Introduction to Numerical Methods and MATLAB Programming for Engineers

Todd Young and Martin J. Mohlenkamp
Department of Mathematics
Ohio University
Athens, OH 45701
youngt@ohio.edu

April 29, 2021

Thanks go to many students who have pointed out typos and mistakes. Thank you to Dr. Yaqin Feng for suggestions to improve the exercises.

Preface

These notes were developed by the first author in the process of teaching a course on applied numerical methods for Civil Engineering majors during 2002-2004 and was modified to include Mechanical Engineering in 2005. The materials have been periodically updated since then and underwent a major revision by the second author in 2006-2007.

The main goals of these lectures are to introduce concepts of numerical methods and introduce MATLAB in an Engineering framework. By this we do not mean that every problem is a "real life" engineering application, but more that the engineering way of thinking is emphasized throughout the discussion.

The philosophy of this book was formed over the course of many years. My father was a Civil Engineer and surveyor, and he introduced me to engineering ideas from an early age. At the University of Kentucky I took most of the basic Engineering courses while getting a Bachelor's degree in Mathematics. Immediately afterward, I completed a M.S. degree in Engineering Mechanics at Kentucky.

While working on my Ph.D. in Mathematics at Georgia Tech I taught all of the introductory math courses for engineers. During my education, I observed that incorporation of computation in coursework had been extremely unfocused and poor. For instance during my college career I had to learn 8 different programming and markup languages on 4 different platforms plus numerous other software applications. There was almost no technical help provided in the courses and I wasted innumerable hours figuring out software on my own. A typical, but useless, inclusion of software has been (and still is in most calculus books) to set up a difficult 'applied' problem and then add the line "write a program to solve" or "use a computer algebra system to solve".

At Ohio University we have tried to take a much more disciplined and focused approach. The Russ College of Engineering and Technology decided that MATLAB should be the primary computational software for undergraduates. At about the same time members of the Department of Mathematics proposed an 1804 project to bring MATLAB into the calculus sequence and provide access to the program at nearly all computers on campus, including in the dorm rooms. The stated goal of this project was to make MATLAB the universal language for computation on campus. That project was approved and implemented in the 2001-2002 academic year.

In these lecture notes, instruction on using MATLAB is dispersed through the material on numerical methods. In these lectures details about how to use MATLAB are detailed (but not verbose) and explicit. To teach programming, students are usually given examples of working programs and are asked to make modifications.

The lectures are designed to be used in a computer classroom with students working MATLAB

examples during the lecture or with students reading the notes and working the examples after a brief introduction. At Ohio University we have had good success with this Lecture/Lab format.

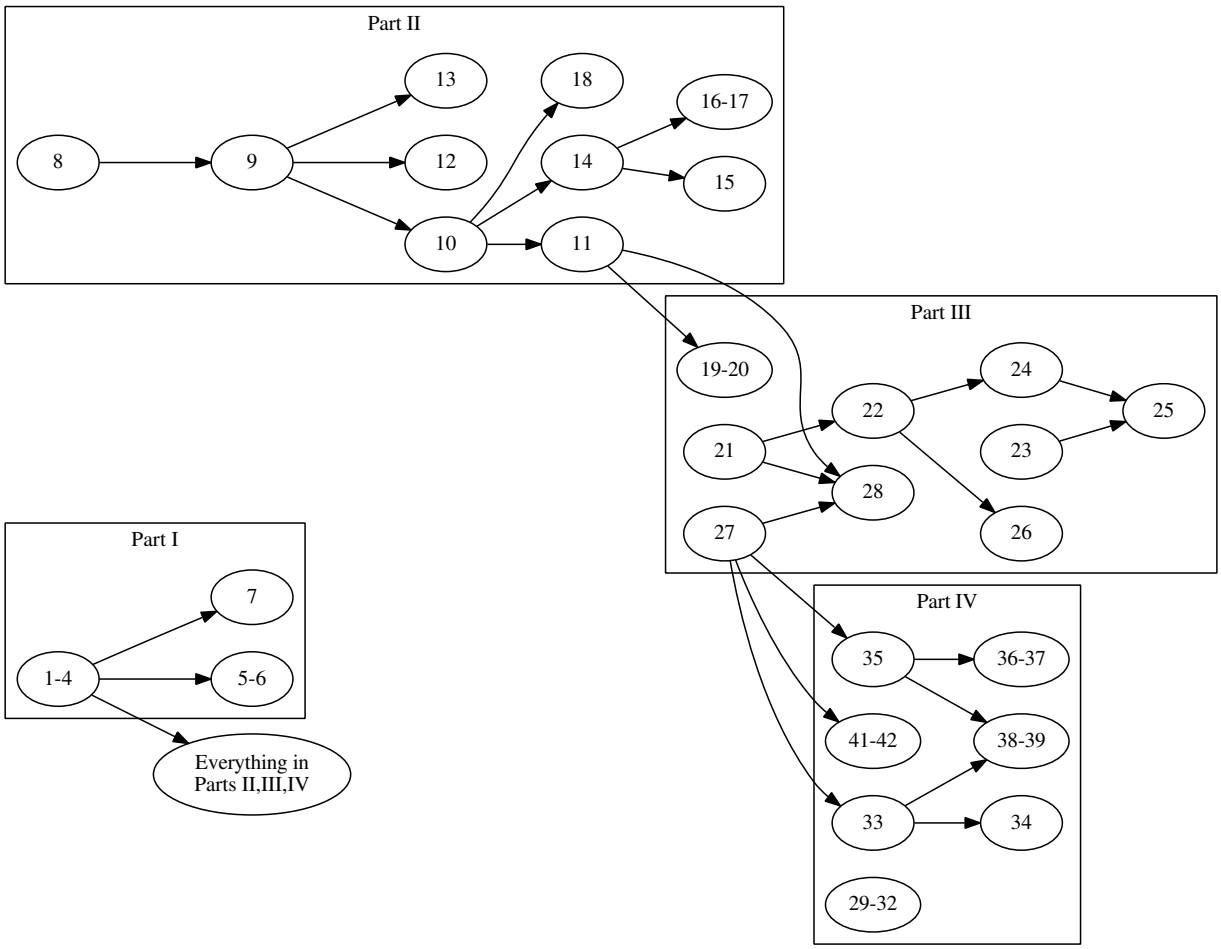
The lectures are divided into four Parts with a summary provided at the end of each Part. Typically we will given an exam covering each of Parts I, II, and III and a comprehensive final exam.

The exercises grow in complexity as the students build their programming skills. At Ohio University we ask students to complete the exercises in groups of 2-3 students and this accounts for a significant portion of the grade (e.g. 30%.).

Todd Young

Dependencies

Below we give the dependencies between Lectures. Almost everything depends on Lectures 1–4, so those links are omitted to reduce clutter. Some lectures, marked with * in the table of contents, have not yet been developed.



Contents