A Survey of Several Advanced Mathematical Concepts Implemented in Students' Computer Science Projects*

Faculty Poster

Vladimir V. Riabov
Department of Mathematics and Computer Science
Rivier University, Nashua, NH 03060
vriabov@rivier.edu

Mathematics has a vital role in the development of computer science, electronic systems, and numerous practical applications. The objective of this poster is to review several advanced mathematical concepts and methods (modular arithmetic; Galois fields; graph theory; singular differential equations; strange attractors; fuzzy logic, and projective geometry) that contribute into the development of applications in cryptography, numerical methods, code complexity reduction, atmospheric dynamics, expert systems, computational visualization, and other areas.

The mathematical concepts, algorithms, and codes are examined by undergraduate and graduate students in various courses taught by the author. These concepts have paved the roads for students' research projects on various applications. Each student works on a selected project analyzing algorithms, creating computer codes (in Python, MATLAB, C/C++ or Java), running them at various parameters, comparing numerical results with known data, and presenting the findings to classmates and the research community. Many students published project summaries in the Rivier Academic Journal [1] and conference proceedings available from the web [2].

The opinions on why computer science students need general knowledge of mathematical concepts have been widely discussed in academia [3]. Several scholars [8] even recommended long lists of mathematical methods and formulas (ironically named as "Computer Science Cheat Sheets") that every computer science student should be familiar with. These "Cheat Sheets" cover mostly basic mathematical concepts (e.g., series, function-value order definitions, permutations, combinations, identities, recurrences, geometry, matrices, special

functions, calculus of derivatives and integrals, Cramer's rule, etc.). Only a few complex math methods are mentioned there [8]: brief reviews of the Number Theory, Graph Theory, and the Master Method for algorithm analyses, but the advanced concepts (e.g., modular arithmetic and Galois fields; fuzzy logic; strange attractors; pattern recognition, etc.) are not included in those reviews.

The theory of numbers plays probably a unique role in the theoretical computer science and various applications. Traditionally, the related topics (e.g., numerical systems, the Fundamental Theorem of Arithmetic, primes, and coprimes) are covered in the Discrete Mathematics course. In our pedagogical practice, the more advanced topics (modular arithmetic, abstract groups, rings, integer domains, and fields) are covered in the Computer Security elective course [7], due to the fact that modern encryption methods utilize the modular arithmetic and Galois field properties framed with the Fermat's Little Theorem and properties of Euler's totient function. Java Applets [7] have been found as an effective tool to introduce these advanced topics and cryptographically-secure message digest algorithms. Many students made overviews [1, 2] on the role of number theory in modern cryptography, coding theory, Advanced Encryption Standard, Remote Authentication Dial-In User Service protocol, and Wi-Fi security issues.

In the Software Quality Assurance course, the structured testing methodology [5] and graph-based metrics [6] have been reviewed by students and applied for studying the C-code complexity and estimating the number of possible errors and the required tests for various networking systems. Comparing different code releases, it is found that the reduction of the code complexity leads to significant reduction of errors and maintainability efforts [6].

Many students selected challenging topics for their research projects in various computer science courses. Here we only make overviews of a few outstanding students' projects that have been performed using the mentioned-above advanced mathematical concepts discussed in class.

David Snogles developed the Personal Encrypted Talk system for his final capstone project [1, 2]. Its primary goal was to secure Instant Messaging communications between two parties on the Internet. Secondary objectives were Java Cryptography Architecture research and the practical experience gained by the student in the development of a scalable Java-based GUI.

Robert Marceau studied Hoare's quicksort algorithm that has become a popular sorting algorithm due to the average performance of (nlog2n), limited use of extra storage (typically (log2n) recursive calls), and better performance on average compared to heapsort algorithm. The major drawback in the quicksort algorithm is the (n2) worse-case performance, which is exhibited for some initial permutations. Robert studied this performance and offered modifications to minimize the probability that the worst-case performance will

be exhibited [1, 2].

Maxim Sukharev-Chuyan studied a simple basic model of chaotic behavior in atmospheric layers known as the Lorenz system [4]. He developed a Java code for an animation of the water-wheel model of the strange attractors for the Lorenz system [1, 2]. The visualized simulations demonstrate chaotic behavior of the numerical solution of the Lorenz system of nonlinear ordinary differential equations [4].

Kevin Gill developed the Living Mars image project [1, 4] that included topics related to computer graphics, software development, and planetary science. The purpose of the project was to create a visualization of the planet Mars as could look with a living biosphere. The algorithms and methods used in generating shadows on digital elevation models were developed in his previous study [1, 2]. These include formulas that are common in computer graphics applications and are often provided by specific frameworks (i.e., OpenGL). The basics of model rendering included the structure of the source data and the interpolation of hypsometric-bathymetric tint colors. The primary algorithm is based on the calculation of shadows using ray tracing. These methods utilized the code [1, 2] from the Kevin's jDem846 open source project.

In the course evaluations, students stated that they became deeply engaged in course activities through examining the challenging problems related to the applications of the advanced mathematical concepts.

References

[1] The rivier academic journal archive. `https://www2.rivier.edu/faculty/vriabov/students_publicat.htm`.

[2] Rivier students' articles. `https://www2.rivier.edu/faculty/vriabov/students_publicat.htm`.

[3] T. Beaubouef. Why computer science students need math. *SIGCSE Bulletin*, 34(4).

[4] E. N. Lorenz. *The Essence of Chaos*. University of Washington Press, 1993.

[5] T. J. McCabe. A complexity measure. *IEEE Transactions on Software Engineering*, SE-2(4):308–320, 1976.

[6] Vladimir V. Riabov. Methodologies and tools for the software quality assurance course. *Journal of Computing Sciences in Colleges*, 26(6):86–92, June 2011.

[7] Vladimir V. Riabov and Bryan J. Higgs. Running a computer security course: Challenges, tools, and projects: Poster session. *Journal of Computing Sciences in Colleges*, 25(6):245–247, 2010.

[8] S. Seiden. Theoretical computer science cheat sheet. *ACM SIGACT News*, 27(4).