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.

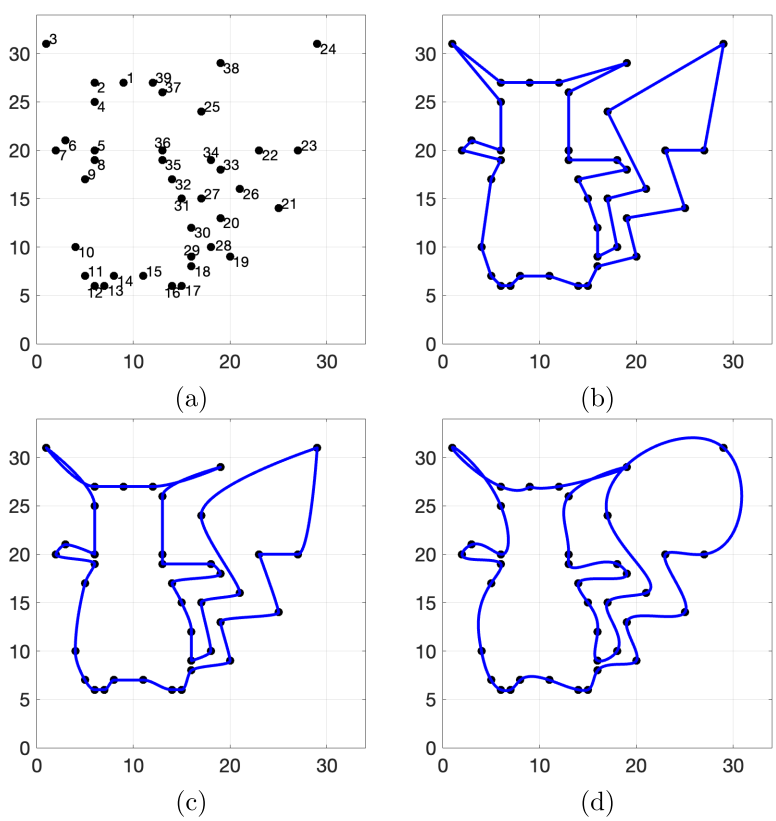

Figure 1: Three different ways to trace a collection of points with given location and order. Figure (b), (c) and (d) are generated via Matlab's built-in `interp1()` function with different choices of interpolation methods.

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!