An Analysis of the Math Requirements of 199 CS BS/BA Degrees at 158 U.S. Universities

CARLA E. BRODLEY, Center for Inclusive Computing at Northeastern University, USA
MCKENNA QUAM, Khoury College of Computer Sciences, Northeastern University, USA
MARK A. WEISS, Knight Foundation School of Computing and Information Sciences, Florida International University, USA

For at least 40 years, there has been debate and disagreement as to the role of mathematics in the computer science curriculum. This paper presents the results of an analysis of the math requirements of 199 Computer Science BS/BA degrees from 158 U.S. universities, looking not only at which math courses are required, but how they are used as prerequisites (and corequisites) for computer science (CS) courses. Our analysis shows that while there is consensus that discrete math is critical for a CS degree, and further that calculus is almost always required for the BS in CS, there is little consensus as to when a student should have mastered these subjects. Based on our analysis of how math requirements impact access, retention and on-time degree completion for the BS and the BA in CS, we provide several recommendations for CS departments to consider.

CCS Concepts: • **Social and professional topics** → **CS1**.

Additional Key Words and Phrases: broadening participation in computing, introductory sequence, math requirements, ABET

ACM Reference Format:
Carla E. Brodley, McKenna Quam, and Mark A. Weiss. Accepted to appear in the CACM. An Analysis of the Math Requirements of 199 CS BS/BA Degrees at 158 U.S. Universities. 1, 1 (April Accepted to appear in the CACM), 13 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

This paper presents the results of an analysis of the math requirements of 199 CS BS/BA degrees from 158 U.S. universities. Our motivation stems from three observations. First, in prior work on degree complexity [27], we observed that, in the sixty schools analyzed, mathematics (particularly calculus) represented a barrier to student progression in the CS sequence. Second, as part of the Center for Inclusive Computing's (CIC)[1] work, we performed all-day site visits at the CS departments of 54 universities and observed significant variation in both the number and placement of required

[1] The CIC (www.cic.northeastern.edu) is a national effort to create systemic, sustainable change in U.S. universities to broaden participation in computing. The CIC works with and funds universities to make systemic changes to the way in which they offer their introductory CS sequence with the goal that true beginners to computing can discover, thrive and persist [14–16, 27, 30, 31]. Ensuring a pathway for true beginners is important for broadening participation in computing because those who are true beginners are often from populations that have been historically marginalized in CS.

Authors' addresses: Carla E. Brodley, c.brodley@northeastern.edu, Center for Inclusive Computing at Northeastern University, 177 Huntington Ave, Boston, Massachusetts, USA, 02115; McKenna Quam, quam.m@northeastern.edu, Khoury College of Computer Sciences, Northeastern University, 177 Huntington Ave, Boston, Massachusetts, USA, 02115; Mark A. Weiss, weiss@fiu.edu, Knight Foundation School of Computing and Information Sciences, Florida International University, 11200 SW 8 St, Miami, Florida, USA, 33199.

Manuscript submitted to ACM

math courses. Third, at Florida International University (FIU) we observed firsthand that a BA in CS, with fewer math requirements than the BS in CS, increased student retention, diversity and demand for computer science.

The goals of this analysis are to determine: 1) if the variability in the requirements and placement[2] of math classes we observed held for a larger sample of universities; 2) if there are differences in the number of math classes for ABET-accredited[3] versus non-ABET accredited CS programs and for BA versus BS CS degrees; and 3) commonalities across institutions as to what math is required to be a successful computer scientist. In this paper we present our methodology, our results, guidelines published by ABET [3] and the ACM [26], and we make several recommendations for CS departments to consider.

2 METHODOLOGY

We started with all U.S. post-secondary bachelor degree granting institutions that graduated 150 or more students in CIP[4] code 11 for academic year 2021 as reported in the Integrated Postsecondary Education Data System (IPEDS) [33]. This led to an initial sample size of 179 distinct universities. We made the choice to select schools with the largest numbers of graduates to analyze the requirements impacting the majority of students studying CS in the U.S. (for example, if we look at CIP codes 11.0101 and 11.0701 alone – the two CIP "sub" codes that most commonly contain CS degrees [38], our sample size captures 60% of all undergraduates earning a CS degree in 2021).

We removed degrees/schools from the sample if 1) the CS major requirements are not publicly available; 2) the degree is interdisciplinary (e.g., Business, or CS+X); and/or 3) the institution is a religious institution with more than 15 credits for religion coursework.[5] The resulting sample contained 199 degrees from 158 universities spanning 39 states plus the District of Columbia and including four online universities. The state with the most representation was California with 28 degrees. Of the 158 institutions, 47 are members of the American Association of Universities [1]. Further statistics of interest are: the sample contained 80% BS and 20% BA degrees; an almost even split between degrees offered by CS departments in and out of engineering (100 and 99 degrees respectively);[6] 83 of the programs are CAC ABET accredited [3]; and 65 of the degrees are from the 55 Minority Serving Institutions (MSIs) included in the sample.

For each degree, using publicly-available degree requirements and plans of study, we recorded all math requirements. In our analysis we made the following design decisions:

- For 26 degrees, discrete math is split over two classes; we counted these as a single class because it did not change the results (e.g., if calculus 1 was required for "Discrete 1" then it is clearly also a prerequisite of "Discrete 2"). Of the 24 degrees with two discrete math classes, six were on the quarter system.
- We defined core CS classes as those that all CS majors must complete.
- We analyze the math requirements only for core CS classes because our focus is on the impact of math requirements on all CS majors at an institution. Higher-level math courses may be needed for some elective CS classes, but a student can choose to avoid those classes.
- We counted classes labeled as "Data Structures & Algorithms" twice: once as data structures and once as algorithms. Schools differ in whether this material is taught over one or two classes because if "Data Structures & Algorithms" has a math pre- or corequisite then the math class is blocking the student from both topics.

[2] By placement we mean how early in the CS curriculum a math course shows up as a pre- or corequisite to a required CS class.

[3] The Accreditation Board for Engineering and Technology (ABET) [2] is a non profit organization which certifies engineering/CS programs.

[4] The Classification of Instructional Programs (CIP) provides a taxonomic scheme that supports the accurate tracking and reporting of fields of study and program completions activity [33]. CIP code 11 is for computing majors and includes CS.

[5] We were concerned that the focus on religious curriculum would lead the faculty to down select CS and math classes in order to make it feasible for a student to graduate in four years.

[6] We count degrees from Colleges of Computer Science in the non-engineering category.

Course	Degrees	Percentage
Calculus 1	191	96.0
Calculus 2	152	76.4
Discrete	198	99.5
Probability/Statistics	140	70.4
Linear Algebra	116	58.3
Calculus 3	43	21.6

Table 1. For each math class we list the number and percentage of the 199 degrees for which the class is required.

3 RESULTS

In Table 1 we report the total number of degrees that require each of the six most prevalent math courses. For example we see that 191 of the 199 degrees require calculus 1. Indeed, we see that calculus 1 and discrete math are required by almost all programs. Calculus 3 is required by only 43 programs (the majority of which are in CS departments housed in a college of engineering and are ABET accredited). Note that some programs require a math elective and often calculus 3 is on the list of possible electives.[7]

Tables 2, 3 and 4 show the pre- and corequisite math classes for core CS classes. For example, we see in Table 3 that calculus 1 is required before discrete math in 64 of the degrees and as a corequisite in nine degrees. We chose to present the detailed data for calculus 1, calculus 2 and discrete math as these are the three most frequently required math classes and also the ones with the most impact on a student's progression through a program. More than one core CS class might explicitly list a math class as a prerequisite (for example a program might explicitly list discrete math as a prerequisite for algorithms and for computer organization because students can take algorithms and computer organization in any order but each can only be taken *after* a student completes discrete math). Note that some courses have a math class as a “transitive prerequisite" but these are not listed in the tables (e.g., if discrete math is a prerequisite for data structures, and data structures is a prerequisite for algorithms, then by transitive closure discrete math is a prerequisite for algorithms).[8]

We draw six conclusions from the data. First, 198 of the 199 degrees analyzed require discrete math be taken at some point in the degree.[9] However, the results shown in Table 2 illustrate that there is no national consensus as to which classes should directly require discrete and whether it should be listed as a pre- or corequisite.

Second, 81% of the BS degrees require both calculus 1 and calculus 2, whereas only 57% of the BA degrees require both. *When* calculus is required varies substantially [27]. This matters because the prescribed timing of when a student must take calculus can have a large impact on time-to-degree. At one end of the spectrum, 33 degree plans have calculus 1 as a pre- or corequisite to CS 1, meaning that a student has to be "calculus ready" in 24 degrees and have completed calculus 1 in 13 degrees to even begin their CS journey. In many universities a substantial proportion of incoming first year students are not calculus-ready and thus their discovery of CS is barred by this requirement (note that only about 20% of high school students take calculus [13]). A question that arises from this variability is whether the requirement of calculus as an early prerequisite stems from an actual need in CS courses or whether it is a historical artifact of how CS programs were constructed decades ago (e.g., most CS departments grew out of mathematics or computer engineering departments).

[7] Number and choice of math electives is not given in Table 1.
[8] We did not calculate the transitive prerequisites as this would have been extremely time-consuming and would not change the key points of this paper.
[9] The one exception is an application-development degree.

Courses for which Discrete is a	Pre-req	Co-req	Total
Algorithms	115	2	117
Data Structures	69	4	73
Theory of Computation	28		28
Programming Languages	22		22
Computer Organization	18		18
Intro to Databases	17		17
Computer Systems	12		12
Automata	12		12
Computer Architecture	11		11
Formal Lang/Assembly	9		9
Software Dev/Engr	8	1	9
CS 2	6	3	9
Artificial Intelligence	4		4
Operating Systems	4		4
Object Oriented Design	4		4
Intro to Cybersecurity	3		3
CS 3	3		3
Intro to Data Science	3		3
Machine Learning	3		3
Computer Networks	3		3
Randomness & Composition	2		2
Digital Logic	2		2
Great Ideas in CS	1		1
Intro to CS and Engineering	1		1
Cryptography	1		1
Secure Programming	1		1
Parallel Computing	1		1
CS 1		1	1
Total	363	11	374
No Post/Co-Req			18

Table 2. Courses for which Discrete Math is a direct pre-requisite or co-requisite (listed in descending order).

Third, there are 64 degrees that require that calculus 1 be completed before discrete math and 13 that require calculus 2 before a student is allowed to take discrete math. This also has major implications for time-to-degree because if a student fails calculus then they are delayed in taking discrete math, which along with data structures are the two courses that almost all higher-level CS classes require [27]. Because many schools have high drop/fail/withdraw (DFW) rates for calculus (in the CIC's portfolio of Implementation Grantees[10] some rates are as high as 50%), this requirement can easily add one to two semesters to a student's time-to-degree.

Fourth, which department teaches discrete math also varies. Discrete is taught by the mathematics department in 51 degrees, by the CS department in 130 degrees, and for 17 degrees a student can take discrete math from either department. We examined whether the prerequisites for discrete math differ depending on the offering department. Of the 51 degrees for which discrete math is taught by the math department, 19 have calculus as a prerequisite and for the 130 degrees for which discrete is taught by CS, 61 degrees require calculus as a prerequisite.

[10] See cic.northeastern.edu for a list of all Implementation Grantees.

Courses for which Calculus 1 is a	Pre-req	Co-req	Total
Discrete	64	9	73
CS 1	13	24	37
CS 2	12	11	23
Data Structures	11	5	16
Algorithms	8		8
Software Dev/Engr	6		6
Digital Logic	2	1	3
Computer Architecture	2		2
Computer Organization	2		2
Theory of Computation	2		2
CS 3	1	1	2
Computer Systems	1		1
Operating Systems	1		1
Numerical Methods	1		1
Parallel Computing	1		1
Computer Networks		1	1
Machine Learning		1	1
Total	128	54	180
No Post/Co Req			75

Table 3. Courses for which Calculus 1 is a direct pre-requisite or co-requisite (listed in descending order).

Courses for which Calculus 2 is a	Pre-req	Co-req	Total
Discrete	13	2	15
Algorithms	14		14
Data Structures	4	1	5
Computer Systems	1	2	3
Theory of Computation	2		2
CS 2		2	2
CS 4	1		1
Operating Systems	1		1
Introduction to Data Science	1		1
Numerical Methods	1		1
Total	38	7	45
No Post/Co Req			108

Table 4. Courses for which Calculus 2 is a direct pre-requisite or co-requisite (listed in descending order).

Fifth, not surprisingly, there are significant differences as to the number and placement of math classes for BA and BS degrees. Calculus 1 and 2 are more frequently pre- and corequisites to CS1, CS2, and discrete math in BS degrees than in BA degrees (e.g., calculus 1 is a prerequisite for discrete math in 41.5% of the BS degrees compared to just 27.5% in the BA degrees). Furthermore, BA degrees require fewer math classes than BS degrees, with an average of 3.4 and 4.9 required math classes, respectively. The distribution of math requirements in BA and BS degrees is shown in Figure 1.

Finally, we examined the difference in math requirements for ABET and non-ABET accredited programs. On average an ABET accredited program requires 5.1 math classes and a non-ABET accredited program requires 4.2 math classes, and this difference is statistically significant. We show the distribution in Figure 2. Note that for BS degrees that are

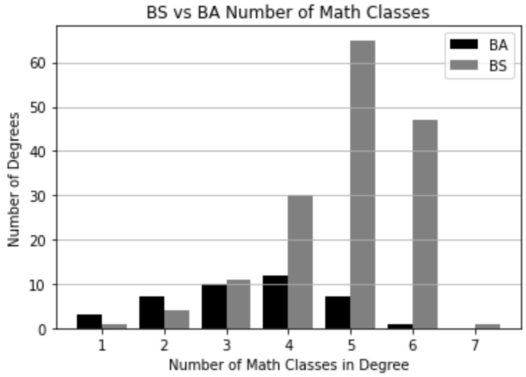

Fig. 1. Distribution of total math classes required for BA versus BS degrees in the sample.

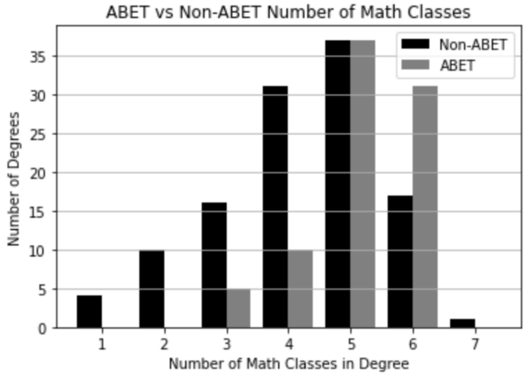

Fig. 2. Distribution of total math classes required for ABET accredited and non ABET accredited degrees in the sample.

not ABET accredited the average number of math classes is 4.6. Although ABET requires 15 credits of math for CS, schools vary in the number of credits assigned to each class and some count “highly-mathematical" CS classes toward this requirement.[11]

4 A CASE STUDY OF THE IMPACT OF MATH IN THE CS CURRICULUM

At Florida International University (FIU), the considerations cited in this paper (timing of math classes, linkages between math classes and the CS sequence, etc) came into play when they examined their ABET accredited BS in CS program in 2015. FIU is a large, urban, minority-serving public university with a majority of "non-traditional students," which the National Center for Education Statistics defines as students who either delay enrollment, attend part time, work full time, are financially independent for purposes of determining eligibility for financial aid, have dependents other than a spouse, are a single parent, and/or do not have a high school diploma [32]. Furthermore, the majority of FIU undergraduates receive Pell grants, meaning they are considered low-income. For these students, graduating with little to no debt into a well-paying job can be life changing.

[11] This is permitted as long as they have different courses in CS to meet the computing credits requirement. That is, programs cannot double count CS courses as Math and as CS to meet the ABET math and CS credit requirements.

At FIU, the CS department is housed in the engineering college. In 2015, like most programs, FIU's CAC ABET accredited program required calculus 1 and 2 (both four-credit courses), discrete math,[12] statistics, and automata theory, a course which allowed the program to meet the ABET math requirement. In addition, like the rest of engineering, FIU required physics 1 and 2 with labs (five credits each) in addition to other science courses. In other words, the CS major was very math and science heavy compared to other non-engineering degrees. Eliminating ABET accreditation was not an option, but the department was interested in offering a less math and science heavy CS degree as many students were either not successful in completing the BS in CS degree or were deciding to not pursue the degree because of the added time to degree completion and/or perceived difficulty.

In FIU's CS curriculum, calculus 1 and 2 were not prerequisites for any required CS course, but were required for physics and statistics. Many students, knowing the importance of completing certain CS courses to be able to apply for internships, and wanting to avoid semesters that included physics, calculus, and programming courses, opted to load up on CS courses early in their studies and delay the "math+physics" part. It was not uncommon to have students complete all the CS course work successfully, have a good job offer, but then have graduation held-up because they did not pass one or more of statistics, physics 2, calculus 2, or automata theory in their final semester.

As the university focused increasingly on reducing time to graduation, students were denied opportunities to switch majors after the first 1-2 years because they could not possibly do so without significant delay in graduation time. Additionally, FIU observed an unfortunate reality that many transfer students self-selected not to be CS majors, even if that had been their intent when starting at FIU. Although Florida universities in theory have strong articulation agreements with their state colleges, in practice many transfer students who were interested in CS found that they didn't have the right math courses to graduate on time. In 2015, the calculus and physics requirements in the CS degree added 18 credits to what was an already packed 60 upper division credits. This also added significantly more cost to the degree as Florida universities are mandated to charge double tuition for "excess hours" [23]. Because FIU's state college transfer student population (those coming to FIU with an Associate's degree from a two-year institution) represent more than half of the university's students, this meant large numbers of students were not pursuing computer science for reasons that were not tied to the core computer science curriculum. Another consideration was that, even pre-COVID, many working students wanted the flexibility of at least some online courses and the math and physics courses at FIU had limited, to no offerings, online because of poor student success in that modality.

With all this, FIU decided to keep its ABET accredited BS in CS and create a new BA in CS, distinguished primarily by 1) eliminating the calculus and physics requirements; 2) replacing the statistics course with one of the math department alternatives that does not require calculus as a prerequisite; and 3) moving a few required courses (such as automata theory) into the elective group. By requiring only discrete structures and statistics, the math and science requirement for the BA in CS became only slightly more than what is required for students in other non-CS and engineering majors. To be clear, although the BS and BA differ in the number of required versus elective CS courses, the main reason students cite in distinguishing the two programs is whether the calculus sequence is required.

FIU's BA in CS alternative has been wildly popular. Today, the BA degree accounts for more than 60% of CS majors, but in absolute numbers the BS program has also grown. Above all, students are empowered to choose the pathway that best meets their needs. Many BA students indicate they would have not been able to start the CS degree without this pathway. Importantly, the graduation rates in CS, which lagged the rest of the university, have caught up, alleviating the perception among students that the CS major was not worth pursuing.

[12] During this period, FIU's computer science faculty created their own discrete structures course as an alternative to the math department's offering, allowing students to choose (and knowing their likely choice). This change increased retention and progression.

Preliminary results presented in [37] indicated that while there was an immediate significant increase in both the number of computing majors overall and in the percentage of women pursing CS degrees, there were no statistically significant differences between the BA and BS degrees in enrollment on the basis of gender or on job attainment. There was a statistically significant difference in enrollment on the basis of race/ethnicity; a larger percentage of Hispanic, Asian, and Nonresident Alien students were enrolled in the BS program compared with other ethnic groups, while the BA program attracted a much higher percentage of Black students than the BS. Surveys completed by 45 BA and 89 BS graduates showed no statistically significant difference in obtaining job offers, but a statistical difference in salary distribution, favoring the BS degree. Beyond the relatively small numbers of responses to the employment survey, a notable limitation of the study was that at the time of the survey distribution, students were job hunting in the height of COVID.

With the exception of the salary disparity, many of the preliminary results have been confirmed in a more recent analysis. Notably, the percentage of computing majors who identify as Black or African American has risen from 11.0% to 13.0% since the full launch of the BA degree, even as the corresponding percentage has declined from roughly 12.5% to 11.3% of the overall FIU undergraduate student body. More recent survey results also show that the salary disparity between BA and BS pathways has disappeared. Details on this work, including results from a qualitative mixed-methods analysis are forthcoming [42].

Of course every institution is unique and has its own circumstances, but FIU's example shows that changes to required courses, prerequisites and corequisites can have significant impact.

5 RELATED WORK

In this section we review the existing literature on the role of mathematics in CS and the recommendations of the two professional organizations that provide curricular recommendations and guidelines for CS. The first, the Accreditation Board for Engineering and Technology (ABET) [2], was founded in 1932 to provide guidelines and oversight of Engineering programs. CS accreditation started in 1985 under the Computer Science Accreditation Board (CSAB), and is currently conducted by ABET's Computing Accreditation Commission [3]. The second, the Association for Computing Machinery (ACM), was founded in 1947 and is a U.S.-based international professional society for computing [4]. ACM publishes curricular guidelines for CS every ten years with the two most recent publications being ACM-2023-Gamma [26] and ACM-2013 guidelines [24].

5.1 ABET-CAC Guidlines

ABET is a non-profit organization that certifies engineering/computing programs (as well as applied and natural science and engineering technology). Its goal is to provide employers and students a guarantee of what knowledge and skills graduates will have mastered. Although likely all engineering colleges require their engineering programs to be ABET accredited, this is not true of CS programs even when the CS department sits within the college of engineering. Indeed only 83 of the 199 programs studied in our sample are CAC ABET accredited.

The 2024-2025 CAC ABET accreditation requirements state that for a CS program to be accredited it must contain 15 credit hours of mathematics and statistics, which must include discrete mathematics, probability, statistics, and have rigor equivalent to introductory calculus [3]. Note that calculus 1 is not explicitly required. Rather, the wording "rigor equivalent to introductory calculus," which was first introduced in the 2019 criteria, only precludes courses such as pre-calculus from counting toward the 15 credit hours [34]: "the intent is to signal to programs that calculus is not necessarily required if it does not support the program's objectives as well as other mathematics courses... it is hoped that programs see it as an opportunity to include mathematics courses that best support the program rather than simply requiring traditional courses such as calculus." Despite this, none of the CAC ABET accredited programs in our sample have removed calculus 1 in response to this change and only a handful have removed calculus 2. Further, the accreditation requirements make no mention of how classes should be ordered. The report does not specify which CS topics require particular math concepts and thus is silent on which CS courses should require calculus (or any other math class) as a prerequisite. Some CAC ABET accredited CS programs code computational complexity and/or algorithms as math courses, which is permitted assuming they have a sufficiently mathematical focus. Also of note, ABET does not require that the credit hours be accounted for in complete courses. So, for instance, in a program that requires calculus 1 and 2, discrete math, and statistics, but for which calculus 1 and 2 are both four credits, the 15-credit hour requirement can be met by finding one credit somewhere inside a three-credit CS course, and counting that course as one math credit and two CS credits, rather than finding a fifth math course.

Although not a subject of this paper, it is worth noting that that this year ABET CAC replaced the previous requirement of six credits of science, with the following requirement: "Coursework that develops and applies the scientific method in a non-computing area" [3]. Thus just *a single science class which meets these criteria is sufficient for CAC ABET 2024 accreditation*, which, for most universities, is no more than the general education requirements for all majors. Schools that find themselves struggling to fit all of the CS classes into their degrees as the discipline continues to evolve and expand would do well by re-examining their legacy science requirements as a first step.

5.2 ACM 2023 Guidelines

The ACM 2023 Computer Science Curricula committee issues curricular guidelines for CS undergraduate programs. This is a partnership of ACM, IEEE-CS and AAAI [26]. The current version is still in draft format, but has been published to ACM's website (https://csed.acm.org).

Historically, the ACM curricular guidelines have *not* recommended that students take calculus before CS1 or discrete math [5, 6, 24]. In the *Math Requirements* section, the 2023 report recommends that students should come to college with pre-calculus, and thus pre-calculus can be viewed as a prerequisite for discrete math. The *Core Hours* section outlines two sets of core math requirements: a minimal "CS-core" set suited to credit-limited majors and a more expansive "KA-core" set to align with technically focused programs. The only course needed to satisfy the "CS-core" set is discrete math. The "KA-core" set contains some coverage of calculus, probability and linear algebra in addition to discrete math, but does not specify how many credit hours should be devoted to their study. The *Course Packing Suggestions* section reports: "Many computer science departments now offer courses that prepare students mathematically for AI and machine learning. Such courses can combine just enough calculus, optimization, linear algebra and probability; yet others may split linear algebra into its own course. These courses have the advantage of motivating students with computing applications, and including programming as pedagogy for mathematical concepts." These CS departments have created their own math curriculum so that they are not forced to require CS students take the same math curriculum used in engineering and mathematics departments. This is the approach taken at Northeastern University where all CS majors are required to take discrete math, calculus 1 and CS2810 Mathematics of Data Models.[13] and at George Washington University which offers CS-4341/6341: Continuous Structures (https://www2.seas.gwu.edu/ simhaweb/contalg/instructors.html). The ACM's

[13] CS 2810 studies the methods and ideas in linear algebra, multivariable calculus, and statistics that are most relevant for the practicing computer scientist doing machine learning, modeling, or hypothesis testing with data. The course covers least squares regression, finding eigenvalues to predict a linear system's behavior, performing gradient descent to fit a model to data, and performing t-tests and chi-square tests to determine whether differences between populations are significant. The course includes applications to popular machine-learning methods, including Bayesian models and neural networks.

flexibility around math is recent, as, historically, the ACM curricular guidelines have favored highly technical math. Indeed, M. Tedre's history of CS education points out that the 1968 ACM guidelines were written by mathematicians not computer scientists [29].

5.3 Research on Mathematics Requirements

For at least 40 years, there has been debate and disagreement as to the role of mathematics in the computer science curriculum. Some question whether the historically high amount of math in many CS programs is still necessary [21, 35, 36], some suggesting that the fundamental math course for CS is discrete math [17], and others arguing that the community has become math-phobic [9, 39]. The overarching argument is that mathematics proficiency often correlates with success in computing [25, 41] and that mathematics reasoning and problem solving translate to skills needed for CS [7]. But the evidence is less clear. For instance, studies twenty years ago had mixed results as to whether math is a predictor of success in programming [10, 40]. More recent work [19] suggests that taking AP Calculus in high school produced better outcomes in introductory programming courses than taking AP CS. However, when controlling for students' background characteristics (such as parental education), that advantage disappeared. Among all this confusion, there are studies suggesting that mathematics requirements in STEM programs, including engineering and computing serve as a barrier to broadening participation, weeding out students who could otherwise succeed [18, 22]. Indeed Bressoud's seminal 2015 study provides a thoughtful discussion of how the current state of college calculus can deter students from mathematics and STEM majors [12].

To the best of our knowledge, this paper presents the first large-scale analysis in the last decade of the math pre- and corequisites listed on the publicly available websites for universities offering CS degrees in the U.S.[14] Papers exist on the design of the CS undergraduate curriculum [11] and on the relationship between math and CS with arguments for CS being reliant on math [28], and for math being reliant on CS [8, 20]. Perhaps the paper most relevant to this work is Baldwin, et al's [7] article mapping the connections between math and CS as laid out in the 2013 ACM guidelines [24]. They found that the most connected topic to CS was probability and statistics with content connections to higher-level CS topics such as theory of computation, information systems, networks, and computer organization. In their paper, Baldwin, et al presented a survey of 25 degree programs, for which they found the average number of required math classes is 4.5 – similar to the results of this paper.

6 CONCLUSIONS AND RECOMMENDATIONS

Our analysis shows that although the majority of CS departments in the U.S. require calculus, there is no consensus as to where in the degree it should be placed. And further that the subject material covered in calculus is likely not needed for most CS classes. It is our belief that the worst placement is as a pre- or corequisite for CS1 as this impacts a student's ability to discover CS. Indeed, as stated, many students are not calculus-ready when they enter college and thus are at risk of being delayed in taking CS1 or, worse, dissuaded from trying out and discovering the field. Only half of high schools in the U.S. offer CS so we absolutely must have a good path for students to discover CS at university. Indeed, the degree plans for 162 of the 199 CS programs share the belief that students can take CS1 without calculus as a pre- or corequisite. Finally, even when it is only listed as a co-requisite, it is not clear that taking calculus 1 and CS1 in the same semester leads to success, particularly for students who are true beginners to coding.

[14] The ACM 2023 committee made their recommendations from the following inputs: 600 responses of a survey distributed to computer science faculty across a variety of institutional types and in various countries; 865 responses from a survey sent to members of the tech industry; the math requirements stated by all knowledge areas in the report; from direct input provided by the CS theory community; and based on a review of past ACM reports including recent reports on data science and quantum computing education.

Perhaps the largest variation across the universities is the placement of discrete math in the program. We conjecture that this is because CS as a field does not yet agree on what topics need be included in discrete math, which in turn causes variation not only in the prerequisites for discrete math but also the post-requistites (i.e., which core classes require discrete math). We know anecdotally that some faculty use math prerequisites for CS classes, not because the subject material of the math course is needed, but because they want a certain "mathematical maturity" before students take the class. But what does mathematical maturity mean? If the content of a CS course does not rely on the content of a math course but instead the quantitative skills a student would build by taking that math course, then perhaps they can develop those skills in the CS course itself?

Through this analysis we arrive at two recommendations for CS programs. First, do not require calculus 1 or 2 as a prerequisite or co-requisite for CS1, CS2, or discrete math. The first class in which these math concepts *might* be used is likely data structures or algorithms (i.e., when students start to analyze run time and use big-O notation). Second, CS departments should separate progression in CS from progression in mathematics. This is particularly important if a university has a high DFW rate in calculus 1 and/or calculus 2. In this case, calculus blocks many students from being able to move forward in their CS classes. Indeed we recommend that every CS department look at the DFW rates of calculus 1 and calculus 2 at their university and strongly consider whether these classes are serving their students at all. And this is particularly important for departments with capacity constraints on the CS major, which use students' GPA in CS1, CS2 and calculus to determine who can major in CS [15]. Third, as mentioned in the ACM 2023 guidelines, departments might want to consider creating a math class taught by their own faculty to teach the CS-relevant topics from continuous math.

In summary, we invite you to ponder this ultimately puzzling question: Are we letting the math department determine who majors in CS?

ACKNOWLEDGMENTS

This project was partially funded by Pivotal Ventures and by the National Science Foundation (award 230845). We would like to thank Raj Rajenda, Rahul Simha, Felix Muzny, Monique Ross, Albert Lionelle, Megan Giordano and Catherine Gill for their comments on earlier drafts of this paper.

REFERENCES

[1] AAU. 2023. Who We Are. https://www.aau.edu/who-we-are (Retrieved December 2023).

[2] ABET. 2023. About ABET. https://www.abet.org/about-abet/ (Retrieved December 2023).

[3] ABET. 2023. Criteria for Accrediting Computing Programs 2024-2025. https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-computing-programs-2024-2025/. (Retrieved December 2023).

[4] ACM. 2023. ACM Homepage. https://www.acm.org/ (Retrieved December 2023).

[5] William F. Atchison, Samuel D. Conte, John W. Hamblen, Thomas E. Hull, Thomas A. Keenan, William B. Kehl, Edward J. McCluskey, Silvio O. Navarro, Werner C. Rheinboldt, Earl J. Schweppe, et al. 1968. Curriculum 68: Recommendations for academic programs in computer science: A report of the ACM curriculum committee on computer science. *Commun. ACM* 11, 3 (1968), 151–197.

[6] Richard H. Austing, Bruce H. Barnes, Della T. Bonnette, Gerald L. Engel, and Gordon Stokes. 1979. Curriculum'78: Recommendations for the undergraduate program in computer science—A report of the ACM curriculum committee on computer science. *Commun. ACM* 22, 3 (1979), 147–166.

[7] Douglas Baldwin, Henry M. Walker, and Peter B. Henderson. 2013. The roles of mathematics in computer science. *ACM Inroads* 4, 4 (dec 2013), 74–80. https://doi.org/10.1145/2537753.2537777

[8] Ehud Bar-On and Rachel Or-Bach. 1988. Programming mathematics: A new approach in introducing probability to less able pupils. *International Journal of Mathematical Education in Science and Technology* 19, 2 (1988), 281–297. https://doi.org/10.1080/0020739880190207 arXiv:https://doi.org/10.1080/0020739880190207

[9] Theresa Beaubouef. 2002. Why computer science students need math. *ACM SIGCSE Bulletin* 34, 4 (dec 2002), 57–59. https://doi.org/10.1145/820127.820166

[10] Jens Bennedsen and Michael E. Caspersen. 2005. An investigation of potential success factors for an introductory model-driven programming course. In *Proceedings of the First International Workshop on Computing Education Research* (Seattle, WA, USA) *(ICER '05)*. Association for Computing Machinery, New York, NY, USA, 155–163. https://doi.org/10.1145/1089786.1089801

[11] Alfs Berztiss. 1987. A mathematically focused curriculum for computer science. *Commun. ACM* 30, 5 (1987), 356–365.

[12] David Bressoud. 2015. Insights from the MAA National Study of College Calculus. *The Mathematics Teacher* 109 (10 2015), 178. https://doi.org/10.5951/mathteacher.109.3.0178

[13] David M. Bressoud. 2021. The strange role of calculus in the United States. *ZDM: Mathematics Education* 53, 3 (Jun 2021), 521–533.

[14] Carla E. Brodley. 2022. Expanding the pipeline: Addressing the distribution of prior experience in CS1. *Computing Research News* 34, 6 (June 2022). http://doi.acm.org/10.1145/1219092.1219093

[15] Carla E. Brodley. 2022. Why universities must resist GPA-based enrollment caps in the case of surging enrollments. *Commun. ACM* 65, 8 (Aug. 2022), 20–22. https://doi.org/10.1145/3544547

[16] Carla E. Brodley and Catherine Gill. 2024. The BPC relevance of common assessment in the introductory sequence. *Commun. ACM* (to appear 2024).

[17] Kim B. Bruce, Robert L. Scot Drysdale, Charles Kelemen, and Allen Tucker. 2003. Why math? *Commun. ACM* 46, 9 (sep 2003), 40–44. https://doi.org/10.1145/903893.903918

[18] Jehanzeb R. Cheema and Gary Galluzzo. 2013. Analyzing the gender gap in math achievement: Evidence from a large-scale US sample. *Research in Education* 90, 1 (2013), 98–112.

[19] Chen Chen, Jane M. Kang, Gerhard Sonnert, and Philip M. Sadler. 2020. High school calculus and computer science course taking as predictors of success in introductory college computer science. *ACM Transactions on Computing Education (TOCE)* 21, 1 (2020), 1–21.

[20] Liesbeth De Mol. 2015. Some reflections on mathematics and its relation to computer science. *Automata, Universality, Computation: Tribute to Maurice Margenstern* (2015), 75–101.

[21] John P. Dougherty. 2017. MATH COUNTS Does mathematics serve computing as a support or a barrier? *ACM Inroads* 8, 1 (2017), 31–32.

[22] Jessica Ellis, Bailey K. Fosdick, and Chris Rasmussen. 2016. Women 1.5 times more likely to leave STEM pipeline after calculus compared to men: Lack of mathematical confidence a potential culprit. *PLOS ONE* 11, 7 (07 2016), 1–14. https://doi.org/10.1371/journal.pone.0157447

[23] Florida Shines. 2023. Surcharge for Excess Hours. https://www.floridashines.org/excess-credit-surcharge

[24] Joint Task Force on Computing Curricula Association for Computing Machinery (ACM) and IEEE Computer Society. 2013. Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. (2013).

[25] John Konvalina, Stanley A. Wileman, and Larry J. Stephens. 1983. Math proficiency: A key to success for computer science students. *Commun. ACM* 26, 5 (1983), 377–382.

[26] Amruth N. Kumar, Monica D. Anderson, Brett A. Becker, Richard L. Blumenthal, Michael Goldweber, Pankaj Jalote, Susan Reiser, Titus Winters, Rajendra K. Raj, Sherif G. Aly, Douglas Lea, Michael Oudshoorn, Marcelo Pias, Christian Servin, Qiao Xiang, Eric Eaton, and Susan L. Epstein. 2023. Computer Science Curricula 2023: Version Gamma. (2023). Most recent draft of the 2023 ACM CS guidlines, published August.

[27] Albert Lionelle, McKenna Quam, Carla E. Brodley, and Catherine Gill. 2024. Does curricular complexity in computer science influence the representation of women CS graduates?. In *Proceedings of the 55th ACM Technical Symposium on Computing Science Education (SIGCSE '24, Vol. 1)*. ACM, New York, NY.

[28] Elise Lockwood, Anna F. DeJarnette, and Matthew Thomas. 2019. Computing as a mathematical disciplinary practice. *The Journal of Mathematical Behavior* 54 (2019), 100688. https://doi.org/10.1016/j.jmathb.2019.01.004

[29] Simon Matti Tedre and Lauri Malmi. 2018. Changing aims of computing education: A historical survey. *Computer Science Education* 28, 2 (2018), 158–186. https://doi.org/10.1080/08993408.2018.1486624 arXiv:https://doi.org/10.1080/08993408.2018.1486624

[30] Felix Muzny, Megan Giordano, Emma Stone, and Carla E. Brodley. 2024. Collecting, analyzing, and acting on intersectional, longitudinal data and pass/fail/withdraw rates in computing courses. In *Proceedings of the 55th ACM Technical Symposium on Computing Science Education (SIGCSE '24, Vol. 1)*. ACM, New York, NY.

[31] Felix Muzny and Michael D. Shah. 2023. Teaching assistant training: An adjustable curriculum for computing disciplines. In *Proceedings of the 54th ACM Technical Symposium on Computing Science Education (SIGCSE 2023, Vol. 1)*, Lina Battestilli, Brian Dorn, and Leen-Kiat Soh (Eds.). ACM, New York, NY.

[32] National Center for Education Statistics (NCES). 2013. Nontraditional Undergraduates. https://nces.ed.gov/pubs2002/2002012.pdf (Retrieved December 2023).

[33] National Center for Education Statistics (NCES). 2023. The Integrated Postsecondary Education Data System. https://nces.ed.gov/ipeds/. (Retrieved July 2023).

[34] Michael J. Oudshoorn, Stan Thomas, Rajendra K. Raj, and Allen Parrish. 2018. Understanding the new ABET Computer Science Criteria. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (Baltimore, Maryland, USA) *(SIGCSE '18)*. Association for Computing Machinery, New York, NY, USA, 429–434. https://doi.org/10.1145/3159450.3159534

[35] Anthony Ralston. 1981. Computer science, mathematics, and the undergraduate curricula in both. *The American Mathematical Monthly* 88, 7 (1981), 472–485.

[36] Anthony Ralston. 2005. Do we need any mathematics in computer science curricula? *ACM SIGCSE Bulletin* 37, 2 (2005), 6–9.

[37] Monique Ross, Mark A. Weiss, Lilia Minaya, Andrew Laginess, Disha Patel, and Kathleen Quardokus Fisher. 2022. Removing a barrier: Analysis of the impact of removing calculus and physics from CS on employability, salary, and broadening participation. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education-Volume 1*. 460–466.

[38] Jodi L. Tims, Cindy Tucker, Mark A. Weiss, and Stuart Zweben. 2023. Computing enrollment and retention: Results from the 2021–22 undergraduate enrollment cohort. *ACM Inroads* 14, 4 (Nov 2023), 24–43. https://doi.org/10.1145/3629980

[39] Allen B. Tucker, Charles F. Kelemen, and Kim B. Bruce. 2001. Our curriculum has become math-phobic!. In *Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education*. 243–247.

[40] Philip R. Ventura Jr. 2005. Identifying predictors of success for an objects-first CS1. *Computer Science Education* 15 (09 2005), 223–243. https://doi.org/10.1080/08993400500224419

[41] Brenda Cantwell Wilson and Sharon Shrock. 2001. Contributing to success in an introductory computer science course: A study of twelve factors. *ACM SIGCSE Bulletin* 33, 1 (2001), 184–188.

[42] Jia Zhu, Disha Patel, Camila Olivero-Araya, Albert Lluberes, Monique Ross, Mark Weiss, and Kathleen Quardokus Fisher. 2024. A Mixed-Method Study on Undergraduate Students' Rational Choice between a Bachelor of Science or Bachelor of Arts in Computer Science (manuscript in preparation). (2024).

Received January 2024; Revised April 2024; accepted April 2024