

The Role of Linear Algebra in the Computer Science Curriculum

submitted to

2013 ACM/IEEE-CS Computer Science Curricula Joint Task Force

Dr. Jeremy Kepner (MIT Lincoln Lab, CSAIL & Mathematics Department)[chair]
Prof. Tim Davis (University of Florida Computer Science Department)
Prof. James Demmel (UC Berkeley Mathematics & Computer Science Departments)
Prof. Alan Edelman (MIT Mathematics Department & CSAIL)
Prof. Howard Elman (University of Maryland Computer Science Department)
Prof. John Gilbert (UC Santa Barbara Computer Science Department)
Prof. Michael Heath (University of Illinois Computer Science Department)
Prof. Dianne O'Leary (University of Maryland Computer Science Department)
Prof. Michael Overton (New York University Computer Science Department & Courant)
Prof. Yousef Saad (University of Minnesota Computer Science Department)
Prof. Ahmed Sameh (Purdue University Computer Science Department)
Prof. Michael Stonebraker (MIT CSAIL)
Prof. Gilbert Strang (MIT Mathematics Department)
Prof. Robert van de Geijn (University of Texas Computer Science Department)
Prof. Charles Van Loan (Cornell Computer Science Department)
Prof. Margaret Wright (New York University Computer Science Department & Courant)

1. Introduction

Computer science has delivered extraordinary benefits over the last several decades. The breadth and depth of these contributions is accelerating as the world becomes globally connected. At the same time, the field of computer science has expanded to touch almost every facet of our lives. This places enormous pressure on the computer science curriculum to deliver a rigorous core while also allowing students to follow their interests into the many diverse and productive paths computer science can take them.

As science and engineering disciplines grow so the use of mathematics grows as new mathematical problems are encountered and new mathematical skills are required. In this respect, linear algebra has been particularly responsive to computer science as linear algebra plays a significant role in many important computer science undertakings. A few well-known examples are:

- Internet search
- Graph analysis
- Machine learning
- Graphics
- Bioinformatics
- Scientific computing
- Data mining

- Computer vision
- Speech recognition
- Compilers
- Parallel computing

The broad utility of linear algebra to computer science reflects the deep connection that exists between the discrete nature of matrix mathematics and digital technology.

2. Mathematics in the Computer Science Curriculum

ACM/IEEE Task Force identifies three specific mathematical subjects that are core to computer science: calculus, differential equations, and linear algebra. In addition to these courses many Computer Science curriculums require statistics and discrete mathematics. At our own institutions requirements include the following:

- Algorithms I & II, AI; Calculus I, Discrete Math OR Probability, Linear Algebra OR Calculus II
- Algorithms I; Calculus I & II, Linear Algebra OR Physics II
- Algorithms I & II; Calculus I & II & III, Probability, Linear Algebra
- Algorithms I, Discrete Math; Calculus I & II, Statistics, Linear Algebra OR Calculus III
- Algorithms I, Discrete Math; Calculus I
- Algorithms I, Discrete Math; Calculus I & II, Statistics, Linear Algebra
- Algorithms I, Discrete Math; Linear Algebra
- Algorithms I, Discrete Math; Calculus I & II & III, Statistics, Linear Algebra
- Algorithms I, Discrete Math; Calculus I & II, Statistics, Linear Algebra
- Algorithms I, Discrete Math; Calculus I & II & III, Linear Algebra

The above sample is not a complete survey of computer science curriculums, nevertheless a few broad observations can be made:

- Linear algebra is required in about half of the computer science curriculums and is optional or not required in the other half
- Linear algebra comes near the end of the mathematical sequence (usually after calculus)

Furthermore, we can infer:

- Computer science courses cannot reliably assume that their students have an understanding of linear algebra
- Linear algebra courses can reliably assume that their students have an understanding of calculus

Based on these observations we have a variety of recommendations to offer with regards to the role of linear algebra in the computer science curriculum.

3. Recommendations

The following recommendations represent the views of the majority of contributors and fall into two categories: computer science recommendations and mathematics recommendations.

Computer Science Recommendations:

- CS1** Encourage making computer science students more aware of the importance of linear algebra in various computer science topics (e.g., internet search, computer graphics, and machine learning)
- CS2** Encourage including linear algebra in computer science theory, algorithm, and data structures courses (e.g., matrix multiply algorithms, adjacency matrix data structures, and SVD data analysis)
- CS3** Encourage making linear algebra a requirement for the computer science majors, particularly for those who are interested in advanced study

Mathematics Recommendations:

- MA1** Encourage including common computer science examples in linear algebra classes (e.g., graph analysis, 3D transformations, and speech recognition)
- MA2** Encourage using software in linear algebra classes to satisfy computer science "second language" goals (e.g., using Python, R, or Matlab)
- MA3** Encourage teaching a version of linear algebra earlier without a calculus prerequisite