



Issues in Informing Science + Information Technology

An Official Publication
of the Informing Science Institute
InformingScience.org

IISIT.org

Volume 15, 2018

A DATA SCIENCE ENHANCED FRAMEWORK FOR APPLIED AND COMPUTATIONAL MATH

Kirby McMaster	Weber State University, Ogden, UT, USA	kmcmaster@weber.edu
Samuel Sambasivam*	Azusa Pacific University, Azusa, CA, USA	ssambasivam@apu.edu
Brian Rague	Weber State University, Ogden, UT, USA	brague@weber.edu
Stuart Wolthuis	BYU-Hawaii, Laie, HI, USA	swolthuis@byuh.edu

*Corresponding author

ABSTRACT

Aim/Purpose	The primary objective of this research is to build an enhanced framework for Applied and Computational Math. This framework allows a variety of applied math concepts to be organized into a meaningful whole.
Background	The framework can help students grasp new mathematical applications by comparing them to a common reference model.
Methodology	In this research, we measure the most frequent words used in a sample of Math and Computer Science books. We combine these words with those obtained in an earlier study, from which we constructed our original Computational Math scale.
Contribution	The enhanced framework improves the Computational Math scale by integrating selected concepts from the field of Data Science.
Findings	The resulting enhanced framework better explains how abstract mathematical models and algorithms are tied to real world applications and computer implementations.
Future Research	We want to empirically test our enhanced Applied and Computational Math framework in a classroom setting. Our goal is to measure how effective the use of this framework is in improving students' understanding of newly introduced Math concepts.
Keywords	framework, applied math, computational math, data science, concordance

Accepting Editor: Eli Cohen | Received: December 2, 2017 | Revised: April 9, 2018 |
Accepted: April 16, 2018.

Cite as: McMaster, K., Sambasivam, S., Rague, B., & Wolthuis, S. (2018). A data science enhanced framework for applied and computational math. *Issues in Informing Science and Information Technology*, 15, 191-206.
<https://doi.org/10.28945/4032>

(CC BY-NC 4.0) This article is licensed to you under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/). When you copy and redistribute this paper in full or in part, you need to provide proper attribution to it to ensure that others can later locate this work (and to ensure that others do not accuse you of plagiarism). You may (and we encourage you to) adapt, remix, transform, and build upon the material for any non-commercial purposes. This license does not permit you to use this material for commercial purposes.

INTRODUCTION

A theoretical *framework* for an academic discipline is a set of interrelated concepts that allow new facts and ideas to be viewed and integrated into a specific context. Synonyms for the word *framework* are paradigm, mental model, and gestalt.

In his book *What the Best College Teachers Do*, Ken Bain (2004) observes that:

“Students bring paradigms to the class that shape how they construct meaning. Even if they know nothing about our subjects, they still can use an existing mental model of something to build their knowledge of what we tell them.”

Multiple frameworks are available in almost every subject matter field. Examples of frameworks include:

1. Newton’s laws of motion in Physics.
2. Mendeleev’s periodic chart of elements in Chemistry.
3. Linnaean system of classification of organisms in Biology.
4. Fundamental reporting objectives described in the FASB Concepts of Financial Accounting.
5. The OSI Reference Model for computer networks and data communications.

FRAMEWORKS FOR MATHEMATICS

In Mathematics literature, much of the presentation is in the form of theorems and proofs. However, according to Polya (1945) in his classic book *How to Solve It*, there are two sides to Mathematics.

“Studying the methods of solving problems, we perceive another face of mathematics. Yes, mathematics has two faces; it is the rigorous science of Euclid, but it is also something else. Mathematics presented in the Euclidean way appears as a systematic, deductive science, but mathematics in the making appears as an experimental, inductive science. Both aspects are as old as the science of mathematics itself.”

A few other authors have identified similar dual frameworks for Mathematics. Henrici (1974) describes the difference between *dialectic* and *algorithmic* mathematics.

“Dialectic mathematics is a rigorously logical science, where statements are true or false, and where objects with specified properties either do or do not exist. Algorithmic mathematics is a tool for solving problems.”

Body and Soul is a program in mathematics education in Sweden. Contributors to the program have written books and software for teaching applied mathematics with a blend of constructive/computational (*body*) and analytical /symbolic (*soul*) elements. Eriksson (2003) writes:

“It would of course be ideal to combine the more rigorous aspects of mathematics with the problem-solving aspects.... This may sound very difficult, but it is exactly what the three volume textbook *Applied Mathematics: Body and Soul ...* aims to do: to combine the constructive / computational (body) aspects with the symbolic (soul) aspects of mathematics in undergraduate teaching.”

Additional literature references on general frameworks for Mathematics (beyond theorems and proofs) are hard to find. Most recent discussions are from educators, who set standards on how to teach Math in high school and lower grades (e.g. California Common Core State Standards: Mathematics).

APPLIED AND COMPUTATIONAL MATH

The terms “Applied Math” and “Computational Math” are often used interchangeably. Princeton University has offered a program in Applied and Computational Math for over a decade. The Princeton program includes courses in a variety of Math fields, such as Statistics, Operations Research, and Differential Equations. The emphasis in these courses is on solving scientific problems that include a substantial mathematical component.

One possible way of distinguishing between Applied Math and Computational Math is the following. Applied Math is primarily concerned with developing abstract mathematical models to represent the essential features of real world situations. Computational Math, on the other hand, deals more with how to express mathematical models and algorithms in a computer language.

In our previous research, we identified frameworks for Mathematics that are relevant to real world situations and computers (McMaster, Rague, McMaster, & Blake, 2008). We developed two scales for Mathematics:

- (1) a *Traditional Math* scale that describes the logical structure of mathematical topics, and
- (2) a *Computational Math* scale that emphasizes using mathematical models and algorithms to solve problems on a computer.

These scales were constructed from words used frequently in samples of “pure math” and “applied/computational math” books.

Our Computational Math scale combines the meanings of Applied Math and Computational Math described above. The frequent Math words used to construct this scale are shown in Table 1. This table contains a measure of relative frequency for the concepts (adjusting for books of different sizes).

Table 1. Computational Math Scale
Based on a sample of Applied Math and Computational Math books.

Concept	No. of Books	Relative Freq.
problem	25	389.1
algorithm/method	24	346.0
solution	25	314.3
variable/value	24	267.1
equation	20	265.2
function	24	263.4
model	20	223.5
system	23	167.2
condition/constraint	25	163.8

Some of the individual words are combined into “word groups” on the scale when they appear to represent related concepts. For example, the words *algorithm* and *method* are often used as synonyms in Math books (e.g. Simplex Algorithm vs. Simplex Method). A *variable* consists of a set of *values*, along with defined operations on those values.

Informed by this Computational Math scale, for our broad framework for Applied and Computational Math we visualize problem-solving as a process that involves three “worlds”: Real World, Abstract (Math) World, and Computer World. On our Computational Math scale, the word *problem* represents the Real World situation. The final *solution* obtained with the assistance of a computer represents Computer World. Most of the other scale concepts fall primarily in the abstract world of Mathematics, although some of these words (e.g. *variable*, *function*, *system*) have meanings in more than one world.

Our explanation for this preponderance of Math World concepts is that we obtained words that appeared frequently in Applied and Computational Math books. Our original sample of books did not include substantive fields (e.g. Physics, Biology) or Computer Science.

DATA SCIENCE

For several years, we have thought about ways to expand our list of Computational Math concepts to include representatives from Real World and Computer World. Recently, our research focus shifted toward a relatively new field called Data Science. In preliminary research, we collected a sample of 18 Data Science books and research papers. We determined which words (excluding common English words) appeared most often in these documents. In our sample, 7 Data Science words had total frequencies greater than 2000. These words, along with their word counts, are listed in Table 2.

In this list of frequent Data Science words, our attention was first drawn to the extremely high word count for *data*. It is not surprising that *data* is the most frequent word, since the name of the field is Data Science. But in our work in other fields, we have never seen a case where the most frequent word occurs four times more often than the next most frequent word.

Table 2. Data Science Frequent Words

Based on 18 DS books and papers.

DS Word	DS Count
data	12348
model	3036
science	2393
value	2808
function	2486
variable	2300
number	2041

Our next observation was that four of the seven Data Science words in Table 2 (*model*, *value*, *function*, and *variable*) appear on our Computational Math scale. This allows us to view Data Science as a Computational Math field. From this perspective, our Computational Math framework could be improved by adding *data* and other Data Science concepts.

PURPOSE OF THIS RESEARCH

The primary objective of this research is to develop an enhanced framework for Applied and Computational Math. The revised framework should better relate abstract mathematical models to real-world data (Applied Math). It should also describe how mathematical models and algorithms can be implemented on a computer (Computational Math). The enhanced framework will improve our ex-

isting Computational Math scale by integrating selected concepts from the field of Data Science. This new framework can serve as an instructional foundation from which students are better able to connect mathematical concepts to real-world applications.

METHODOLOGY

The methodology used to enhance our Computational Math framework involves five stages. First, draw a sample of Mathematics and Computer Science books, organized into categories. Second, determine the most frequent words in each of the books. Third, standardize the word counts so that books of different sizes are comparable. Fourth, calculate the average of the standardized frequencies for words in each category. Fifth, select candidate concepts from the prior calculations to enhance our Computational Math framework.

SAMPLE OF BOOKS

Our combined sample of Mathematics and Computer Science books was 151. The books were divided into six categories to see which areas of Math and CS support different Computational Math concepts.

Table 3. Book Counts by Category

Code	Category Description	Books
APPL	Applied Math	19
COMP	Computational Math	20
STOR	Statistics & Operations Research	21
OOP	Object-Oriented Programming	31
DB	Database	30
SE	Software Engineering	30
Total		151

A list of the book categories (Codes and Descriptions) is shown in Table 3, along with the number of books in each category. Three of the categories represent Applied and Computational Math, while the other three categories apply to fields in Computer Science. The category counts for books are irregular because, as the data were collected over time, different numbers of books became available.

Many of the books in the APPL category include the word “Applied” or “Application” in the title. On the other hand, most of the COMP category books include the work “Computational” or “Computing” in the title. The remaining Math category is STOR, which contains books on Statistics, Operations Research, and Simulation. The books in the STOR category, especially Statistics books, share themes with the field of Data Science. Table 4 lists one representative book from each of the six categories.

Table 4: Example Books by Category

Code	Author	Title
APPL	Sullivan & Mizrahi (2004)	Mathematics: An Applied Approach
COMP	White (2003)	Computational Mathematics
STOR	Snedecor & Cochran (1989)	Statistical Methods
OOP	Wu (2009)	Object-Oriented Programming with Java
DB	Connally & Begg (2004)	Database Systems
SE	Pressman (2004)	Software Engineering

By adding Computer Science books to the sample, we revealed concepts that pertain to converting abstract models and algorithms into the language of computers. The OOP category consists of books for several current programming languages that support object-oriented programming. The books in the DB category focus on “data” and relational database management systems, with a strong emphasis on the central concept of Data Science. The SE category books explain common problems involved in developing software systems.

Book concordances

For each book in our sample, we needed a list of the most frequent concepts in that book. Budget and resource constraints prohibited us from acquiring complete texts in digital form, followed by processing the texts for word counts. Instead, we chose books for which Amazon provides a concordance. This constrained our sampling, but we were able to include a diverse set of books in each of the six categories.

An Amazon concordance lists the 100 most frequently used words in a book, not including common English words (Fry, Kress, & Fountoukidis, 1993). For each word in a concordance, the frequency and rank (1 to 100) of the word are provided. We collected similar concordance data in earlier research studies. The first of these studies allowed us to build the original Computational Math scale.

STANDARD FREQUENCIES FOR WORDS

Raw frequencies for a word (e.g. “model”) in different books are not directly comparable. Larger books with more total words tend to have larger counts for individual words. To make word counts from various size books comparable, we rescaled each word frequency in each concordance into a *Standard Frequency* score, using the following steps.

1. For each book, calculate the *average frequency* for all 100 words in its concordance.
2. For each word in a concordance, divide the word frequency by the average frequency for the 100 concordance words.
3. Multiply this ratio by 100 to yield the *Standard Frequency* for the word in the specific book.

For example, suppose the word “model” has a word count of 1275 in a concordance, and the concordance has an average frequency of 492. Then the Standard Frequency for “model” in this concordance is:

$$\text{Standard Frequency} = 100 * (1275/492) = 100 * (2.591) = 259.1$$

A Standard Frequency of 100 represents the result for the average word in a concordance. A word with a Standard Frequency of 200 appears twice as often as the average concordance word. A Standard Frequency of 50 indicates that the word appears half as often as the average concordance word.

Please note that Standard Frequency scores are calculated only for words that appear in a concordance. If the word “model” does not appear in the concordance for a book, “model” does not have a Standard Frequency score for that book.

An *average Standard Frequency* was calculated for each word in a category by calculating the average of the Standard Frequency scores for category books that include the word in their concordances. For example, if the word “model” appears in concordances for 17 of the 20 books in a category, then only the Standard Frequency scores for those 17 books are averaged. Each book that does not contain a particular word in its concordance is ignored in calculating the average Standard Frequency for the word in the book’s category.

ENHANCED COMPUTATIONAL MATH WORDS

In this research, we were searching for words to add to our Computational Math framework. We wanted our criteria for identifying candidate words to be reasonably flexible. Basically, we desired words that appear “often” in books in a category.

“Often” can be measured in terms of how many books in a category contain the word in their concordances. “Often” can also be defined by the average Standard Frequency for the word in a category. We combined these two meanings of “often” by defining the following thresholds:

1. We consider a word to be “often” if it appears in at least 75% of the concordances for books in the category. For example, if a category has 20 books, then an “often” word should appear in at least 15 concordances for the category. For a category of 30 books, 75% is 22.5 books, which we rounded up to 23.
2. In addition, for words that meet this “75% rule” for a category, the average Standard Frequency should be above 150 for the word in that category. That is, the word appears more than 1.5 times as often as the average concordance word.

To summarize, we are interested in words that appear on most of the concordances in a category. On these concordances, these words appear frequently.

ANALYSIS OF FREQUENT WORDS

We performed our data analysis of word frequencies in concordances for Mathematics and Computer Science books to (1) reaffirm concepts on our original Computational Math framework, and (2) add concepts that would improve the framework’s coverage of Real World applications and Computer implementations.

We first determined which original CM framework concepts appear often in the three categories of Math books. We then checked which of these current CM concepts are also supported in the Programming, Database, and Software Engineering books.

The larger task was to find candidate concepts that should be added to the CM framework. We did this by looking for new concepts that appear frequently in the concordances. We performed this analysis separately for the Math and Computer book categories.

In the next section, we applied the above results to form an *enhanced* Computational Math framework. Some judgment was required in the selection of candidate words to construct this framework.

ORIGINAL CM CONCEPTS IN MATH BOOKS

We examined the concordances of the Applied Math (APPL), Computational Math (COMP), and Statistics/OR (STOR) books to find which CM framework words are used frequently. To be considered a “frequent” word, the word had to appear in at least 75% of the concordances of books in a category. The word must also have an average Standard Frequency above 150 for the category. The original CM scale words that satisfy these constraints are listed alphabetically in Table 5.

Table 5: Original CM Words in Math Books

CM Word	Category	Books	Avg Std Freq	DS Words
algorithm	COMP	16	197.3	1261
equation	APPL	19	276.5	175
	COMP	16	185.8	
function	APPL	17	308.7	2486
	COMP	20	226.8	
	STOR	18	189.4	
method	COMP	17	222.2	1186
model	STOR	20	284.5	3036
problem	APPL	19	300.0	1305
	COMP	18	262.5	
	STOR	19	239.2	
solution	APPL	19	228.5	879
	COMP	16	166.5	
value	STOR	21	169.5	2808
variable	STOR	20	161.3	2300

For comparison, Table 5 also includes word counts for the CM words as they appear in the set of Data Science documents mentioned in the Introduction.

Function and *problem* appear frequently in all three Math categories. *Equation* and *solution* are frequent words in both the APPL and COMP categories. Two of the words (*algorithm* and *method*) are frequent only in the COMP category.

The remaining three words (*model*, *value*, and *variable*) are frequent in the STOR category. *Function* (APPL, 308.7) has the highest frequency. *Variable* (STOR, 161.3) has the lowest frequency for words that appear in a single category in Table 5.

Three of the original CM scale words (*system*, *condition*, and *constraint*) are not listed in Table 5 because they did not occur often enough in our current sample of Math books to qualify as “frequent” words. Overall, the concepts in our original CM framework, with these three exceptions, are well-supported in our current sample of Applied Math books.

ORIGINAL CM CONCEPTS IN COMPUTER BOOKS

We next measured the support for CM framework words in the Computer books. The results are presented in Table 6. Again, the table includes word counts for these words in the Data Science documents.

Table 6: Original CM Words in Computer Books

CM Word	Category	Books	AvgStd Freq	DS Words
method	OOP	25	333.8	1186
model	SE	27	186.3	3036
	DB	24	161.1	
system	SE	29	366.7	653
	DB	28	176.1	
value	OOP	30	174.4	2808
	DB	28	157.6	

Four CM words appeared often enough in concordances to qualify as “frequent” words. Each Computer book category contains 30 or 31 books, so a frequent word had to appear in about 23 (75%) or more concordances. As in the previous section, the lower limit for average Standard Frequency remained at 150.

Model, *system*, and *value* are frequent words in two of the Computer book categories. *Method* is frequent in just one category. The meaning of these words in the Computer books is not always the same as in the Math books. For example, *method* has a high frequency in the OOP category because object-oriented programming organizes methods as components of a class. In Applied Math, *method* is often a synonym for *algorithm*.

It is interesting to note that the remaining original CM framework concepts listed in Table 5 are not frequent words in the Computer books. However, the Computer books provide more than just partial support for the current CM framework.

ENHANCED CM CONCEPTS IN COMPUTER BOOKS

The main reason for including Computer books in this study was to suggest new concepts to add to our Computational Math framework. We were especially interested in words that describe CM features for Real World applications and Computer World implementations. The number of candidate words grew, so we split the list into two tables based on word frequency. The first of these tables is Table 7.

Table 7 shows nine new candidate CM framework words that meet the 75% rule and have an average Standard Frequency above 200 for at least one Computer book category. The rows, listed by decreasing average frequency, present all book categories for a word when the additional categories have a frequency above 150. Only *data* and *object* are frequent in more than one category.

The word *data* (highlighted in the Data Science documents) is one of two concepts that have an average Standard Frequency above 400. This word appears in all 30 Database (DB) concordances. In 10 of these concordances, *data* has a rank of 1. Related words provided by DB category books are *database*, *table*, and *query*. Most of the books in this category emphasize relational databases.

Table 7: Enhanced CM Words in Computer Books

CM Word	Category	Books	AvgStd Freq	DS Words
class	OOP	31	469.0	685
data	DB	30	406.0	12348
	SE	26	167.3	
software	SE	30	383.8	297
database	DB	30	321.6	549
table	DB	27	316.8	805
object	OOP	31	309.1	420
	DB	23	207.8	
process	SE	30	262.0	690
project	SE	26	233.6	634
query	DB	25	210.6	149

The Object-Oriented Programming (OOP) category contributes the frequent words *class* (Avg StdFreq 471.1) and *object*. These words are closely related in OOP. A *class* is a section of code that describes data and behaviors for an object. The *objects* are created as “instances” of the class. The word *object* also appears often in the Database books.

The Software Engineering (SE) category adds the words *software* (Avg StdFreq 383.8), *process*, and *project*. *Software* appears often as part of the field name Software Engineering, as well as to designate all non-hardware components of a computer system. *Process* (or *processing*) describes how data is transformed in the system. To effectively solve “big” problems involving large amounts of data and processing, the work is often organized into *projects*.

Some of these concepts apply to more than one phase in problem-solving. *Data* is collected in the Real World, but it is managed and transformed in the Computer World using algorithms developed in the Math World. Much computer implementation work starts out as abstraction from Real World to Math World. Project management starts early in Real World and continues as maintenance in all three worlds.

MORE CM CANDIDATES IN COMPUTER BOOKS

Table 8 lists eight less frequent CM candidate words, each of which (for some category) satisfies the 75% rule and has an average Standard Frequency between 150 and 200. Five concepts represent the OOP category, while the remaining three are for SE books.

Several of these concepts are of special interest, especially with regard to how they describe the Real World application and the Computer World implementation. A *requirement* indicates work to specify what data is needed to solve a problem. *Design* covers a variety of tasks, including how to collect the data. This usually involves some form of sampling and measurement.

Table 8: More CM Candidates in Computer Books

CM Word	Category	Books	AvgStd Freq	DS Words
string	OOP	26	182.2	344
type	OOP	28	175.7	926
requirement	SE	25	173.4	74
program	OOP	27	173.4	251
design	SE	30	166.0	294
code	OOP	29	161.8	920
development	SE	30	155.5	158
file	OOP	26	154.5	691

Program and *code* are general terms that refer to the instructions given to a computer. How these instructions are organized and presented depends on the programming language. Permanent storage is needed for both programs and data. A *file* is a common system resource to provide this storage. If the amount of data becomes large (e.g., “big data”), a special type of highly scalable database system is often justified.

MORE CM CANDIDATES IN MATH BOOKS

In Table 9, we present three candidate words mentioned in the Math books that were not part of the original Computational Math scale.

Table 9: More CM Candidates in Math Books

CM Word	Category	Books	AvgStd Freq	DS Words
distribution	STOR	18	173.7	1316
point	COMP	18	167.9	1848
	APPL	19	158.2	
number	COMP	20	154.1	2041

Each word qualifies as a frequent word based on the “75% rule” and has an average StdFreq score above 150. One explanation for the brevity of this list is that the current CM scale words were obtained using a similar methodology.

All Table 9 words have a standard frequency below 200. A *distribution* is an important concept in Statistics and Data Science, with or without reference to probability models. Distributions are often used to express patterns in data.

Point appears as a frequent word in two categories. *Point* and *number* (and *string* in Table 8) specify different types of data. A data *type* (also in Table 8) consists of a set of values, plus operations to transform these values (e.g. string concatenation). In current OOP languages, user-defined data types are created using *classes* and *objects* (see Table 7).

ENHANCED COMPUTATIONAL MATH FRAMEWORKS

The main purpose of the Computational Math (CM) framework is to provide a context in which students and practitioners can better understand and use Math concepts, both in Real World applications and in Computer implementations. Several years ago, we developed a CM scale that integrates a short list of concepts. This scale was presented in this paper as Table 1.

Most of the words on our original CM scale were Math concepts, even though the framework was intended to include relationships between Math and Real World applications, and Math as described to a computer. In this study, we searched for additional words that better express the conversion of applications to Mathematical abstractions, followed by the expression of Math models and algorithms in a programming language. We present two versions of an enhanced CM framework--a shorter version and a slightly longer version. The versions differ only in the words added to our original CM scale.

ENHANCED COMPUTATIONAL MATH FRAMEWORK - VERSION 1

Our first Enhanced CM framework is displayed in Table 10. This framework includes all of the words in our original CM framework (see Table 1). We added all nine of the words in Table 7, which are frequent in the Computer books but are not on the original CM scale. All of these words have an average Standard Frequency above 200 for at least one Computer category.

The added words are shown in *italics* in Table 10. The only subjectivity in the choice of these words is in setting the lower limit for average Standard Frequency at 200.

Table 10: Enhanced CM Framework -Version 1
(Added concepts in *italics*)

Real World	Abstract World	Computer World
problem	model	solution
<i>data</i>	algorithm/method	system
<i>process</i>	variable/value	<i>software</i>
<i>project</i>	function	<i>class/object</i>
	equation	<i>database</i>
	condition/constraint	<i>table/query</i>

The *data* concept applies in some form to all three worlds. We placed data in the Real World, since that is where it is tied to the application. *Process* and *project* issues usually begin before mathematical abstractions are proposed to explain the data.

We placed six of the enhanced concepts (as four word groups) in the Computer World portion of the framework. They describe features of computer programs and how data is stored and transformed using a database. No concepts were added to Abstract World in this version of the Enhanced CM framework.

ENHANCED COMPUTATIONAL MATH FRAMEWORK - VERSION 2

A more detailed Enhanced CM framework is shown in Table 11. This version adds two Real World words, one Math World word, and three Computer World worlds (with *program* and *code* combined

into one word group). The Math word was taken from Table 9, while the remaining words were from Table 8.

Table 11: Enhanced CM Framework - Version 2
(Additional concepts in **bold**)

Real World	Abstract World	Computer World
problem	model	solution
<i>data</i>	algorithm/method	system
<i>process</i>	variable/value	<i>software</i>
<i>project</i>	distribution	program/code
requirement	function	<i>class/object</i>
design	equation	<i>database</i>
	condition/constraint	<i>table/query</i>
		file

The choice of words to add in Version 2 of the Enhanced CM framework was subjective. We added words that we thought added informative descriptions of the activities within a section of the framework. However, any of the words in Table 8 or Table 9 could have been chosen for our Version 2 Enhanced CM framework. The reasons for our word choices are presented next.

The *requirement* addition to Real World emphasizes the importance of stating what data and what processing will be needed to solve the problem. The *design* activities describe how the data will be collected, organized, maintained, and transformed.

The statistical concept of a *distribution* in Abstract World highlights patterns in values of variables, often expressed as special types of functions. *Program* and *code* in Computer World refer to the general nature and organization of software, which may or may not include features of object-oriented programming (such as classes and objects). Programs and data are often organized into *files*, even when a database system is implemented.

This version of the Enhanced CM framework explains in more detail how to view Math models and algorithms as the link between Real World problems and finished Computer systems. The framework provides a more complete context for understanding the central role of Mathematics in applications and computing.

ELEMENTARY SCHOOL EDUCATION EXAMPLE

We now provide a simple case study (not a formal experiment) in which the application of the Enhanced Computational Math framework helps students understand Math concepts. The situation is an elementary school course on computer literacy and keyboarding. Students are in grades 1 thru 6, so the Math concepts in this example are relatively simple (but still important).

Each student is given a small packet of Skittles candy. The research question is: Which Skittles color is most common? The students open their packets and record the color of each Skittle they find. (Students are told not to eat their Skittles until the exercise is completed.) The packets are small, so the total number of Skittles for each student is about 11. This is a Real World activity that introduces students to data.

The concept of a *frequency distribution* is explained in class, initially in the form of a table with a Skittles count for each color. Percentage distributions may also be discussed. A visual representation of a distribution is then displayed in the form of a *bar graph*. These are Abstract World topics.

The students are then assigned the task of putting their raw data into an Excel *spreadsheet*. They usually have been introduced to Excel in earlier lessons. Once the data is in Excel, the students must create a frequency distribution for the data. Then they define a bar graph for this distribution. These steps use Computer World to reinforce the concept of a distribution.

So far, each student has a small sample of data, which has been summarized in a distribution. When students compare their results, they can see that not all distributions are the same. To obtain a better answer to the research question, the teacher combines the data from all students and forms a single distribution for the entire class (along with the associated bar graph). This combined distribution can be used to determine which Skittles color is most common (or conversely, if all colors are produced in approximately equal amounts).

When students complete this Skittles assignment, they have learned about a distribution as more than a Math concept. They can also remember this concept in terms of Skittles (Real World) and bar graphs generated in Excel (Computer World). The distribution concept presented in this three-world framework helps students remember and better understand the underlying Mathematics.

The target audience of this case study is elementary school students. However, we can safely say that the underlying mathematical framework can be applied appropriately to students in higher grades, such as high school and college.

SUMMARY AND CONCLUSIONS

In previous research, we developed a framework for Computational Math. This framework describes the central role of Mathematics in linking real world problems and computer solutions. We constructed this CM framework from words used frequently in a sample of Applied Math books.

The purpose of the CM framework is to help students and practitioners better understand new mathematical concepts by viewing them in the context of applications and computers. Unfortunately, our original framework placed more emphasis on mathematical abstraction, with less discussion of application and computer issues. In hindsight, the focus on the mathematical section of the framework appears to be due to our sample consisting entirely of Math books.

Recently, we explored concepts in a relatively new field called Data Science. In a sample of Data Science documents, the most frequent word mentioned was *data*. We re-examined the sample of books used to construct our original CM framework and found that the word *data* was used unevenly in the Math books. The word *data* did not appear often enough to satisfy our criteria for inclusion in the framework.

We sought to improve our CM framework by adding concepts that describe real world and computer implementation activities. We drew a new sample that included Applied Math books and Computer Science books. The books in this sample were grouped into 3 Math categories and 3 Computer categories.

We listed words used frequently in the book categories. From this new set of words, we selected a subset to add to the original CM framework to create an Enhanced CM framework. Most of the added words came from the Computer books. These words describe real world data definition and data collection, as well as how to organize OOP programs and maintain databases. The Enhanced CM framework still has mathematical abstraction in the central role for problem-solving, but *data* is now the primary thread throughout the framework.

FUTURE RESEARCH

As Data Science continues to rise in demand and popularity, textbooks germane to this new discipline will increase in number. Current trends in the development of Data Science focus on four key areas: computation, statistics, mathematics, and data visualization. We would like to review the key words inherent in the content of these four unique yet integrated disciplines to learn their common lexicon.

We would also like to empirically test our Enhanced Computational Math framework in a classroom setting to measure how effectively the use of this framework improves students' understanding of newly introduced Math concepts.

REFERENCES

- Bain, K. (2004). *What the best college teachers do*. Harvard University Press.
- Connolly, T., & Begg, C. (2004). *Database systems* (4th ed). Addison Wesley.
- Eriksson, K., Estep, D., & C. Johnson, C. (2003). *Applied mathematics: Body and soul volume 1*. Springer.
- Fry, E. B., Kress, J. E., & Fountoukidis, D. L. (1993). *The reading teacher's book of lists* (3rd ed). Center for Applied Research in Education.
- Henrici, P. (1974). Computational complex analysis. *Proceedings of the Symposium in Applied Mathematics* 20. <https://doi.org/10.1090/psapm/020/0349957>
- McMaster, K., Rague, B., McMaster, T., & Blake, A. (2008). Two gestalts for mathematics: Logical vs. computational. *Information Systems Education Journal*, 6(20).
- Polya, G. (1945). *How to solve it*. Princeton University Press.
- Pressman, R. (2004). *Software engineering* (6th ed). McGraw-Hill.
- Snedecor, G., & Cochran, W. (1989). *Statistical methods* (8th ed). Iowa State University Press.
- Sullivan, M., & Mizrahi, A. (2004). *Mathematics: An applied approach* (8th ed). Wiley.
- White, R. (2003). *Computational mathematics* (1st ed). Chapman and Hall. <https://doi.org/10.1201/9780203494479>
- Wu, T. (2009). *An introduction to object-oriented programming with java* (5th ed). McGraw-Hill.

BIOGRAPHIES



Dr. Kirby McMaster is retired from the School of Computing at Weber State University. To remain active, he has been a visiting professor at several colleges and universities. His primary research interests are in database systems, software engineering, and frameworks for Mathematics and Computer Science.



Dr. Samuel Sambasivam is Chair Emeritus and Professor of Computer Science at Azusa Pacific University. His research interests include computer security, optimization methods, expert systems, client/server applications, database systems, and genetic algorithms. He served as a Distinguished Visiting Professor of Computer Science at the United States Air Force Academy in Colorado Springs, Colorado during 2008-09 & 2016-17 academic years. He has conducted extensive research, written for publications, and delivered presentations in Computer Science, data structures, and Mathematics. He is a voting member of the ACM.



Dr. Brian Rague is Professor and Chair of the School of Computing at Weber State University. His research interests involve parallel computing, programming languages, and opportunities for creative software engineering in the fields of education, biomedicine and physics.



Stuart L. Wolthuis is Assistant Professor and Chair of the Computer & Information Sciences Department at Brigham Young University--Hawaii. His teaching focus includes software engineering, HCI, and information assurance. He brings almost 24 years of service in the USAF to the classroom with real world experiences as a program manager and software engineer. When not enjoying Hawaii's great outdoors, his research interests include melding together information systems and marine biology. His current project, Ocean View, will link land-locked educators and students to live underwater ocean views via an educational website.