced modeling of complicated spaces (informally, shapes). They are parameterized locally as Euclidean sub-spaces of the same dimension. The collection of these overlapping sub-spaces and their pairwise mappings is referred to as an atlas of the underlying space. The smooth manifolds are differentiable, and allows calculus. The special case is tangent manifolds that can be assigned a tangent plane at every point. It is sometimes desirable to follow a continuous curve that lies within the manifold. The manifold hypothesis assumes that manifolds are low-dimensional smooth objects that are embedded in high-dimensional spaces. Thus, manifolds represent the target phenomenon to be modeled, which are observed in high-dimensional noise space.

Computing and machine learning models can benefit from considering the definitions and properties of standard tensors. The signal and data processing can be made more robust by extracting information manifolds from noisy measurements. These methods are mathematically more evolved, and usually not considered in standard textbooks. For example, the manifolds can be considered for analyzing and visualizing deep learning models with large number of parameters.

B. Set Theory and Logic

Zermelo-Fraenkel (ZF) set theory is a very general formal approach to ordinary mathematics [8–10]. It adopts a small set of axioms, which are then used to derive the true statements about the universe of sets. The Axiom of Choice assumes that the elements in any subset can be distinguished. Such a property cannot be proved, but it is fundamental in the proofs of many theorems. The key idea is that many mathematical objects, such as numbers can be encoded as sets including natural, ordinal and cardinal numbers. Basic arithmetic operations can be defined for all these number sets. It is possible to compare the sizes of even uncountably infinite sets.

Binary relations on sets allow defining their orders. The relations can be transitive, symmetric, and reflexive. The set is said to be well-ordered if the ordering is possible for all pairs of elements, and every subset has the smallest element, otherwise the set may be only partially ordered. Equivalence relations allow partitioning elements into equivalence classes.

Mathematical statements are often formulated as logical statements. Even the notion of mathematical proof can be precisely defined. There is presently no proof that the ZF theory is consistent (i.e., there are no statements that contradict themselves). The logical system is said to be complete, if any true statements that can be described in this system can be proved from axioms. The theorems proved by Gödel in 1930's showed that no mathematical system can be complete and consistent at the same time [11]. Moreover, consistency of a system cannot be proved even from within that system.

Mathematical logic is formal, unambiguous language of statements and their proofs. This language is exact, verifiable and reproducible. Propositional logic formalizes deductions (i.e., drawing specific conclusions from proofs or general knowledge). It is language with syntactical rules how mathematical statements are constructed as strings. There are also associated semantic rules allowing valuations of statements for given truth assignments of propositional variables. The valuations can be satisfiable and logically valid. Logical formulas are logically equivalent to Boolean functions. More importantly, every satisfiable propositional theory is consistent, and every complete propositional theory is satisfiable.

Propositional logic is only concerned with implications, which is rather limiting. First-order logic extends propositional logic with quantifiers, predicates, relations, functions and constants. It is sufficiently powerful, so that many mathematical concepts can be naturally expressed as the first-order formulas. There are again precise syntactic rules how to define statements, and semantic rules how to interpret their validity for given truth assignments. Gödel's theorems are normally considered within first-order logic.

Mathematical logic can greatly expand the capabilities of computational systems. Gödel's completeness and incompleteness theorems are fundamental for understanding the limits of Artificial Intelligence (AI) systems. Moreover, since mathematical statements are much more constrained than statements in natural languages, it facilitates automated reasoning systems and formal provers.

C. Abstract Algebra

Abstract Algebra is a classical field of mathematics, which is normally covered in undergraduate curricula as part of mathematical, but not engineering programs. Some introductory textbooks that are more accessible to beginners are, for example, [12–17]. The main idea is to consider sets that have special properties. The elements of these sets are mere representations of mathematical objects; the actual objects considered are not important. This is a very powerful abstraction, which has connections to many other areas of mathematics including geometry, topology, and number theory.

The most common objects in Abstract Algebra are groups, rings, and fields. The key property is that a small number of unary and binary operations defined over the elements of these sets are closed. It is often useful to consider subsets, so that they retain the key properties of the original set. These subsets are called sub-groups and sub-rings, respectively. The existence of inverse elements allows defining complementary operations, such as subtraction for addition, and division for multiplication. The groups can be cyclic, and generated by a small number of elements. The permutation groups are useful in systematically searching a space of all possibilities. The cosets allow natural partitioning and factoring of groups. Multiple groups can be combined using internal and external direct products. Group homomorphism enables finding representative sub-groups of small order. Rings adds additional properties to groups. Integral domains and their special case, fields, have additional properties over those that are required for rings.

There are many other advanced group topics, which may be useful for defining and working with complex engineering and machine learning models. Examples of these advanced topics include geometric constructions of groups, classes and their equations, group symmetries, groups with additional conditions yielding many specific properties, actions on groups, group compositions, group representations, and other. Abstract Algebra allows exploring interesting analogies between different mathematical objects, such as integers and polynomials. There is an interplay between group theory and linear algebra, whereas number theory can be interpreted geometrically.

Commutative rings have applications in algebraic coding theory and cryptography. Complex shapes of molecules and crystals can be described using the concept of groups. Fields and Galois theory are of interest when designing data processing and machine learning methods involving finite precision numerical values. Such number representations greatly reduce the memory requirements of large models, as well as speed up arithmetic computations while providing exact results.

Newer research problems can consider how to define abstract representations of large complex models, such as those that are used in deep learning applications. If these abstract representations form a group, many properties and results from group theory would immediately follow. This can pave the way for more guided and interpretable design of deep learning models, and create new connections between statistics, probability, and algebraic methods.

D. Algebraic Geometry and Topology

Topological spaces generalize metric spaces, which generalize vector spaces [18][19]. Topology allows comparing shapes of geometrical objects. For example, the topologist cannot tell the difference between a circle and a square that lie in a two-dimensional plane. More specifically, topology is a collection of certain subsets of a given set. It is useful to consider continuous maps between topological sets. A special, but common case is Euclidean topology.

Homeomorphism is a relaxed notion of equality in topology. It is a continuous bijection, but it may be too restrictive when comparing shapes. The important property of homeomorphic spaces is that they share all topological invariants including connectedness (spaces are a single piece), compactness (spaces are closed and bounded), and Hausdorffness (all points can be separated by spaces). Topological spaces can be compactified by adding extra points. New topological spaces can be created from simpler topological spaces using Cartesian products and connected sums. It is also possible to create new topological spaces using group actions. It allows partitioning the space into pieces that are connected by group elements.

An important concept in topology are identification spaces. They are unified representations of topological shapes, which greatly simplifies reasoning and visualizations. The surfaces are topological equivalents of manifolds defined in Calculus. The identification spaces of complex topological surfaces are collections of polygons. The polygons consist of vertices, edges and faces; they can be viewed as hyper-graphs. In addition, it can be shown that any closed surface is homeomorphic to either a sphere, a torus, or a projective plane.

Homotopy can be used to define equivalences between maps of topological spaces. It is much less restrictive than homeomorphism, since the former only requires equivalence of topological structures. Homotopy classes of loops in a topological space forms a fundamental group of that space. The fundamental group is another topological invariant.

Homology plays a fundamental role in topological algebra. It assigns algebraic objects to topological spaces, so that they remain unchanged under homeomorphic deformations. This can be used to detect holes in different number of dimensions. Homotopy can be used to represent topological spaces as simplical complexes, which can be practically obtained, for example, using triangular meshing. Such a compact representation is suitable for developing algorithms for counting holes within topological spaces; these counts are known as Betti numbers. In practice, it is sufficient to assume abstract simplical complexes that do not depend on particular geometric embeddings. The boundary maps are connected sums of simplices along the boundary. Considered recursively, the underlying linear boundary maps form a chain of simplical complexes of decreasing orders. The boundary kernel and the previous boundary image then define a homology group.

Persistent homology enables algorithmically evaluating topological features in metric spaces, such as point clouds [20]. The idea is to associate points with parameterized simplical complexes, such as Vietoris-Rips complex. When the distance parameter increases, the underlying homology changes; this process is referred to as filtering. The task is to identify, which homological features are most persistent during the filtration. The filtered simplical complexes can be then assumed to be approximations of the underlying topology of data points. Topological Data Analysis (TDA) is very robust in scenarios with missing and noisy data. It can be visualized, for example, as barcodes, and persistence diagrams. However, computing the filtration is numerically very costly.

Geometric Algebra is concerned with k-vectors as a generalization of ordinary 1-vectors that are placed in $(n \geq k)$-dimensional vector spaces [21–24]. The 0-vectors are referred to as scalers, or simply scalars. Every k-vector has a magnitude, direction, and orientation. The k-vectors can be multiplied using an outer (wedge) product, which is anti-commutative. There is also an inner product of k-vectors. These products have simple geometric interpretation; i.e., the wedge product of parallel or linearly dependent vectors is zero, and so is the inner product of orthogonal or linearly independent vectors. Geometric product combines both inner and outer products. Moreover, every k-vector has its inverse, which can be used to divide k-vectors; such operation is not defined for the standard vectors. Multi-vectors are the sums of simple k-vectors with possibly different orders, k.

The power of Geometric Algebra lies in its ability to greatly simplify mathematical models and expressions, especially those that are related to calculus over vector fields. Multi-vectors can be also used to compactly define differentiable manifolds using directional and geometric derivatives.

E. Category Theory

Category Theory generalizes foundational mathematical concepts by formulating a high-order theory, which is governed by the laws of free algebra [25–29]. Categories consist of objects, and morphisms between these objects. Morphisms are maps that are associative, and composable, and as any other maps, they can be surjective, injective, or both. The focus is on high-level properties of mathematical objects, while abstracting away intrinsic details. A graph-like structure of categories defines external associations between objects, which can yield information about axioms, rules of association, and the properties that are transferable between objects, as well as universal to all objects. Categories can be studied as hierarchical graphs of objects and their morphisms, or as algebras of functions, i.e., functions of objects, and composition of functions. The associativity and composability of morphisms implicitly restrict the structure of categories.

Some examples of categories include sets and functions over sets, vector spaces and linear transformations, metric spaces and continuous maps, and groups and homomorphisms. The categories are described assuming generalizations of common objects, such as initial and terminal objects, pushouts and pullbacks, (co-) products, (co-) limits, and (co-) completeness. For instance, products and pullbacks generalize limits, whereas sums and pushouts generalize co-limits. The morphisms between complete categories are referred to as functors. Like morphisms, functors must preserve composition. For example, there is a functor from category of topology spaces with continuous maps to category of sets of points with functions between these sets. The functors themselves can form a category with morphisms referred to as natural transforms. Further generalizations lead to the notion of fibers, and sheafs. The idea of fibers is to index one category over another category akin to indexing of one set over another set. The sheafs introduce locally defined data that are attached to open sets in a topological space.

The composability of large systems from simple parts using simple rules is one of the demonstrated applications of Category Theory. The objective is to scale-up systems without unnecessarily increasing the system complexity, for example, by avoiding creating unnecessary couplings.

IV. Discussion and Future Outlook

Modern modeling methods must incorporate advanced mathematical objects to be capable of describing processes and phenomena across different abstraction layers and at different scales. Mathematical objects themselves can be subject to algebraic operations and functions in order to form, for example, an algebraic group. Mathematical objects can be assigned numerical values to enable calculus in addition to manipulating their structure. For example, graphs can be assigned numerical values as attributes to all vertices and edges. The numerical values can be assigned assuming certain probability distributions, which would lead to signal and data processing methods over these mathematical structures.

Mathematics has been traditionally much closer to computer science than to engineering. Mathematics likely has many results that are very useful in creating and analyzing advanced models of modern intelligent and autonomous systems, but these results are mostly unknown in engineering. Bridging such a knowledge gap requires a dialogue between engineers and mathematicians. The former can identify unsolved practical problems, while the latter may suggest relevant mathematical concepts and tools that could be considered.

Another highly desirable strategy is to revise the current engineering curricula that can no longer keep up with very fast evolving complex technologies including the emergence of AI powered systems. It is crucial to identify knowledge invariants with a long-term validity. This also concerns choosing the key topics in applied or even pure mathematics that should be added to engineering curricula. However, teaching advanced mathematics to engineering students can be challenging. It may require choosing plenty of motivating examples to gain the intuition, even before attempting to solve any problems.

A very interesting research avenue is re-representing the results from mathematical literature, so they become accessible to technical researchers without training in advanced mathematics [30–37]. Many mathematical concepts can easily become inaccessible even to expert mathematicians, depending how deeply a given topic is explored. In general, mathematical literature appreciates and strives for notational brevity, which is a roadblock for engineers. It would be very desirable to explain common mathematical notations to non-mathematicians including how to read mathematical literature.

However, neither mathematical objects, nor algebraic structures were explicitly defined in this paper. At present, it is unclear how to define the models involving advanced mathematical objects and structures that could be constructed and analyzed with moderate efforts. The justification of these models may require substantial research in order to demonstrate their effectiveness as well as interpretability. It also includes computational feasibility. There is likely a trade-off between model compactness, informativeness, and its utility in the real-world engineering applications.

Furthermore, large language models and other AI tools undoubtedly have enormous impact on teaching and research. However, the current generation of these tools did not reach the required level of factual accuracy. Fortunately, these issues are more of a concern when working on advanced tasks that require more in-depth investigations. For introductory teaching and learning of advanced mathematics (and other subjects), the AI tools can reliably identify and briefly summarize the key concepts and terminologies, and play a role of a powerful interactive tutor akin to how Wikipedia was used until recently. Nevertheless, hand-picked, human chosen teaching and learning materials still appear to be much better suited to the student needs, especially when these materials are prepared by teachers who have many years of teaching and research experience.

V. Conclusions

The main objective of this paper was to outline mathematical structures beyond vectors and matrices as the first-class compositional objects that could be used in defining advanced mathematical models. Basic algebraic operations and functions could be defined for general mathematical structures to construct other such structures. These operations and functions cannot be arbitrary, but they must be rigorously derived, for example, using external associations of categories in category theory. In addition, mathematical structures can be assigned numerical attributes, which would allow numerical calculations over these structures. The important question is whether rather complex mathematical notations, and indeed even mathematical theories themselves could be simplified for the pragmatic use in engineering. Moreover, the strict mathematical rigor could be sacrificed in exchange of allowing novel practical modeling and design methods for complex engineering systems.

It is obvious that there are many open research problems, and issues to be resolved. The future work will consider how to present more advanced mathematical concepts to engineering students. It will lead to defining many open research problems. For example, one can define attached different numerical attributes to different components of mathematical objects, and then consider signal processing over these objects.

References

[1] N. J. Higham, *Functions of Matrices*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2008, ISBN: 9780898716467.

[2] B. V. S. Kumar and H. Dutta, *Discrete Mathematical Structures, A Succinct Foundation*. Boca Raton, FL: CRC Press, 2020, ISBN: 9780429624742.

[3] D. A. Fleisch, *A Student's Guide to Vectors and Tensors*. Cambridge, UK: Cambridge University Press, 2012, ISBN: 9780521171908.

[4] J. H. Heinbockel, "Introduction to tensor calculus and continuum mechanics", Lecture Notes, 2001, [Online]. Available: https://www.academia.edu/105055636/.

[5] M. Tabatabaian, *Tensor Analysis for Engineers*, 3rd ed. Dulles, VA: Mercury Learning And Information LLC, 2023, ISBN: 1683929628.

[6] J. Tromp, *A Geometrical Introduction to Tensor Calculus*. Princeton, NJ: Princeton University Press, 2025, ISBN: 9780691269221.

[7] N. Islam, *Tensors and Their Applications*. New Delhi, India: New Age International (P) Ltd., Publishers, 2006, ISBN: 8122418385.

[8] A. Karagila, "Axiomatic Set Theory", Lecture Notes, 2025, [Online]. Available: https://karagila.org/files/set-theory-2017.pdf.

[9] M. Bays, "B1.1 Logic", Lecture Notes, 2025, [Online]. Available: https://people.maths.ox.ac.uk/bays/teaching/b1logic/B1.1.pdf.

[10] M. Bays, "B1.2 Set Theory", Lecture Notes, 2025, [Online]. Available: https://courses.maths.ox.ac.uk/pluginfile.php/128327/mod_resource/content/1/B1.2-HT25.pdf.

[11] P. Smith, *An Introduction to Gödel's Theorems*, 2nd ed. Cambridge, UK: Cambridge University Press, 2013, ISBN: 9781107606753.

[12] L. Alcock, *How to Think About Abstract Algebra*. Oxford, UK: Oxford University Press, 2021, ISBN: 9780198843382.

[13] J. A. Gallian, *Contemporary Abstract Algebra*, 10th ed. Boca Raton, FL: CRC Press, 2021, ISBN: 9780367651787.

[14] A. W. Knapp, *Basic Algebra and Advanced Algebra*, 2nd ed. East Setauket, New York: Published by the Author, 2016, ISBN: 9781429799928.

[15] S. Shahriari, *Algebra in Action: : A Course in Groups, Rings, and Fields*. Providence, Rhode Island: American Mathematical Society, 2017, ISBN: 9781470428495.

[16] J. Scherk, *Algebra, A Computational Introduction*, 2nd ed. Toronto, Canada: Creative Commons Copyright, 2009.

[17] A. Macdonald, *Linear and Geometric Algebra*. CreateSpace Independent Publishing Platform, 2025, ISBN: 9781453854938.

[18] D. Chatterjee, *Topology, General & Algebraic*. New Delhi, India: New Age International Pvt. Ltd., Publishers, 2007, ISBN: 9788122419436.

[19] T. Leinster, "General Topology", Lecture Notes, 2015, [Online]. Available: https://www.maths.ed.ac.uk/~tl/topology/topology_notes.pdf.

[20] T. Needham, "Introduction to Applied Algebraic Topology", Lecture Notes, 2019, [Online]. Available: https://tgda.osu.edu/math-4570-applied-algebraic-topology/.

[21] C. Perwass, *Geometric Algebra with Applications in Engineering*. Berlin Heidelberg: Springer-Verlag, 2009, ISBN: 9783540890676.

[22] D. Hildenbrand, *Foundations of Geometric Algebra Computing*. Berlin, Heidelberg: Springer-Verlag, 2013, ISBN: 9783642317941.

[23] L. Dorst, D. Fontijne, and S. Mann, *Geometric Algebra for Computer Science*. San Francisco, CA: Morgan Kaufmann, 2007, ISBN: 9780123749420.

[24] U. Görtz and T. Wedhorn, *Algebraic Geometry I: Schemes*, 2nd ed. Wiesbaden, Germany: Springer Fachmedien Wiesbaden GmbH, 2020, ISBN: 9783658307325.

[25] D. I. Spivak, *Category Theory for the Sciences*. Cambridge, MA: MIT Press, 2014, ISBN: 9780262028134.

[26] E. Riehl, "Category theory in context", Lecture Notes, 2023, [Online]. Available: https://math.jhu.edu/~eriehl/161/context.pdf.

[27] F. W. Lawvere and S. H. Schanuel, *Conceptual Mathematics*, 2nd ed. Cambridge, UK: Cambridge University Press, 2009, ISBN: 9780521719162.

[28] B. Fong and D. I. Spivak, *An Invitation to Applied Category Theory*. Cambridge, UK: Cambridge University Press, 2019, ISBN: 9781108711821.

[29] S. Roman, *An Introduction to the Language of Category*. Cham, Switzerland: Birkhäuser, 2017, ISBN: 9783319419176.

[30] J. Gallier and J. Quaintance, "Algebra, Topology, Differential Calculus, and Optimization Theory For Computer Science and Machine Learning", Lecture Notes, 2025, [Online]. Available: https://www.cis.upenn.edu/~jean/math-deep.pdf.

[31] S. Serovajsky, *Architecture of Mathematics*. Boca Raton, FL: CRC Press, 2021, ISBN: 9780429893544.

[32] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, *Geometric deep learning grids, groups, graphs, geodesics, and gauges*, 2021. arXiv: 2104.13478v2 [cs.LG].

[33] D. Simovici, *Mathematical Analysis for Machine Learning and Data Mining*. Singapore: World Scientific Publishing Co. Pte. Ltd., 2018, ISBN: 9789813229709.

[34] M. Ghayoumi, *Mathematical Foundations for Deep Learning*. Boca Raton, FL: CRC Press, 2026, ISBN: 9781040389072.

[35] S. Warner, *Pure Mathematics for Beginners*. Self-published: Get 800, 2018, ISBN: 9780999811757.

[36] R. M. Golden, *Statistical Machine Learning, A Unified Framework*. Boca Raton, FL: CRC Press, 2020, ISBN: 9781138484696.

[37] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, *Mathematics For Machine Learning*. Cambridge, UK: Cambridge University Press, 2021, ISBN: 9781108455145.

 ISBN: 978-1-68558-360-6