Teaching an Introductory Linear Algebra Course

Rob Beezer
beezer@pugetsound.edu

Department of Mathematics and Computer Science
University of Puget Sound

Project NExT @ MathFest
August 1, 2012

Experience and Background

- U of Puget Sound == small liberal arts college (no engineers)
- Sophomore-level **mathematics** course, also our “transition” course
- Taught this course moe than 30 times
- Author: open-source First Course in Linear Algebra
- Contributor to open source system, Sage
- Lead PI, NSF grant: UTMOST
  Combine teaching, open textbooks and open software
  (linear algebra, abstract algebra)

"Common Learning Difficulties"

- Language
  - "A matrix has infinitely many solutions."
  - "A system of equations is nonsingular."
  - "The null space is empty." (instead of "trivial")
  - "The spanning set has dimension 4." (instead of "cardinality 4")
  - Banned vocabulary: "it" (pronouns), "it works," "thing" (and variants)
  - Question: "If we add something to it, will it still work?"
- Are your vectors columns, rows, points, arrows?
  - Someplace you confront row operations versus a column space
- Span of a **finite** set is an **infinite** set
- Decompositions ("can be expressed/written as")
  - Totally contrary to high school emphasis on **collecting** terms
  - Partial fractions may be their only similar experience
- Confusing the converse of a theorem
- The (well-known) leap to abstract vector spaces

Use of Technology

- Computation matters
  - In practice, i.e. for numerical linear algebra, it is central
  - For introductory course an exact approach is better (e.g. over rationals)
- Exact linear algebra
  - Reduced row-echelon form
  - Eigenvalues via roots of polynomials
  - Sage has the "field of algebraic numbers" implemented
- I want my students to understand use of:
  - primitives: eigenvectors from null space of $A - \lambda I$
  - high-level routines: `A.eigenvectors()`
- Testing understanding:
  - Sage on laptops during exams (library loaners, notebook server)
  - Largish matrices (5×7) cut/paste off "hidden" web page
  - Sage cell server in future?

Technology Demonstration

- Sage Cell Server
  - Web page text-box communicating with a running Sage server
  - Extremely simple to add to any web page ($\sim$ 4 lines of Javascript)
  - Interactive “interacts”, pre-loaded commands, wide-open practice area
  - Demonstration on UTMOST home page
- First Course in Linear Algebra
  - Heavy use of knowls
  - Sage examples implemented using Sage cell server (almost there!)
  - Rough cut at: Version 3.00 Preview

utmost.aimath.org

links to high-quality open-source textbooks. This initiative is supported in part by UTMOST.

Sage-Enhanced Textbooks

Rob Beezer announced his work on enhancing his linear algebra textbook and also announced his work on enhancing Judson's abstract algebra textbook. Both of these projects involved work sponsored by UTMOST.

Embedding Sage in a webpage (beta)

You can now embed Sage into any webpage! A beta version of the Sage Cell server was released. See the documentation for embedding a computation.

As an example, click the button below to explore a Taylor polynomial

Explore Taylor polynomials

or generate graph paper (including a pdf)

Make graphing paper

or try whatever Sage computation you want below.

```
factorial(30) # edit me
```

Syntax Highlighting

Evaluate

More news...

Contact Us

utmost.aimath.org

links to high-quality open-source textbooks. This initiative is supported in part by UTMOST.

Sage-Enhanced Textbooks

Rob Beezer announced his work on enhancing his linear algebra textbook and also announced his work on enhancing Judson's abstract algebra textbook. Both of these projects involved work sponsored by UTMOST.

Embedding Sage in a webpage (beta)

You can now embed Sage into any webpage! A beta version of the Sage Cell server was released. See the documentation for embedding a computation.

As an example, click the button below to explore a Taylor polynomial

Explore Taylor polynomials

or generate graph paper (including a pdf)

Make graphing paper

or try whatever Sage computation you want below.

```
A = matrix(QQ, [[ 204,   98, -26, -10],
                [-280, -134,  36,  14],
                [ 716,  348, -90, -36],
                [-472, -232,  60,  28]])
A.eigenvalues()
```

Syntax Highlighting

Evaluate

Permalink, Shortened Temporary Link

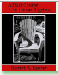
```
[4, 0, 2, 2]
```

Powered by Sage

More news...

Contact Us

file:///home/rob/books/fcla/conversion/htmlout/section-RREF.html



Reduced Row-Echelon Form

After solving a few systems of equations, you will recognize that it doesn't matter so much *what* we call our variables, as opposed to what numbers act as their coefficients. A system in the variables $x_1,\ x_2,\ x_3$ would behave the same if we changed the names of the variables to $a,\ b,\ c$ and kept all the constants the same and in the same places. In this section, we will isolate the key bits of information about a system of equations into something called a matrix, and then use this matrix to systematically solve the equations. Along the way we will obtain one of our most important and useful computational tools.

Subsection MVNSE Matrix and Vector Notation for Systems of Equations

Definition M Matrix

An $m\times n$ **matrix** is a rectangular layout of numbers from $\mathbb{C}$ having m rows and n columns. We will use upper-case Latin letters from the start of the alphabet $(A,\ B,\ C,\dots)$ to denote matrices and squared-off brackets to delimit the layout. Many use large parentheses instead of brackets — the distinction is not important. Rows of a matrix will be referenced starting at the top and working down (i.e.\ row 1 is at the top) and columns will be referenced starting from the left (i.e.\ column 1 is at the left). For a matrix A, the notation $\left[A\right]_{ij}$ will refer to the complex number in row i and column j of A.

□

Be careful with this notation for individual entries, since it is easy to think that $\left[A\right]_{ij}$ refers to the *whole* matrix. It does not. It is just a *number*, but is a convenient way to talk about the individual entries simultaneously. This notation will get a heavy workout once we get to Chapter M.

Example AM **A matrix**

Sage M **Matrices**

When we do equation operations on system of equations, the names of the variables really aren't very important. $x_1,\ x_2,\ x_3$, or $a,\ b,\ c$, or $x,\ y,\ z$, it really doesn't matter. In this subsection we will describe some notation that will make it easier to describe linear systems, solve the systems and describe the solution sets. Here is a list of definitions, laden with notation.

Definition CV Column Vector

A **column vector** of **size** m is an ordered list of m numbers, which is written in order vertically, starting at the top and proceeding to the bottom. At times, we will refer to a column vector as simply a **vector**. Column vectors will be written in bold, usually with lower case Latin letter from the end of the alphabet such as $\mathbf{u}$, $\mathbf{v}$, $\mathbf{w}$, $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$. Some books like to write vectors with arrows, such as $\vec{u}$. Writing by hand, some like to put arrows on top of the symbol, or a tilde underneath the symbol, as in $\underset{\sim}{u}$. To refer to the **entry** or **component** of vector $\mathbf{v}$ in location i of the list, we write $\left[\mathbf{v}\right]_i$.

file:///home/rob/books/fcla/conversion/htmlout/section-RREF.html

Once we have constructed a matrix, we can learn a lot about it (such as its parent). Sage is largely object-oriented, which means many commands apply to an object by using the “dot” notation. `A.parent()` is an example of this syntax, while the constructor `matrix([[1, 2, 3], [4, 5, 6]])` is an exception. Here are a few examples, followed by some explanation:

```
A = matrix(QQ, 2, 3, [[1,2,3],[4,5,6]])
A.nrows(), A.ncols()
```

```
(2, 3)
```

```
A.base_ring()
```

```
Rational Field
```

```
A[1,1]
```

```
5
```

```
A[1,2]
```

```
6
```

The number of rows and the number of columns should be apparent, `.base_ring()` gives the number system for the entries, as included in the information provided by `.parent()`.

Computer scientists and computer languages prefer to begin counting from zero, while mathematicians and written mathematics prefer to begin counting at one. Sage and this text are no exception. It takes some getting used to, but the reasons for counting from zero in computer programs soon becomes very obvious. Counting from one in mathematics is historical, and unlikely to change anytime soon. So above, the two rows of `A` are numbered 0 and 1, while the columns are numbered 0, 1 and 2. So `A[1,2]` refers to the entry of `A` in the second row and the third column, i.e.\ `6`.

There is much more to say about how Sage works with matrices, but this is already a lot to digest. Use the space below to create some matrices (different ways) and examine them and their properties (size, entries, number system, parent).

```
# Sage practice cell
# Enter new commands, or cut/paste examples
```

Syntax Highlighting

Evaluate

./knowls/sage.M.knowl

When we do equation operations on system of equations, the names of the variables really aren't very important. x_1, x_2, x_3, or a, b, c, or x, y, z, it really doesn't matter. In this subsection we will describe some notation that will make it easier to describe linear systems, solve the systems and describe the solution sets. Here is a list of definitions, laden with notation.

Web: http://buzzard.ups.edu/talks.html

Blog: http://beezers.org/blog/bb