Week 0

# Getting Started

## 0.1 Opening Remarks

### 0.1.1 Welcome to LAFF


☛ View at edX

Welcome to UT.5.02x Linear Algebra: Foundations to Frontiers (LAFF), a Massive Open Online Course offered on the edX platform.

This course is not only designed to teach the standard topics in a typical linear algebra course, but also investigates how to translate theory into algorithms. Like typical linear algebra courses, we will often start studying operations with small matrices. In practice, however, one often wants to perform operations with large matrices so we generalize the techniques to formulate practical algorithms and their implementations. To understand how to create software for solving matrix problems, we must thoroughly understand the underlying theory of linear algebra. Upon completion, you will grasp basic linear algebra concepts and get a glimpse of cutting edge research on the development of high-performance linear algebra libraries, which are used throughout computational science.

**What is LAFF?** LAFF is an online course on linear algebra that mirrors an undergraduate course taught by the authors through the Department of Statistics and Data Sciences.

**Why is linear algebra important?** Linear algebra is generally considered as important a tool for science (including the social sciences) as is calculus. There is an interesting post on reddit.com that started with the question "What is the point of linear algebra?". It generated a flood of comments that are worth reading.

**What level does LAFF target?** At a typical university, a number of different linear algebra courses are often offered: Introductory Linear Algebra, Applications of Linear Algebra, and Numerical Methods (which often includes other topics). LAFF corresponds to the first, introductory course and prepares you for subsequent courses. We added optional "enrichments" that also expose you to some of the frontiers of the subject.

**Who is the audience for LAFF?** Judging by who completed LAFF last year (in Spring 2014), people with very different kinds of background found the course useful. Participants included novice high school students as well as PhDs with decades of experience, and every level of education in between. You may want to check out reviews of the course on CourseTalk to see what former participants are saying.

**How and what will I learn?** Through: short videos, exercises, visualizations, and programming assignments you will study standard topics in an introductory linear algebra course.

Linear algebra deals with functions of many variables. These many variables are viewed as "vectors" of numbers. Thus, we start by discussing vectors. The functions we focus on have special properties: they are "linear transformations", which are extensions of the linear functions you studied in high school algebra. In Week 2 we define linear transformations as well as investigate and prove properties of these functions. This then allows us to link linear transformations to their representation as matrices. Next, you discover why multiplying a matrix times a vector or times a matrix is defined the way you may have been taught in high school. The reason comes from how matrices represent linear transformations. In the first third of the course you will also learn how to slice and dice matrices into pieces. This extends your concrete knowledge about operations with small matrices to operations with matrices of any size.

Solving systems of linear equations is a core topic in linear algebra. By the time you encounter this topic, in Week 6, you will be very comfortable with matrices and vectors. You build on this as you discover practical methods for finding solutions.

In the remainder of the course, you study how to solve linear systems with more or fewer equations than unknowns and find that there may be one, many, or no solutions. If there is no solution, what is the best approximate solution? The course wraps up with eigenvalues and eigenvectors.

All along, you not only learn *what* (the methods), but also *why* (the theory that underlies the methods).

**Will we learn how to program?** This is not a course that focuses on teaching you how to program for Matlab or in any other programming language. We teach just enough about how to program with M-script to support what we want you to learn about linear algebra. We link abstractions in mathematics to abstractions in code. That is not the same as teaching you how to program.

**Will we see applications?** LAFF tries to give you the background so that you can understand the application of linear algebra rather than focusing on applications themselves. We do use a few simple applications to motivate. We will point you to applications of linear algebra in some of the "enrichment" sections. We encourage you to share applications that have caught you interest on the discussion board.

This spring, there are at least two MOOCs offered that do focus on the application of linear algebra:

- Applications of Linear Algebra by Tim Chartier (Davidson College) on edX.
- Coding the Matrix: Linear Algebra through Computer Science Applications by Philip Klein (Brown University) on Coursera.

We suspect that both courses require an introductory course in linear algebra. Thus, LAFF may prepare you for those "application of linear algebra" courses.

## Wish You Were Here


☛ View at edX

Since most of you are not The University of Texas at Austin students, we thought we'd give you a tour of our new building: the Gates Dell Complex. Want to see more of The University of Texas at Austin? Take the ☛ Virtual Campus Tour.

### 0.1.2 Outline

Following the "opener" we give the outline for the week:

**0.1. Opening Remarks . . . 1**
0.1.1. Welcome to LAFF . . . 1
0.1.2. Outline . . . 4
0.1.3. What You Will Learn . . . 5
**0.2. How to LAFF . . . 6**
0.2.1. When to LAFF . . . 6
0.2.2. How to Navigate LAFF . . . 6
0.2.3. Homework and LAFF . . . 6
0.2.4. Grading and LAFF . . . 7
0.2.5. Programming and LAFF . . . 8
0.2.6. Proving and LAFF . . . 8
0.2.7. Setting Up to LAFF . . . 9
**0.3. Software to LAFF . . . 10**
0.3.1. Why MATLAB . . . 10
0.3.2. Installing MATLAB . . . 10
0.3.3. MATLAB Basics . . . 10
**0.4. Enrichments . . . 11**
0.4.1. The Origins of MATLAB . . . 11
**0.5. Wrap Up . . . 11**
0.5.1. Additional Homework . . . 11
0.5.2. Summary . . . 11

### 0.1.3 What You Will Learn

The third unit of the week informs you of what you will learn. This describes the knowledge and skills that you can expect to acquire. In addition, this provides an opportunity for you to self-assess upon completion of the week.

Upon completion of this week, you should be able to

- Navigate through LAFF on the edX platform.
- Keep track of your homework and progress through LAFF.
- Download and start MATLAB.
- Recognize the structure of a typical week.

## 0.2 How to LAFF

### 0.2.1 When to LAFF

The beauty of an online course is that you get to study when you want, where you want. Still, deadlines tend to keep people moving forward. To strike a balance between flexibility and structure, we release the material one week at a time and give a generous yet finite period during which to complete the homeworks.

The course schedule can be found at

https://courses.edx.org/courses/UTAustinX/UT.5.02x/1T2015/3b39b7df8cc543259c734c4f1d3f3ed0/

(This is the page to which the "Calendar" tab in the edX navigation bar links.)

In theory, all times should be automatically converted from UTC 23:00 to your local time. Please reference this schedule often as any official changes will appear here.

The highlights:

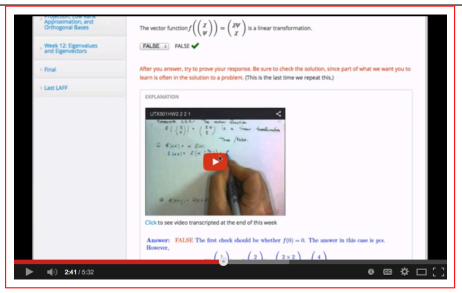
- Each Week is posted on a Wednesday, at UTC 23:00.
- Assignments for a week are due on a Monday, approximately 19 days from its posting, at UTC 23:00.

In the unit corresponding to this section on the edX platform, we give a link for iCal/ics format for those who wish to use their own calendar software.

### 0.2.2 How to Navigate LAFF


☛ View at edX

### 0.2.3 Homework and LAFF



☛ View at edX

When future weeks become available, you will notice that homework appears both in the notes and in the various units on the edX platform. Most of the time, the questions will match exactly but sometimes they will be worded slightly differently.

Realize that the edX platform is ever evolving and that at some point we had to make a decision about what features we would embrace and what features did not fit our format so well. As a result, homework problems have frequently been (re)phrased in a way that fits both the platform and our course.

Some things you will notice:

- "Open" questions in the text are sometimes rephrased as multiple choice in the units.
- Video answers appear as embedded YouTube, with then a link to the end of the week where the same video, with captioning and optional download from an alternative source, can be found. This was because edXs video player could not (yet) be embedded in answers.

Please be patient with some of these decisions. Our course and the edX platform are both evolving, and sometimes we had to improvise.

### 0.2.4 Grading and LAFF

How to grade the course was another decision that required compromise. Our fundamental assumption is that you are taking this course because you want to learn the material, and that the homework and exams are mostly there to help you along. For this reason, for the homework, we

- Give you multiple chances to get an answer right;
- Provide you with detailed answers; and
- Allow you to view the answer if you believe it will help you master the material efficiently.

In other words, you get to use the homework in whatever way helps you learn best.

Don't forget to click on "Check" or you don't get credit for the exercise!

How your progress is measured is another interesting compromise. The homework for each week is worth 5% of the total points in the course. There are 12 graded weeks, and hence this adds up to 60%. Now, within each week, limitations of the edX platform did not match with how we wanted to present the course. We very much wanted the homework close to the material so that the homework helps you along. This means that homeworks are scattered throughout the units and subsections. But the edX grade book starts with 100 points, and allows only integers to be assigned to subsections within weeks, unless subsections are equally weighted...

Let us explain how this now in practice works, using Week 1 as an example.

- Homework from Week 1 is worth 5 points towards the 100 point total.
- Week 1 has 6 graded subsections (1.2, 1.3, 1.4, 1.5, 1.6, and 1.8). Some of these subsections have a lot of homework problems, others have a few. Still, each subsection is worth the same. So, if you get all homework points in Subsection 1.2, then that gives you 1/6 x 5 = 5/6 points towards the total 100 points for the course. Subsection 1.3, which has a different number of homework problems, also gives you 1/6 x 5 = 5/6 points towards the total 100 points for the course.

Not much we can do about it without totally reformatting the course.

To view your progress, click on "Progress" in the edX navigation bar. If you find out that you missed a homework, scroll down the page, and you will be able to identify where to go to fix it. Don't be shy about correcting a missed answer. The primary goal is to learn.

Some of you will be disappointed that the course is not more rigorously graded, thereby (to you) diminishing the value of a certificate. The fact is that MOOCs are still evolving. People are experimenting with how to make them serve different audiences. In our case, we decided to focus on quality material first, with the assumption that for most participants the primary goal for taking the course is to learn.

Let's focus on what works, and be patient with what doesn't!

### 0.2.5 Programming and LAFF

In this course, we invite you to learn the theory of linear algebra hand-in-hand with the practice of developing a software library.

Programming is about abstracting. It will help us extend our concrete knowledge of how matrix operations work with small sized matrices to any size matrices. We encourage you, as you engage in LAFF, to take an active part in the abstraction process by extending what you know and thinking in general terms to construct algorithms and think about their costs.

We will be using the MATLAB® tool. In Week 0 we include instructions on how to set up your environment to use these tools. You do not need any previous programming knowledge or experience. You do not need to know M-script (the programming language for MATLAB®) nor is the purpose of this course to teach you MATLAB® and M-script. We will use this language in a very targeted way so that you master just enough of it to be able to use it for our purposes. In the beginning, we will completely talk you through the package construction. Later we will provide program skeletons and you will be asked to use your knowledge about the slicing and dicing of matrices for performing the linear algebra operations to fill in commands. We hope that you will come to appreciate, understand, and, PRODUCE components of a layered library.

We will share our own implementation of this library so you can build implementations of more complex operations. Please get into the habit of trying on your own before peeking at our solutions. If you encounter any implementation issues try conferring with others on the discussion boards.

In no time, you will be experiencing the frontier of linear algebra library development. Our FLAME research group prides itself on writing the most beautiful and among the highest performing code for many linear algebra operations. We will share this brilliance with you. If you dont agree you can laugh at us otherwise LAFF with us!

### 0.2.6 Proving and LAFF

Traditionally, Linear Algebra is a course that develops ones ability to prove mathematical facts. In this course, we invite you to expand upon your reasoning skills.

Proofs are first and foremost persuasive arguments. They help us connect, justify, and communicate our ideas. We encourage you, as you engage in LAFF, to question your developing intuitions and take an active part in the abstraction process by extending what you know and thinking in general terms.

In the beginning, communicating your ideas in your own words to convince yourself is valuable. However, to convince others and reveal your thought processes, making your arguments more formal is beneficial. We hope that you will come to appreciate, understand, and, YES, produce proofs.

Throughout this course, you will be asked to think a little deeper, extending your knowledge of properties of number systems to new structures that we encounter. Our Always/Sometimes/Never as well as True/False exercises are designed to do this. In addition to your answer, we urge you to think first, and then write a convincing argument explaining why you selected this answer. We will share with you, in text and video, our own formal proofs. Please get into the habit of trying on your own before peeking. Even if your proof differs from ours, remember that there is often more than one way to prove a result. You may want to discuss your proof with others on the discussion boards.

### 0.2.7 Setting Up to LAFF

It helps if we all set up our environment in a consistent fashion. The easiest way to accomplish this is to download the file LAFFSpring2015.zip and to "unzip" this in a convenient place. We suggest that you put it either in your home directory or on your desktop.

Once you unzip the file, you will find a directory LAFFSpring2015, with subdirectories. I did this in my home directory, yielding the directory structure in Figure 1.

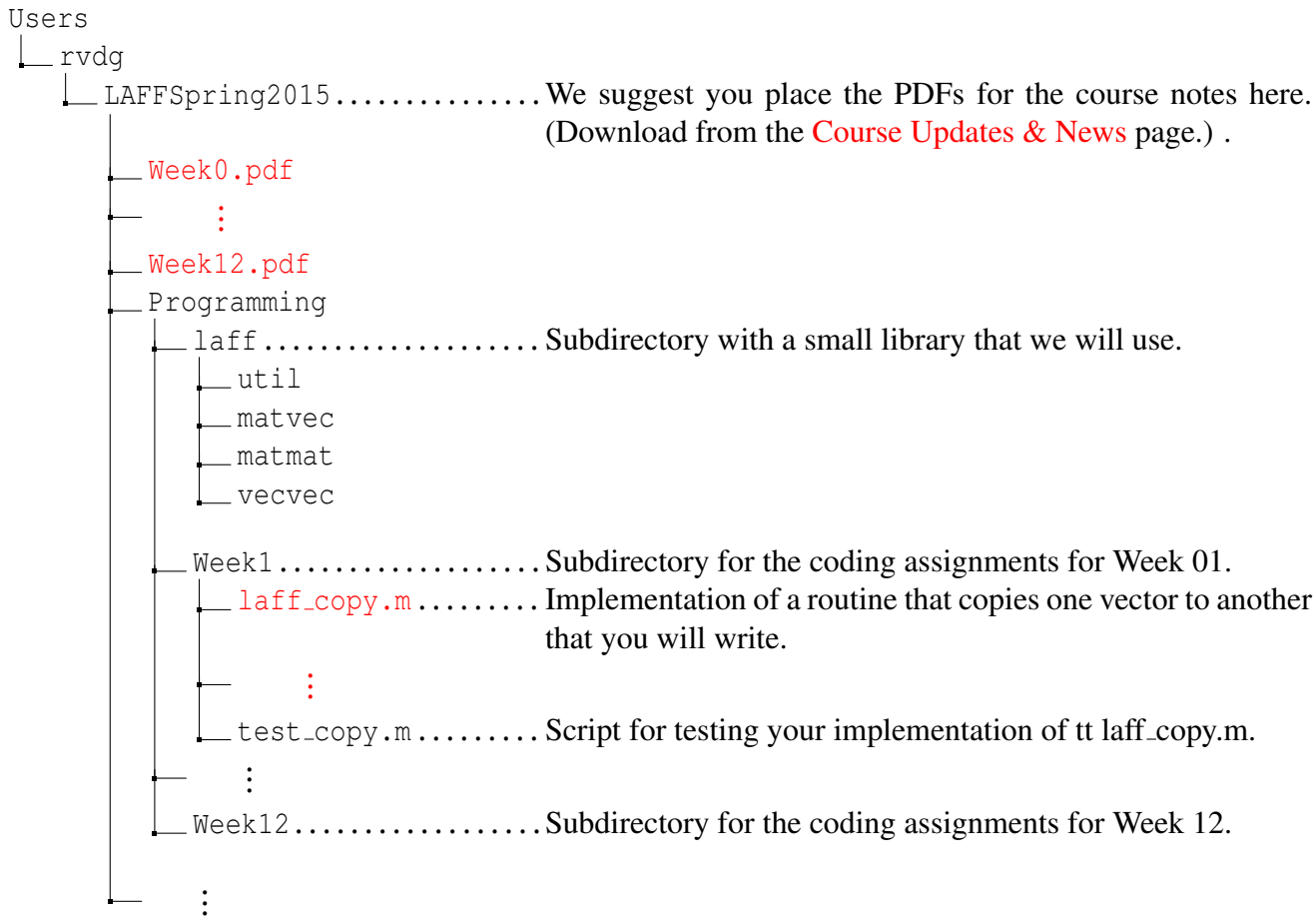



Figure 1: Directory structure for your LAFF materials. Items in red will be placed into the materials by you. In this example, I placed `LAFFSpring2015` in my home directory `Users -> rvdg`. You may want to place it on your account's "Desktop" instead.

## 0.3 Software to LAFF

### 0.3.1 Why MATLAB

We use MATLAB® as a tool because it was invented to support learning about matrix computations. You will find that the syntax of the language used by Matlab, called M-script, very closely resembles the mathematical expressions in linear algebra.

### 0.3.2 Installing MATLAB

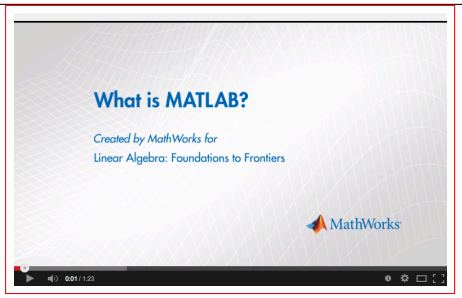
For information on how to install MATLAB for the course, visit Unit 0.3.2 on the edX platform.

### 0.3.3 MATLAB Basics

Below you find a few short videos that introduce you to MATLAB®. For a more comprehensive tutorial, you may want to visit MATLAB Tutorials at MathWorks and clicking "Launch Tutorial".

HOWEVER, you need very little familiarity with MATLAB in order to learn what we want you to learn about how abstraction in mathematics is linked to abstraction in algorithms. So, you could just skip these tutorials altogether, and come back to them if you find you want to know more about MATLAB® and its programming language (M-script).

#### What is MATLAB®?




☛ View at edX

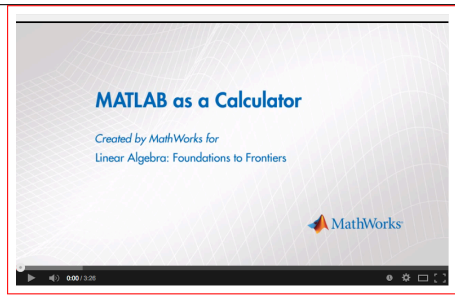
#### The MATLAB® Environment


☛ View at edX

### MATLAB®Variables


☛ View at edX

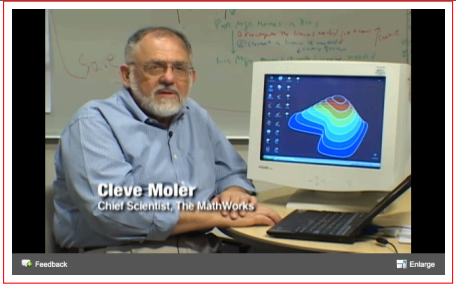
### MATLAB®as a Calculator




☛ View at edX

## 0.4 Enrichments

### 0.4.1 The Origins of MATLAB




☛ View at edX

## 0.5 Wrap Up

### 0.5.1 Additional Homework

For a typical week, additional assignments may be given in this unit.

### 0.5.2 Summary

You will see that we develop a lot of the theory behind the various topics in linear algebra via a sequence of homework exercises. At the end of each week, we summarize theorems and insights for easy reference.