Eigenvalues and random matrices

Jonathan Husson

Random matrices were incepted in the 1920s by the statistician John Wishart to study empirical covariances and then resurfaced in the 1950 when physicist Eugene Wigner used them to model heavy nuclei (for which he was awarded the Nobel Prize in 1963). Since then, the field has known an important expansion to a wide array of scientific fields from number theory to mathematical physics. Now, random matrices can be found in finance, in machine-learning and in the study of complex networks.

But what is a random matrix? Well, it is a matrix whose entries are chosen (*drum roll*) randomly. For instance a classical model of random matrices, *Wigner matrices* are a family of random $N \times N$ symetric matrices W_N that looks like this :

$$W_N = \frac{1}{\sqrt{N}} \begin{pmatrix} \sqrt{2}a_{1,1} & a_{1,2} & \cdots & a_{1,N} \\ a_{1,2} & \sqrt{2}a_{2,2} & \cdots & a_{2,N} \\ \vdots & \vdots & \ddots & \\ a_{1,N} & a_{2,N} & & \sqrt{2}a_{N,N} \end{pmatrix}$$

where all the entries $(a_{i,j})_{1 \leqslant i \leqslant j \leqslant N}$ are independent copies of the same random variable whose average is 0. In random matrix theory, we are often interested in the matricial properties of W_N (such as its eigenvalues, its eigenvectors or its determinant) when N becomes very large. For example, a question of particular interest is: how does the spectrum of such matrices globally behaves when N tends to infinity? It turns out for such a model, the asymptotic distribution of the eigenvalues is known (this result is called *Wigner's theorem*).

The goal of this project is to introduce its participants to the basics of the study of eigenvalues of random matrices. We will go through some classical random matrix models to illustrate some of the most celebrated results of the field. The approach will mix proofs and simulations depending on the students backgrounds and wishes. Then if time permits there are several directions the project may take, like the study of extremal eigenvalues of a perturbed random matrix or a foray into the world of free probability and its links to polynomials in random matrices.

Prerequisites:

- Some familiarity with probability theory (ideally 425: Introduction to probability or equivalent).
- 215 Multivariable and vector calculus or equivalent
- 217 Linear Algebra or equivalent
- Some coding experience (Matlab, Mathematica, C++, Python, etc...)

Exploring numerical schemes for conservation laws

Maria Han Veiga

February 2019

1 Abstract

The **shallow water equations** [1] describe a thin layer of fluid of constant density in hydrostatic balance, bounded from below by a bottom topography, and from above by a free surface. For example, the propagation of a tsunami can be described accurately by the shallow-water equations (until the wave approaches the shore).

The shallow water equations are a set of non-linear hyperbolic partial differential equations and there is no general closed form solution. Thus, in order to study and describe the solution at some time t, we must solve these equations numerically, through **numerical schemes** [2].

In this project, we will use a well-known finite element method to derive a numerical scheme to solve these equations. Furthermore, we will study the discretisation error of the numerical scheme, as well as strategies to mitigate this error. Namely, we will look in detail on the notion of *well-balanced* and *asymptotically preserving* numerical schemes, which retain analytical properties of the continuous model at the discrete level.

Pre-requisites:

- Programming skills (matlab or python or fortran)
- Multivariate calculus
- Linear Algebra
- Differential equations
- Interest in fluid dynamics

References

[1] R. B. Kellogg. The shallow water wave equations: Formulation, analysis and application (i. kinnmark). *SIAM Review*, 30(3):517–518, 1988.

[2] E. Godlewski and P.A. Raviart. *Numerical Approximation of Hyperbolic Systems of Conservation Laws.* Number n.^{o} 118 in Applied Mathematical Sciences. Springer, 1996.

Fibonacci Links

Martin Strauss

Fall 2022

Project Description

A thin, flexible, chain hangs in the shape of a catenary, the hyperbolic cosine, $\cosh(x) = (e^x + e^{-x})/2$, and the derivative is the hyperbolic sine, $\sinh(x) = (e^x - e^{-x})/2$. These functions are related to the Fibonacci numbers 0, 1, 1, 2, 3, 5, 8, and Lucas numbers, 2, 1, 3, 4, 7, 11, ..., defined by the recursion that makes each number the sum of the previous two numbers (with different starting conditions). For example, the eighth Fibonacci number is given by the horrendous-looking

$$\frac{2\sinh(8\ln(\phi))}{\sqrt{5}}$$

and similarly for other even-indexed Fibonacci numbers, gotten by replacing 8 with some other even number while leaving the structural constants 2, 5, and the golden ratio ϕ. Type into google! Try replacing sinh with cosh and/or removing the $\sqrt{5}$ to get odd-indexed Fibonaccis and odd- and even-indexed Lucases.

In this project, we'll study the hyperbolic trigonometric formulas, Fibonacci numbers, and Lucas numbers. We will build a "discrete catenary" of inflexible links attached to each other at pivots, so that the chain will naturally hang in a shape that (after normalization) miraculously goes through integer-valued coordinates given by Fibonacci and Lucas numbers.

Prerequisites

Math 217

RESEARCH PROJECTS

Dao Nguyen

This research topic involves developing the method of *finite difference (discrete) approximations* for optimization problems, implementing numerical algorithms, and applying them, especially, to the calculations in machine learning, deep reinforcement learning, artificial intelligence, and their applications. Below, I briefly discuss some of the main ideas and sketch some applications for the future research.

1 Model Predictive Control

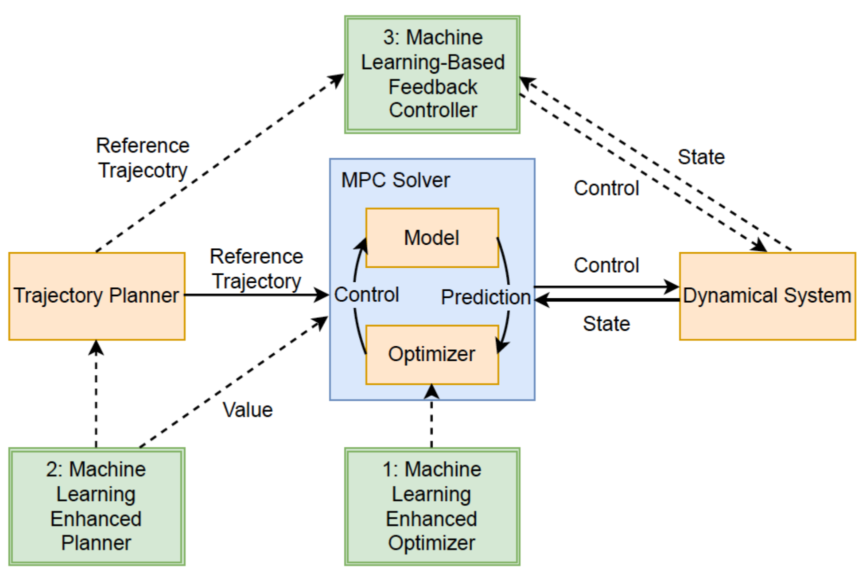
Solving complex optimal control problems have confronted computational challenges for a long time. Recent advances in machine learning have provided us with new opportunities to address these challenges. This project takes model predictive control, a popular optimal control method, as the primary example to survey recent progress that leverages machine learning techniques to empower optimal control solvers. We also discuss some of the main challenges encountered when applying machine learning to develop more robust optimal control algorithms.

Our approach is based on developing the method of discrete approximations. We derive the necessary optimality conditions of the discrete Euler-Lagrange type and the *numerical algorithms* to compute the optimal solution to the optimization and optimal control problems in robotics. Our further research goals concerning this model include developing efficient numerical algorithms (and coding in Python and Matlab) to solve the optimal control problems for them with large numbers of robotics in the corresponding models. It could be done, in particular, by using an appropriate discretization and employing numerical algorithms of finite-dimensional optimization to the discrete-time problems obtained in this way.

In this project we formulate the *optimal control problem* of type (P) that can be treated as a continuous-time counterpart of the discrete algorithm. Consider the cost functional

$$\text{minimize} \quad J[x,u] := \frac{1}{2}\left\|x(T)\right\|^2, \tag{1.1}$$

We describe the continuous-time dynamics by the controlled process and the dynamic noncollision condition $\|x^i(t) - x^j(t)\| \geq 2R$. Next, applying the necessary optimality conditions for the controlled model allows us to obtain the optimal solution for this model.



We derive a new *discrete approximation method* to obtain the necessary optimality conditions for optimal control problem and new *numerical algorithms* to find the optimal solution using the obtained conditions.

2 Applications

- Students can implement the code for the robot model in the case n is a large number using the necessary optimality conditions that we obtained. We propose to use the *discrete approximation method* to approximate solution of the optimal control problems. Students can try to construct a numerical scheme for some well-known robotic models. It is also worth comparing the rate of convergence between the new method and the traditional one.

- We also propose a machine learning enhanced algorithm for solving the optimal landing problem. Using Pontryagin's minimum principle, we derive a two-point boundary value problem for the landing problem. The proposed algorithm uses deep learning to predict the optimal landing time and a space-marching technique to provide good initial guesses for the boundary value problem solver. The performance of the proposed method is studied using the quadrotor example, a reasonably high dimensional and strongly nonlinear system. Drastic improvement in reliability and efficiency is observed.

Such those projects can benefit the students in multiple ways. First, they are introduced new areas related to optimization and optimal control. Second, some well-known models help them to engage mathematics in real-world issues. Finally, it would also give the student an opportunity to gain a new skill by learning a programming language such as Matlab and Python to implement the numerical methods needed to verify the approximation's accuracy.

3 Prerequisites

- Coding skills in Matlab or Python.
- Courses or Tests:
  - Calculus classes: I, II, III, IV/ Minimum Grade of B/ May not be taken concurrently.
  - Linear Algebra/ Minimum Grade of B/ May not be taken concurrently.
  - Real Analysis/ Functional Analysis (Optional).