

Math 5620/6865: Numerical Analysis II

Varun Shankar

December 24, 2015

Administrivia

About me

I am Dr. Varun Shankar, currently a 3-year Assistant Professor in the Department of Mathematics. I got my PhD from the School of Computing here at the University of Utah. My webpage is <http://www.math.utah.edu/~vshankar/>. My research involves numerical analysis and the design of new numerical methods for solving PDEs on irregular domains and surfaces, with applications to biology (specifically, fluids and elasticity).

Meeting me

I sit in LCB 313. You can consult my webpage for a calendar schedule; you can swing by to meet me whenever I am free on the calendar. If you email me in advance, you can ensure I'm in my office when you swing by.

Class timings

This class meets MWF from 8-9:25 am in the William Browning Building, Rm 206 (WBB 206).

1 Introduction

Math 5620 (Numerical Analysis II) is a continuation of Math 5610. In 5610, you studied interpolation, numerical differentiation and integration, direct and iterative methods for linear systems, and possibly root-finding for nonlinear equations. 5620 will focus on the application of these techniques to the *numerical* solution of Ordinary Differential Equations (ODEs) and Partial Differential

Equations (PDEs). 5620 is *not* a specialized course on solving PDEs; there are more advanced graduate level courses that cover that. Instead, the goal of this class is to give you a broad overview of the principles and methods involving in solving ODEs or PDEs.

We will focus on derivation of most of these ODE and PDE methods from first principles. However, “first principles” is a loaded term. There are many underlying principles that these different numerical methods share. In this course, we will treat all these numerical methods as arising from interpolation, remarking on deviations where appropriate. While we will primarily focus on polynomial interpolation and its connection to ODEs and PDEs, we will also discuss a few other interpolation techniques that can be applied to generate numerical methods that are “cousins” to the more familiar polynomial-based methods. Specifically, we will discuss Fourier interpolation, and Radial Basis Function (RBF) interpolation; the former is a well-established field, while the latter is an exciting new field of research. Hopefully, as the semester progresses, it will become obvious to you as to how to use other interpolation/approximation techniques to generate numerical methods for ODEs and PDEs (splines, moving least squares, quasi-interpolation, least-squares approximation, etc.).

2 Textbooks and other material

This course has no formal text. However, for certain parts of the course, I will be following Uri Ascher’s textbook titled “Numerical Methods for Evolutionary Differential Equations” (<http://bookstore.siam.org/cs05/>). You are *not* required to buy this textbook. I will be supplying you with typeset lecture notes for every topic covered in class. The lecture notes will be posted in advance of each lecture. I will also be suggesting reference books and other material as we go along. These will be posted on the class webpage at <http://www.math.utah.edu/~vshankar/5620>.

3 Assignment structure

This course will be layered. We will start with polynomial interpolation, and steadily apply this very basic technique (in a very robust and stable form) to generate numerical methods for ODEs, and then PDEs. Your assignments will reflect this layering. Assignments will be centered on coding in Matlab or Octave. You will start by writing a robust polynomial interpolation code, and continue using that same code over the semester to solve ODEs and PDEs. There will be no exams; instead, this course will have a final project, possibly with project presentations.

You will turn in your assignments via email. Every handin must contain a PDF

and the code that you wrote; any code you turn in should run and produce meaningful output per the assignment guidelines. The PDFs will contain your word answers to the non-coding part of the assignments, and also instructions on how to run your code. We will have an assignment once every couple of weeks, and you will know the assignment in advance of the material to be covered.

Students signed up for Math 6865 will have extra work for each assignment and for the final project. This will be indicated in each assignment.

4 Final Project

The final project will involve an application of what you have learned over the semester. You will pick an interesting PDE of your choice (or ask me about one), and then use your previously written code to generate a numerical method to solve that PDE on an interesting domain.

In general, if you have a question about what is an appropriate final project, ask!