Teaching Introductory Linear Algebra Incorporating MATLAB

David Hill
Temple University

dhill001@temple.edu

"Good teaching is one fourth preparation and three-fourths theatre."

Gail Godwin

"Education is the ability to listen to almost anything without losing your temper or your self-confidence."

Robert Frost

Integrating MATLAB into linear algebra courses can be accomplished in a variety of ways. For example,

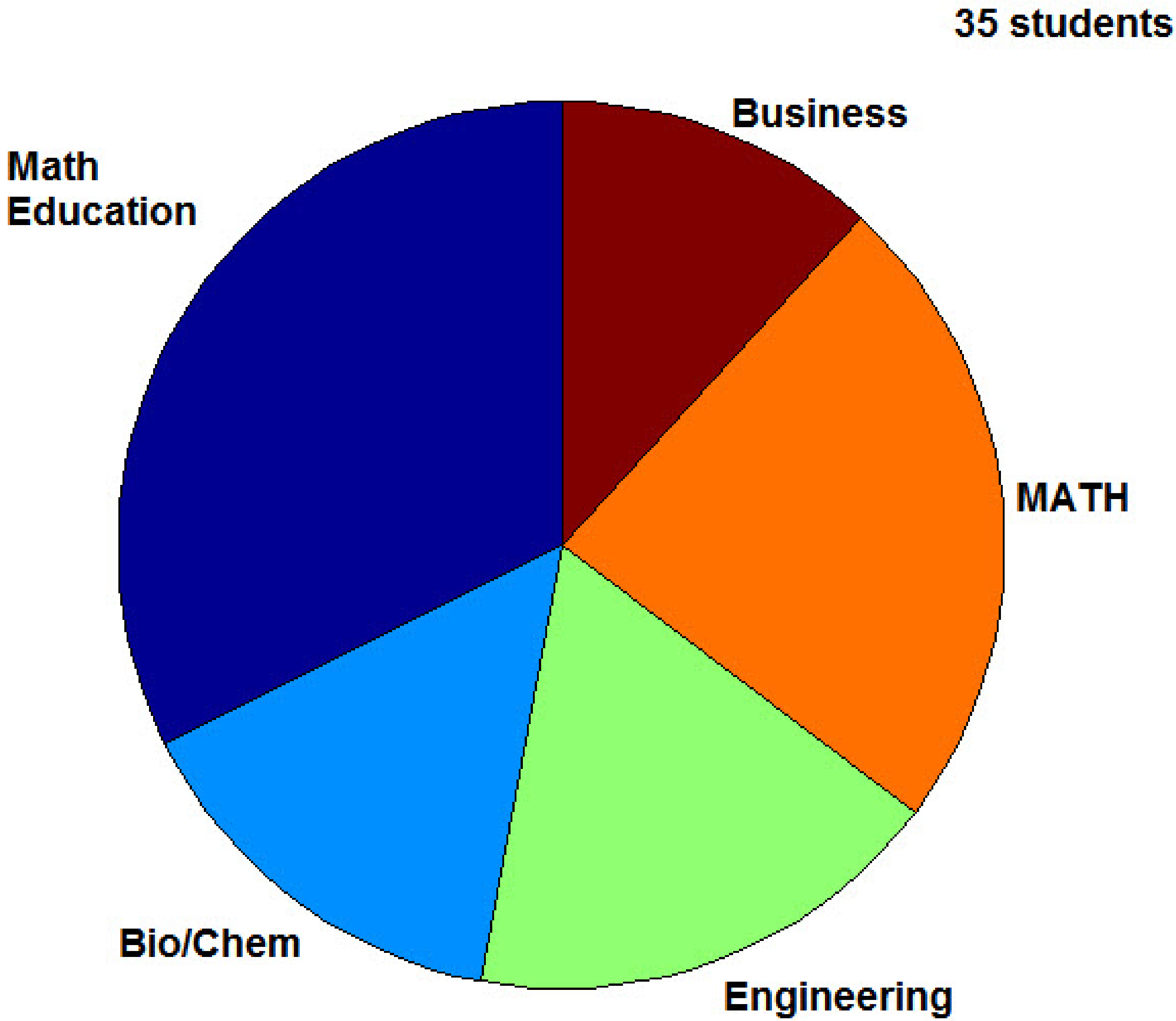
- T(eaching)-codes (Gil Strang at MIT)
- MATLAB projects (Jane Day at San Jose State University)
- MATLAB off-stage (Roe Goodman at Rutgers University)

  Involves three courses at the sophomore, junior-senior, and graduate levels with different types of involvement for both instructors and students. Lab type assignments are involved.
- Using WeBWorK with MATLAB assignments derived from the ATLAST project resources. (Tom Hagedorn at The College of New Jersey)
- A search on ‘teaching linear algebra with MATLAB’ generates a vast variety of responses.

My experience has focused on a combination lecture-lab format for a heterogeneous introductory linear algebra where MATLAB is integrated into the lecture.

Currently I’m teaching such a course minus the lab.

The students I usually teach are a heterogeneous group. The current group:



Background:

The first wave of computer technology supporting mathematics instruction surged during the 1990s. (This included MATLAB.)

Much was written about how to use the various technologies and warnings like 'Seeing is not learning!' appeared frequently.

In 1990 we obtained a grant to develop a computer based laboratory experience to accompany our general introductory linear algebra course.

We viewed this as an opportunity to engage students on a variety of conceptual levels, reveal how linear algebra and geometry are connected, and provide exploration of applications.

Over a decade or so we experimented with various instructional formats for linear algebra including

- **Lecture + Lab with MATLAB,**
- **As-A-Lab with an electronic text using MathCad,**
- **Lecture Incorporating MATLAB (no lab experience)**

Experiences led us to make observations and decisions about the mode and format we preferred to use which is Lecture + a Lab.

The diversity of the student population in terms of majors and computer experience, together with several years of experimentation led us to blend MATLAB into lectures, and not relegate computing information to the lab.

DO NOT try to teach programming in MATLAB. (Supply m-files and occasional code segments .)

We saw that it was important to establish an *esprit de corps* (a morale) in the class that linked the Math, (level appropriate) Applications, and computing with MATLAB. We wanted individual students, and the group as a whole, to have a comfort level with the integrated format.

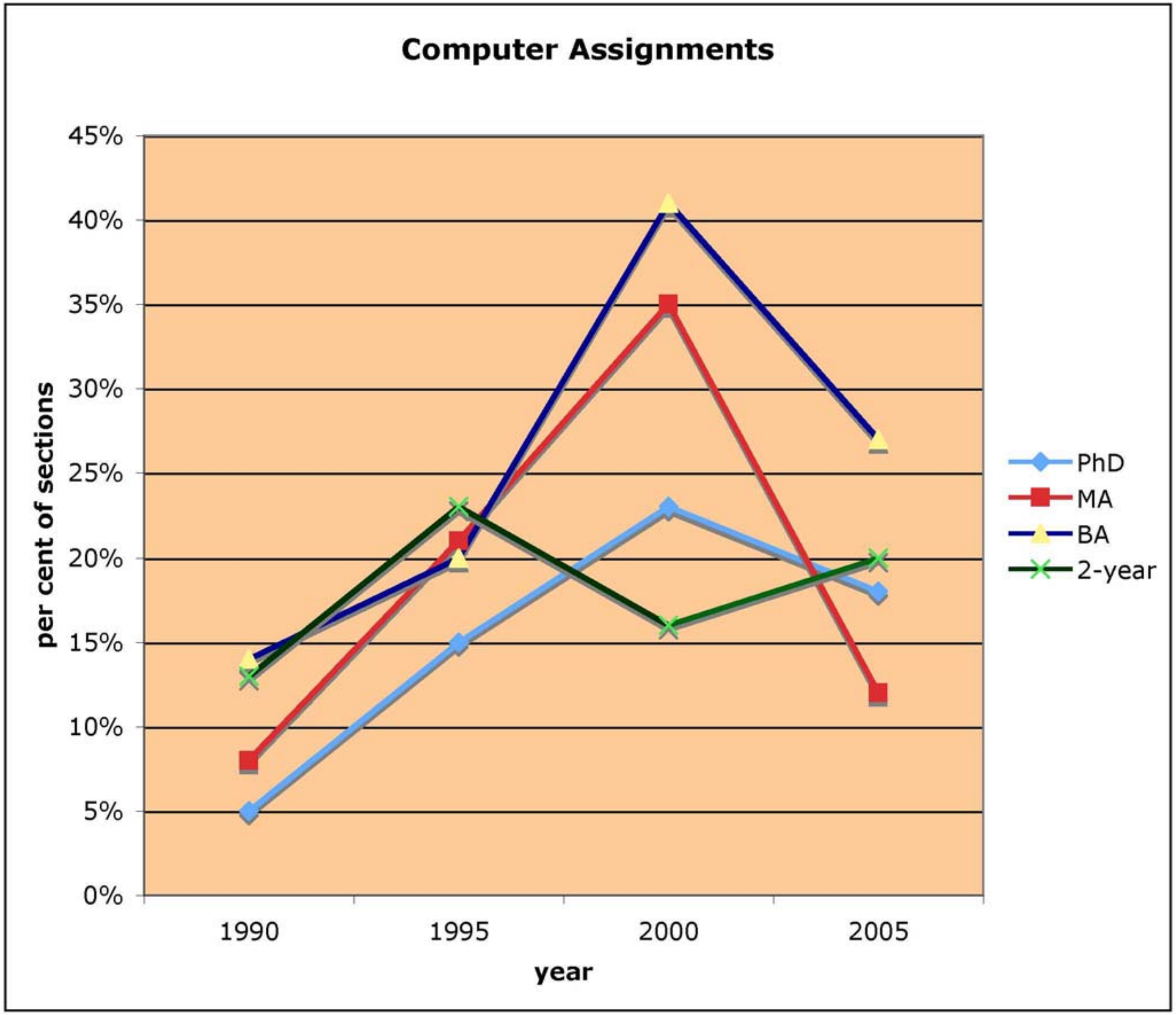
A fundamental objective was to support linear algebra concepts using the technology to provide graphics for visualization, design experiments to engage students on a variety of levels, and explore connections.

In addition, as MATLAB's capabilities grew we found that providing interactive MATLAB-based tools for concepts and applications became an important learning component.

Unfortunately the use of LABS in conjunction with lectures for mathematics has been decreasing .

This graphic is from the Conference Board of the Mathematical Sciences (CBMS).

Why the decrease? ‘This is not because they do not work, but because they are time- and energy-intensive if they are done right.’



Some of the reasons to integrate MATLAB and some of the benefits:

In the past 45+ years the linear algebra course has changed;

abstract (for serious majors)➔ a first “intro to proof” (for all math majors)

➔ matrix oriented (for a wide variety of majors)

A change is a necessity & an opportunity.

We can emphasize that linear algebra is pervasive & useful in today’s world.

No serious application of linear algebra happens without computing.

All applied science fields use computations with linear algebra models.

Experience with high quality software is needed in math-science-engineering careers as well as by math majors and future math teachers.

Students need concept images (visual models for ideas), not just concept definitions.

Labs reinforce the ideas presented in class; they bridge the gap between theory and computation.

Applications raise the awareness of the applicability of math and computing to provide models for problems and provide approaches to solutions.

We never reach SVD, but we can illustrate the (related) ideas several ways.

As the course develops we have accumulated the following facts/information.

For **symmetric** n x n matrices

have only **real** eigenvalues

eigenvectors corresponding to distinct eigenvectors are **orthogonal**

we can determine a set of **n orthonormal eigenvectors** (may need Gram-Schmidt)

have the **spectral decomposition theorem**

If n x n matrix **A** is symmetric with eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ and corresponding orthonormal eigenvectors $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n$, then

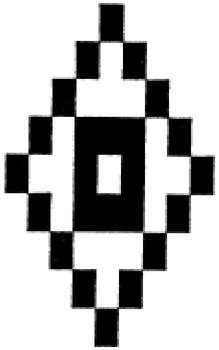
$$\mathbf{A} = \sum_{i=1}^{n} \lambda_i \mathbf{p}_i \mathbf{p}_i^T$$

If we assume that $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n|$, then **"partial sum"** $\mathbf{S_k} = \sum_{i=1}^{k} \lambda_i \mathbf{p}_i \mathbf{p}_i^T$

provides an approximation to **A** using the k most important pieces of eigen information.

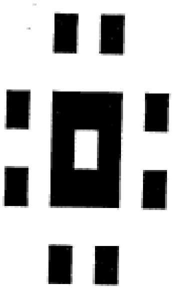
In order to **visualize** the process suppose that **A** represents digitized photographic information. To keep things simple, assume that **A** is a matrix of zeros and ones with zero representing a **white block** (or dot) on the photo and one representing a **black block** (or dot). Successive image approximations are constructed using partial sums S_1, S_2, …, converted to a black & white pattern.

Example: A 9 x 9 symmetric matrix **G** of zeros and ones represents the figure ➔



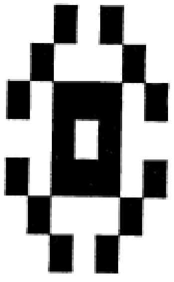
partial sum

$S_1 = \lambda_1 \mathbf{p_1 p_1}^T$



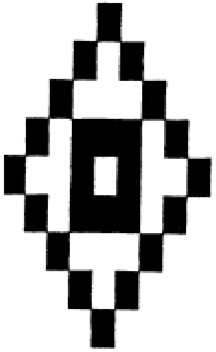
partial sum

$S_2 = \lambda_1 \mathbf{p_1 p_1}^T + \lambda_2 \mathbf{p_2 p_2}^T$



partial sum

$S_3 = \lambda_1 \mathbf{p_1 p_1}^T + \lambda_2 \mathbf{p_2 p_2}^T + \lambda_3 \mathbf{p_3 p_3}^T$



What do we do in the non-symmetric case and with color?

Answer: we can “generalize” the previous ideas to an important set of concepts that we will call **Singular Value Decomposition (SVD).**

We can illustrate this in MATLAB.

Go to MATLAB and the **imaggame**. figsc.mat

Generate approximations to a figure using the SVD of a color image. The image can be constructed using utility **makemess** which uses integers pointing to colors in an associated colormap.

Utility **makemess** creates a message on the graphics screen. An input string is converted to 8 by 6 block letter form using a built-in alphabet. The background and character color are selected by the user.

The idea of the image game is to decipher the message using the fewest singular values.

The fog Rolls IN:

In the article ***Teaching Linear Algebra: Must the Fog Always Roll In***? by David Carlson there is a lament that many instructors have expressed. It goes something like

My students grasp matrix algebra and solving linear systems by row operations,

'But when we get to subspaces, spanning, and linear independence, my students become confused and disoriented. It is as if a heavy fog has rolled in over them, and they become confused and disoriented.'

Looking at the terminology that they meet for the first time we find the list is long and interrelated.

This term I have chosen to introduce Closure and Spanning Set as we discussed matrices & n-vectors.

Vector Space

Closure

Subspace

Spanning Set

Linearly Independent set

Linearly Dependent set

Basis

For Closure there is a set of m-files that provide the opportunities to analyze whether a specified set S of matrices is closed under a particular operation. (Addition, scalar multiplication, matrix multiplication, and transpose.)

The user can explore the operation on matrices from set S both symbolically and numerically. After the exploration, the user is to write up their answer using set of guide lines are supplied.

For Span we have had a set of examples and exercises that ask students to determine a set of matrices that can be used to generate all matrices that are in a particular class of matrices.

Emphasizing linear combinations also helps prepare students for the FOG!

The jury is still out regarding the FOG! Preliminary results are encouraging.

A recent addition to the instructional arsenal.

Flap: A Matlab Package for Adjustable Precision Floating-Point Arithmetic
G. W. (Pete) Stewart (January 2009)

Flap is a package to implement floating-point arithmetic with adjustable precision. Specifically, operations are performed on Matlab doubles but are rounded to a user specified number of decimal digits after each operation. The number can be changed dynamically. Flap is intended to make it easy to generate examples of the effects of rounding error for classroom use. For more details open the file Flap.pdf.

http://www.cs.umd.edu/~stewart/flap/flap.html

Other instructional support.

Matrix Algebra Demos

FLASH videos with audio on introductory linear algebra topics:

The Method of Elimination
Matrix Addition
Scalars and Scalar Multiplication
Matrix Differences (subtraction)
Linear Combinations
Dot Product
Matrix Multiplication
Matrix-Vector Product
Matrix Transformations
Matrix Inverse
Reduced Row Echelon Form (easy)
Reduced Row Echelon Form (typical)
Solving a 3 by 3 Linear System
Solving a Homogeneous System

Determinants of Small Matrices (Computing Determinants of 2 by 2 and 3 by 3 Matrices)
Determinants by Expansion.
Determinants Using Row Operations
An Application of Determinants
Span Part 1
Span Part 2
Closure
Linear Independence/Dependence
A Discussion about Eigenvalues and Eigenvectors
Computing Eigenvalues/Eigenvectors of a 2 by 2 Matrix
Computing Eigenvalues/Eigenvectors of a 3 by 3 Matrix
Computing Eigenvalues/Eigenvectors of a 3 by 3 Matrix (a second example)

http://mathdemos.gcsu.edu/mathdemos/matrix_algebra_demos/matrix_algebra_demos.html

Concept support areas:

Matrix algebra, types of matrices, linear combinations, row operations, echelon forms, systems of equations, matrix inverse, determinants, vector space properties, span, independence/dependence, inner product spaces, orthogonality, projections, linear transformations, eigen concepts, symmetric matrices & the principal axis theorem (We never get to SVD.)

LAB Applications:

Digraphs, Markov Chains, Leslie Population Models, Interpolation, Linear & Quadratic Regression, Cryptography, Linear Transformations from R^2 to R^2, Unit Balls of Vector Norms, Discrete Dynamical Systems, Accessibility of Cities in a Trade Route Network

References/Sources

David M. Bressoud, from MAA Online, *Launchings from the CUPM Curriculum Guide: Reform Fatigue,*
https://www.maa.org/columns/launchings/launchings_06_7.html
Carl C. Cowen, *On the Centrality of Linear Algebra in the Curriculum,*
http://www.maa.org/features/cowen.html

Roe Goodman, *Using Technology Offstage in Linear Algebra Courses*,
http://www.math.rutgers.edu/~goodman/pub/Matlab_talk.pdf

http://www.math.rutgers.edu/~goodman/pub/NExT_slides.pdf

Thomas R. Hagedorn, *Improving Linear Algebra Instruction Using WebWork and the ATLAST Problems,* Electronic Proceedings of the Sixteenth Annual International Conference on Technology in Collegiate Mathematics Chicago, Illinois, October 30-November 2, 2003,
http://archives.math.utk.edu/ICTCM/i/16/S134.html

Bernard Kolman *and* **David R. Hill**, *Introductory Linear Algebra (An Applied First Course)* 8e, Prentice Hall, 2005

David R. Hill *and* **David E. Zitarelli***, Linear Algebra Labs with MATLAB, 3e*
Prentice Hall, 2004