PROBLEM SET 3: APPLIED MATHEMATICS 216

Due: Monday February 21, at 11:59pm

Goals for the week.

(1) Automatic differentiation with JAX

(2) Solving ODEs with JAX

(3) Inverting the laws of physics with JAX

Problems.

(1) **Lorenz equations**

The following system of ordinary differential equations is called the Lorenz system:

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y-x) \\ \frac{dy}{dt} &= x(\rho - z) - y \\ \frac{dz}{dt} &= xy - \beta z \end{aligned} \tag{1}$$

1. The role of the parameters and the initial conditions. To start with, use the code provided in the notebook to plot the solution of Eq. (1) for a given set of parameters ρ, σ and β and of initial conditions $(x(0), y(0), z(0))$. Then, try changing some of the parameters following the suggestions in the notebook.

(a) What happens when you change the initial conditions?
(b) Give an estimate of ρ^*, the value of ρ at which the solution changes regime.
(c) What happens when you change the initial conditions for a system with $\rho > \rho^*$?

2. Predict the parameters of the equations. Consider the two datasets “Lorenz_traj1.txt” and “Lorenz_traj2.txt”, which contain two trajectories that we generated using Eq. (1). You have seen in class and in section that you can in principle find all the unknown parameters, by using gradient descent with JAX. But there is a complication for the Lorenz equations: above a certain value of ρ, the solutions to the equations become chaotic. This means that there is extreme sensitivity of the solutions on initial conditions. In this regime,bad (and sometimes even good) guesses of the initial parameters get you stuck in local minima during the optimization process. So we simplify your life and we give you the value of some of the parameters.

In both problems, $\beta = 8/3$.

(a) Determine the value of ρ and of the initial condition (x_0, y_0, z_0) of the trajectory found in “Lorenz_traj1.txt”, where $\sigma = 10$
(b) Determine the values of ρ and σ for the trajectory found in ”Lorenz_traj2.txt”, where the initial condition is $(x_0, y_0, z_0) = (5, 5, 5)$

(c) The data that we have generated is for times between $0 \le t \le 5$. Repeat the reconstruction exercise for a time window $0 \le t \le 30$. Does it still work? Could you comment on why?

Hint 1: for the initial guess of (x_0, y_0, z_0), look at your data!

Hint 2: start with low values of ρ

(2) **Planetary orbits**

The planet Neptune was mathematically predicted before it was directly observed. With a prediction by Urbain Le Verrier, telescopic observations confirming the existence of a major planet were made on the night of September 23–24, 1846, at the Berlin Observatory, by astronomer Johann Gottfried Galle (assisted by Heinrich Louis d'Arrest), working from Le Verrier's calculations. It was a sensational moment of 19th-century science, and dramatic confirmation of Newtonian gravitational theory. In François Arago's apt phrase, Le Verrier had discovered a planet "with the point of his pen".

In this homework Problem, we are letting you to discover Neptune in a Sun-Uranus-Neptune system using the modern version of the pen, namely with JAX. We will start with a single-planet problem to practice.

(a) Consider a two-body Uranus-Sun system. Follow the colab, and load the data from "orbit-data-2A.txt" and use JAX to find out the mass of the Sun m_S.

(b) Now we consider the three body system including the Sun, Uranus and Neptune. You will be given
– A short trajectory of Uranus "orbit-data-3B.txt". – Mass m_U. Distance from the Uranus to the Sun at the starting point r_{US}.
and **you will predict**
– **The mass of Neptune** m_N
– **The distance from Neptune to the Sun at the starting point** r_{NS}
More configuratiosn of the system:
– The Sun is fixed at $(0, 0)$
– The starting positions of the Uranus and the Neptune are $(r_{US}, 0)$ and $(r_{NS}, 0)$.
– Uranus and Neptune are put quite close to each other in order for the influence between Uranus and Neptune to show up quickly.
– In this question, the velocities are set to $\sqrt{M/r_0}$, which is the velocity of a circular orbital when feeling only the gravity from the sun. You can find it in the following code but you do not need to do any modification.
Report your findings to Kaggle!

(c) **Extra Credit** There is something fundamentally unsatisfactory about the way we are solving the optimization problem, at least as described in class. There we computed the L_2 error between the predicted and the ground truth trajectories, summed over timesteps. In reality we know that for planetary orbits, there are conservation laws – which are exact for the elliptical orbits of a planet going around the sun. This suggests another strategy for finding an optimal solution – which is to match the conserved quantities of the trajectory you are given with those of the predictions and use this to find the optimal solution. Please reformulate your optimization problem with conserved quantities as central and use this to show that we can get better convergence to the right answer.

(d) **Extra Credit** Historically, astronomers did not have differentiable models. How did they do it? Could you incorporate these ideas into differentiable models?