VISUALIZING ALGORITHMS AND LARGE DATASETS

MARIA HAN VEIGA AND ZACH NORWOOD

Project description. A mathematical visualization becomes much more expressive when it is made *interactive* or *animated*. In this project, students will build on existing tools to create a framework to generate dynamic, interactive visualizations. Early on, we will focus on animated visualizations of standard algorithms, possibly including some algorithms that arise in machine learning. Graph algorithms, dynamic-programming algorithms, and algorithms that approximately solve NP-complete problems are some other examples of what we might try to animate.

Tools to extract qualitative, human-readable information from large datasets are more important than ever, since large datasets are so abundant. The visualization tools created during the first part of the project should be flexible enough that we can adapt them visualize and explore large datasets (for example, a dataset from Wikileaks such as the DNC email dump). This will be the focus of the second part of the project.

Prerequisites. Math 217 and some programming experience, preferably with Python. Familiarity with D3.js or javascript would be helpful, but isn't essential. Math 416 (Algorithms) or Math 498 (Machine Learning) would be helpful, but neither is essential.

Date: Winter 2022.

Computations of Steady Solutions in Rayleigh-Bénard Convection

Baole Wen

Introduction — The objective of this project is to utilize new advanced computational tools to discern the fundamental mechanisms governing the global transport of heat and momentum in an outstanding problem in fluid dynamics, i.e., Rayleigh-Bénard convection, the buoyancy-driven flow in a fluid layer heated from below and cooled from above. Specifically, we seek to elucidate the flow structure and heat transport properties of steady albeit *dynamically unstable* solutions in the strongly nonlinear regime. In order to push the computations to sufficiently large/small parameter regimes to discover the asymptotic transport behavior, **the participants are expected to modify the existing Matlab code to a C/C++ code.**

Background and significance — Convection is buoyancy-driven fluid flow resulting from density variations in the presence of a gravitational field. Beyond its role in myriad engineering applications, convection underlies many of nature's dynamical designs on larger-than-human scales, including the atmospheric and oceanic motions central to meteorology and climate science. A key feature of convection is heat transport, and predicting transport for large applied temperature gradients in the strongly nonlinear regime remains a major challenge for the field. Since the 1960s two distinct scaling theories have contended to quantitatively characterize the strongly nonlinear regime, yet no clear winner has emerged. Our recent investigations of steady roll solutions [1, 2] offer new evidence for one of these theories. Interestingly (and perhaps somewhat surprisingly), the preliminary work reveals that aspect-ratio-optimized steady roll solutions transport *more* heat than turbulent experiments or simulations at comparable parameters. This study has the potential to resolve the 60-year conundrum concerning asymptotic transport in turbulent convection.

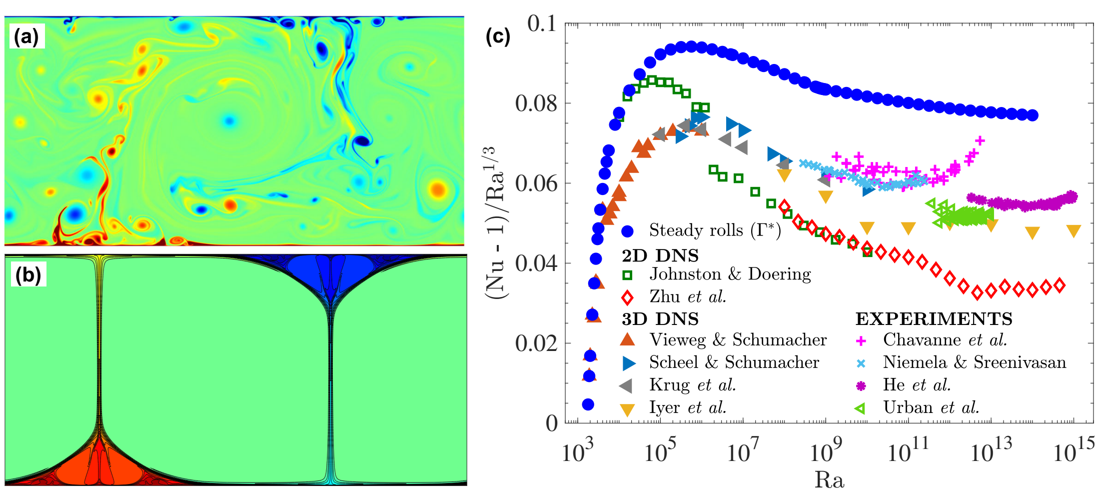

Figure 1: (a,b): Temperature fields for (a) direct numerical simulations of turbulent 2D convection and (b) the corresponding fully resolved unstable steady solution at similar parameters. Steady rolls comprise the backbone of turbulent convection. (c): Compensated heat flux for aspect-ratio-optimized steady rolls and turbulent 2D & 3D direct numerical simulations and experiments.

Prerequisites — Linear Algebra (Math 214) and Numerical Methods (Math 371 or 471 or 571). Coding skills in Matlab, C/C++ or Fortran.

References

[1] B. Wen, D. Goluskin, M. LeDuc, G. P. Chini, C. R. Doering. 2020 Steady Rayleigh–Bénard convection between stress-free boundaries, J. Fluid Mech. 905 (R4).

[2] B. Wen, D. Goluskin, C. R. Doering. 2021 Steady Rayleigh–Bénard convection between no-slip boundaries, in press in J. Fluid Mech. (arXiv:2008.08752).

Exploring Numerical Interpolation: Tracing Points, Numerical Stability, Applications and More

Advisor: Yabin Zhang and Hanliang Guo

LoG(M) Project Winter 2022

Let's start with a simple puzzle. Take a look at Figure (a) below. It contains a collection of points with numbering. Imagine that you are given a piece of paper with this figure printed on it and are asked to uncover a shape or pattern from these points. Your very first impulse might be to use a pencil to trace out lines or curves connecting these points following the given numbering. And then you may be able to recognize an interesting pattern from your drawing and realize that this collection of points is not entirely meaningless. Although this puzzle is quite easy to solve, your approach for tackling it is quite significant and universal. In real life applications, available data is often limited and only given for certain scenarios, like the points in the puzzle, and scientists want to use these data to predict or estimate for other more interesting scenarios of which no data is available, like tracing out lines or curves to fill in the gap between each pair of consecutive points in your puzzle solution. Mathematically, such estimates may be obtained via *numerical interpolation*, one of the major fundamental topics of numerical methods. The numerical interpolation family contains many different methods. And certain choices may be more suitable than others depending on the particular applications. Figure (b) (c) and (d) below illustrate three different "drawings" done by Matlab's three different built-in interpolation methods. As you can tell, the drawings look quite different from each other, and this is due to the different properties of the underlying interpolation methods.

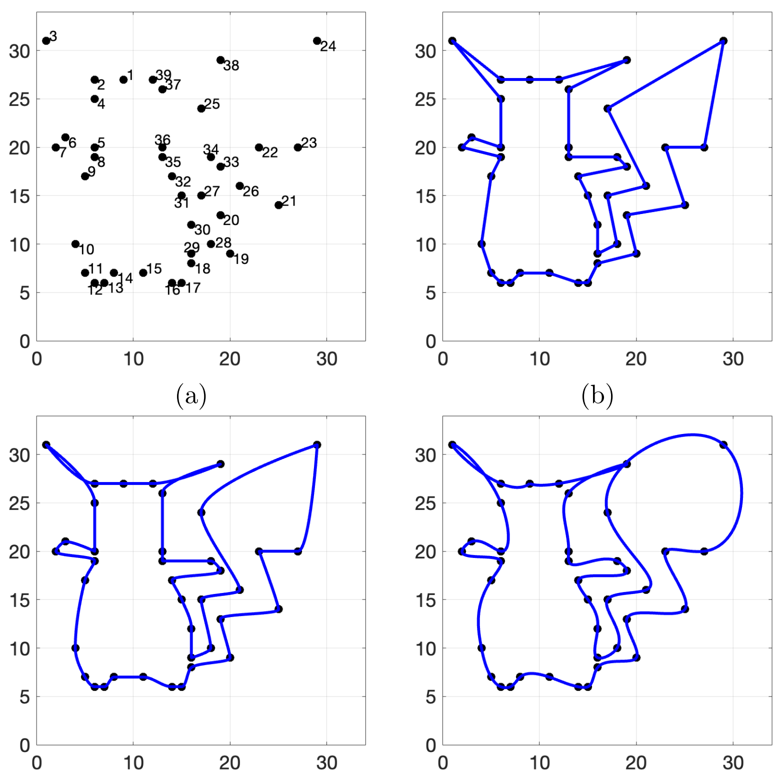

Figure 1: Three different ways to trace a collection of points with given location and order. Figure (b), (c) and (d) are generated via Matlab's built-in `interp1()` function with different choices of interpolation methods.

What to expect? In this project, we will explore various interpolation methods and some applications in three stages:

- Stage I: get to know them, implement them and understand their pros and cons. We will investigate the *cost*, *accuracy* and *stability* of the methods. (We will improve our implementations by breaking them.)
- Stage II: use the interpolation methods to build a Matlab app that connects a given collection of points on the plane satisfying users' special requests, e.g., closed curve with continuous first and second order derivatives.
- Bonus Stage: use the app to create geometry description for physical problems and solve them. Examples include two-dimensional *viscous flow* in confined geometry and *acoustic wave scattering*. We will formulate the problems as *boundary integral equations (BIEs)* and solve them. Partucularly, the app will convert a collection of data points to a suitable description for the *boundary* of the BIE.

What kind of skill set is needed to be on-board?

- Multi-variable calculus and linear algebra.
- Some experience in differential equations.
- Some experience in coding.
- Some experience in numerical methods will be a big plus but we will introduce concepts such as stability and go through definitions such as Lagrange interpolation in details.
- Curiosity and passion!

Temporal Dynamics of the Network of Mathematics Sub-disciplines

Have you ever wondered which came first, algebraic geometry or algebraic topology? Maybe you have wondered if dynamical systems or graph theory were more important for the development of network theory? Perhaps you want to know which decade category theory started to become mainstream?

Well in this Winter 2022 Log(M) project you could help answer these questions and more. Work with mathematics librarians Sam Hansen and explore the relationships and connections between different areas of mathematics by digging into who cited whom, when the citations took place, and how often. This work will utilize citation and classification information from multiple major bibliographic datasets, such as: Web of Science, zbMath, and Microsoft Academic. During this project you will become familiar with a number of different methods for analyzing very large data sets, how to apply many bibliometrics measures and techniques, and learn how to visualize networks that change over time. This is an exciting area of research into the nature of relationships within mathematics, an area that has had very little previous done in it. So, this is your chance to not only help mathematicians better understand their discipline but also to stamp your name as one of the first to ever explore the dynamics of mathematical knowledge over time.

Prerequisites: Coding experience

LOG(M) EXTREMAL VARIETIES
PROJECT DESCRIPTION

TIM RYAN AND JANET PAGE

One major element of the historical study of (algebraic) surfaces has been the study of the lines that lie on each surface. Over fields of characteristic 0, the situation is relatively well understood. Over fields of positive characteristic, there have been recent developments showing that the situation is very different. In this project, we will explore the geometry of certain surfaces with extremely large numbers of lines in positive characteristic.

To be more explicit, we will start with the case where our field has characteristic 0. Let $f \in \mathbb{R}[x, y, z]$ be a polynomial of degree d in three[1] variables. The points where f vanishes determines a surface S in $\mathbb{R}^3$, i.e.

$$S = \{\vec{x} \in \mathbb{R}^3 : f(\vec{x}) = 0\}.$$

One interesting feature of surfaces is the number of lines they contain. If $d > 3$, then for a general f, the surface S contains no lines. When S is smooth (and defined over a field of characteristic 0), the number of lines on S is bounded above by $11d^2 - 28d + 12$.

If we instead let our field be a field of positive characteristic p, (for example, think about $\mathbb{Z}/p\mathbb{Z}$), then there are surfaces that are known to wildly violate this bound. In particular, a recently studied class of surfaces, called *extremal surfaces*, are known to have $(d^2 - 3d + 3)d^2$ many lines. Extremal surfaces are very approachable as they can be defined in terms of linear algebra. These surfaces and their configurations of lines have deep connections to arithmetic geometry, combinatorics, and coding theory.

In this project, we will study the configurations of lines on extremal surfaces. More specifically, students will aim to classify the sizes of (maximal) sets of skew lines, i.e. the sizes of sets of lines where all of the lines are disjoint from one other. The team will use the algebra and geometry of the surfaces but may also use statistical techniques such as Markov Chain Monte Carlo methods to approach this problem. Time permitting, this project has several further directions. In parallel to this, we will building a code base for the current and future study of these surfaces.

Prerequisites: Math 412 or Math 493 or equivalent

[1]In reality, we will be working with four variables, because we will be working on something called *projective space*, and we sometimes prefer to think about about *algebraically closed* fields like $\mathbb{C}$ instead of $\mathbb{R}$.