Slide rules rule

Martin Strauss

Project Description: Mathematics explains physical phenomena, but, turned on their head, the same principles enable analog computers. We'll study and build our own linear, circular, and helical slide rules, which have long been used to multiply real numbers. Next, we'll proceed to slide rules that multiply complex numbers, which have also been produced commercially, though they don't fit in a pocket even with a heavy-duty pocket protector (`https://collection.maas.museum/object/598597`). Finally, as time, ambition, and eyeroll forbearance allow, we'll generalize to related devices to compose linear functions $x \mapsto mx + b$ over the reals or complexes and even more contrived contraptions to multiply two-by-two matrices over the complexes.

Making ordinary slide rules involves learning a plotting package like LaTeX/TikZ and using a black and white printer or, in a pinch, drawing by hand on graph paper. The more ambitious slide rules will involve some kind of fabrication, e.g., 3-d printing.

Multiplying two-by-two matrices over the complexes amounts to composing Moebius transformations. This is a beautiful theory with wider applications to complex analysis, geometry including linear perspective in art, mapmaking, and special relativity.

Prerequisites: Math 217

Modes of infinitely long strings

Jörn Zimmerling

Project Description: The oscillations of a string with variable density that is clamped on two ends is described by a differential equation well described by regular Sturm-Liouville theory. The differential operator of such a problem has real eigenvalues and eigenfunctions that are orthogonal and complete.

An infinitely long string no longer fits this theory, however, in certain cases the differential operator has complex eigenvalues with eigenfunctions "orthogonal" in a (non-positive definite) bilinear form. In the literature such eigenfunctions are known as quasi-normal modes.

In this project we try to expand solutions to the infinite string equations using these quasi-normal modes. We will study the connections between differential operators and their discretizations. The main goal is to develop and implement an algorithm that computes the solution to an infinite string with variable density on a bounded interval excited by an external force.

Prerequisites:
- Some form of Differential equations and Linear algebra
- Knowing what a complex number is
- Some form of programming
- (numerical methods for ODEs) is a big plus but we will go through everything in this project.

A Machine Learning Approach to Classify Microswimmers

Hanliang Guo

Background Biological and artificial microswimmers swim using various strategies. For example, bacteria *E. coli* swim by rotating a helical tail bundle at the back of the cell, algae cell *Chlamydomonas* swim by whipping two flagella in a breaststroke fashion, ciliated eukaryotic cell Paramecium swim by orchestrating the cilia carpet on the cell surface in a wave fashion. Despite the difference in the detailed locomotion strategies, if we take a step back and investigate where the activations (driving forces) concentrate, they can be classified into three main groups of microswimmers: *pushers* (activation in the back), *pullers* (activation in the front), and *neutral swimmers* (activation uniformly distributed). For a given axisymmetric shape, efficient numerical algorithms exist to optimize the slip velocity on the swimmer surface to maximize the swimming efficiency, and subsequently classify the microswimmers.

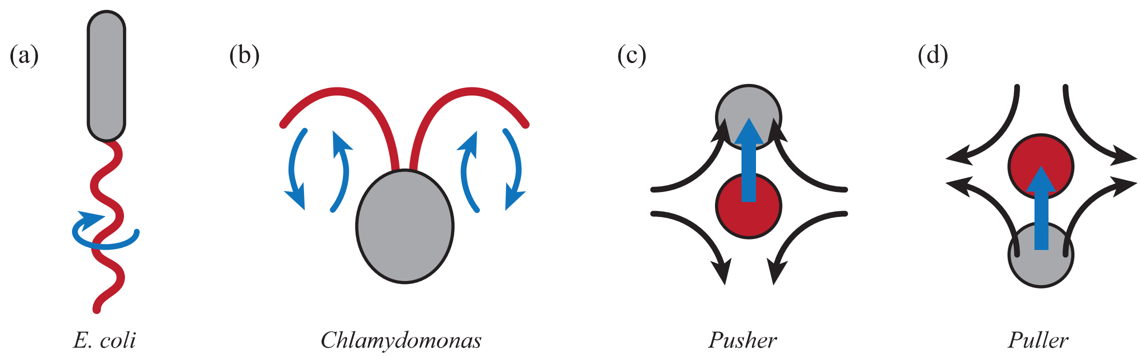


Figure 1: Examples of pusher and puller. Cartoons showing the swimming mechanisms of *E. coli* (a) and *Chlamydomonas* (b) and their reduced-order swimming types *pushers* (c) and *pullers* (d) respectively. The cell body (passive part) are depicted in grey and the flagella (active part) are depicted in red. The flagellar motion is denoted by the blue arrows.

Goals In this project, we will build a machine learning framework (most likely an artificial neural network) to classify the optimal microswimmers for a given shape, <u>without</u> actually optimizing them. In other words, we are trying to classify the optimal microswimmers based solely on their shapes. The more challenging second goal is to understand the classification results provided by the machine learning algorithm. That is, we want the machine learning algorithm to be interpretable to inform future design of microswimmers.

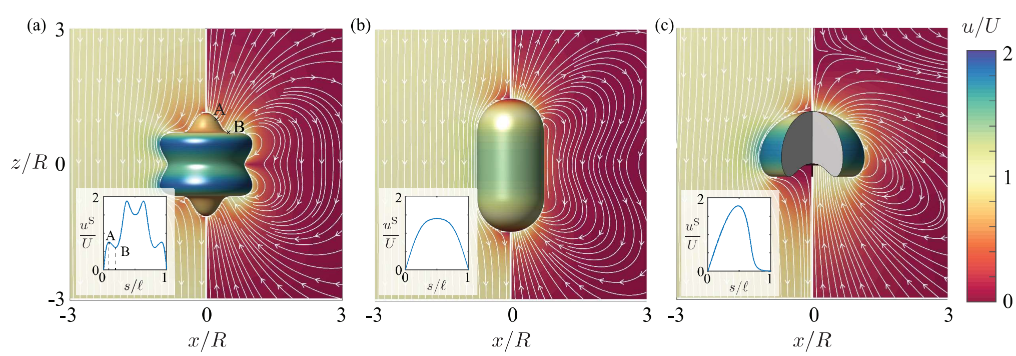


Figure 2: Optimal slip velocities of microswimmers with different shapes. (Adapted from [HG et al. JFM (2021)])

Prerequisite Linear algebra (Math 214 or equivalent) and Multivariable Calculus (Math 215 or equivalent). Coding experience (Python, Matlab) will always help.

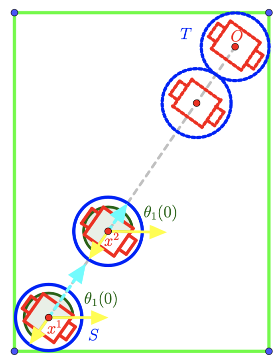
RESEARCH PROJECTS

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 results and sketch some ideas for the future research related to mobile robot models.

1 Optimization of Controlled Mobile Robot Model with Obstacles

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 and investigate an *optimal control* version of the *mobile robot model* with obstacles which dynamics is described as a sweeping process. We formulate the *sweeping optimal control problem* of type (P) that can be treated as a continuous-time counterpart of the discrete algorithm of the controlled mobile robot model by taking into account the model goal stated above. Consider the cost functional

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

which reflects model goal to *minimize the distance* of the robot from the admissible configuration set to the target. We describe the continuous-time dynamics by the controlled sweeping process and the dynamic noncollision condition $\|x^i(t) - x^j(t)\| \geq 2R$ amounts to the pointwise state constraints $x(t) \in C \iff \langle x_*^j, x(t)\rangle \leq c_j$ for all $t \in [0,T]$ and $j = 1,\ldots,n-1$, due to the construction of C and the normal cone definition. Next, applying the necessary optimality conditions for the sweeping controlled robot 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 for sweeping processes and new *numerical algorithms* to find the optimal solution using the obtained conditions. We implemented the code for the robot model in the case n is a large number using the necessary optimality conditions that we obtained.

2 Undergraduate Research Projects

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.

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).