Problem Set 5

Issued: Monday, November 15, 2021.

Recommended due date for Problem 5.1. (a) and (b): Friday, November 19, 2021, 11:59 PM.
Part I: Due on Wednesday, November 24, 2021, 11:59 PM.
Part II (Optional): Due on Monday, November 29, 2021, 11:59 PM.

Part I: Ocean Flows

The Philippine Archipelago is a fascinating multiscale ocean region. Its geometry is very complex, with multiple straits, islands, steep shelf-breaks, and coastal features, leading to partially interconnected seas and basins. In this part, we will be studying, understanding, and navigating through the ocean current flows.

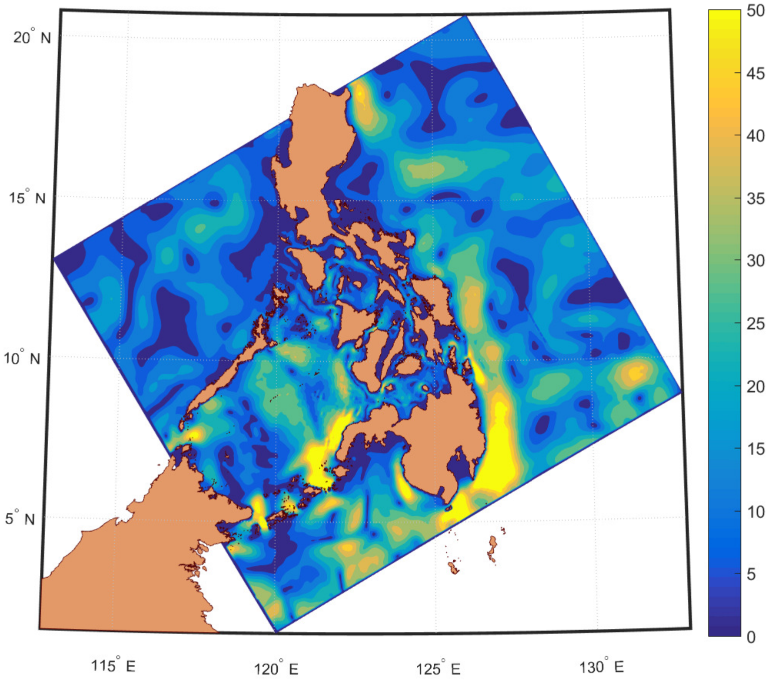

Figure 1: Snapshot of the ocean flow speed in the Philippine Archipelago.

The dataset may be found in `OceanFlow.zip`. It consists of the ocean flow vectors for time $T = 1, 2, ..., 100$. The flow in the dataset is an averaged measure over depth values, ranging from the surface to either (i) near the bottom or (ii) 400m of depth, whichever is shallower. The flow is provided as a 2-D vector field. The files `*u.csv` contain the horizontal components of the vectors, while the files `*v.csv` contain the vertical components. The numbers in the file names indicate the corresponding time T. For instance, files `24u.csv` and `24v.csv` contain pieces of information related to the flow at time $T = 24$. If needed, the file `mask.csv` contains a a 0-1 matrix identifying land and water.

Additional pieces of information and description of units: The data has been collected in January 2009. Of note, you need to multiply the flow values by 25/0.9 to get a unit of cm/second (cmps). The time interval between any two data snapshots is 3 hours. The grid spacing used is 3 kilometers. In this problem, the matrix index $(0, 0)$ will correspond to the coordinate (0km, 0km), or the ***bottom, left*** of the plot. For simplicity, we will not be using longitudes and latitudes in this problem.

Data sources: The data has been provided by the MSEAS research group at MIT (`http://mseas.mit.edu/`). The flow field is from a data-assimilative multiresolution simulation obtained using their MSEAS primitive-equation ocean modeling system. It simulates tidal flows to larger-scale dynamics in the region, assimilating a varied set of gappy observations.

Recommendations: We advise you to complete Problem 5.1 by Friday, November 19, 2021. This will give you more time to work on Problem 5.2 between Friday, November 19, 2021 and the due date for Part I, which is on Wednesday, November 24, 2021.

Problem 5.1: [30pts] Flows and correlations.

First, we propose to study spatial correlations in the ocean flow.

(a) Visualize the average flow (i.e., averaged over all times T, but not over locations) as a 2-D vector field. What are the strongest flow currents that run in the Philippine archipelago? Again, remember that the matrix index $(0, 0)$ corresponds in this problem to the coordinate (0km, 0km), or the ***bottom, left*** of the plot.

(b) Visualize the average speed of the flow (i.e., for each grid point, take the average of speed values over all times T) as a 2-D graph. Do you notice any areas with high average speed but low average flow, or vice versa? Why might that be?

(c) **Note: You may want to wait until after the lecture given by Professor Jegelka on Monday, November 22 to start exploring this question.** Now, consider the spatial correlation of this dataset. For this, consider take two points: $(140, 115)$ and $(400, 400)$. For each of these two points, plot the average correlation of ocean flow (choose either the speed measure or both the horizontal and vertical flows) with other points as a function of grid distance/L1-distance (consider only points within a reasonable distance, e.g., distance < 100 or 200). How would you describe the spatial correlations in ocean flow? Would Gaussian processes be a good model for spatial correlations in ocean flow? Explain why or why not.

Problem 5.2: [20pts] Predicting trajectories.

The goal of this problem is to simulate the trajectory of a particle moving in the ocean flow.

(a) We assume that a particle in the ocean, characterized by a given set of coordinates (x, y), will inherit the velocity corresponding to the flow at these coordinates. Implement a procedure to track the position and the movement of the particle, caused by the time-varying flow. Explain the procedure, and show that it works by providing examples and plots.

Suggested approach: The data provides a discretization of the ocean flow. The particles will however be moving on a continuous surface. For simplicity, let us assume that the surface is the plane $\mathbb{R}^2$. The data can be seen as providing flow information at integer points, namely at (m, n) for m and n integers. Divide the continuous surface into squares, such that each square contains a unique data point. For this, you can assign to every point on the surface the closest data point. For instance, given $(x, y) \in \mathbb{R}^2$, this would consist in rounding both x and y to the closest integer. You may then suppose that each square has the same flow information as the data point it contains.

Now take a particle at (x, y) in a certain square. The flow in the square will displace it at the registered velocity. Once the particle moves out of this square, it is then governed by the flow information of the new square it is in.

(b) A (toy) plane has crashed in the Sulu Sea at $T = 0$. The exact location is unknown, but data suggests that the location of the crash follows a Gaussian distribution with mean $(100, 350)$ (namely $(300km, 1050km)$) and with covariance matrix $\sigma^2 I$. The debris from the plane have been carried away by the ocean flow. You are about to lead a search expedition for the debris. Where would you expect the parts to be at 48, 72, and 120 hours after the crash? Study the problem by making the variance of the Gaussian distribution vary. Either pick a few variance samples or sweep through the variances if desired. For the variance, you may want to consider a set of values ranging from 0km to 100km for instance. **Hint:** Sample particles and track their evolution.

Recommended Python Packages for this problem:

Some Python packages that might be useful for implementing the coding parts of this assignment include the following:

```
matplotlib.pyplot.quiver
matplotlib.pyplot.imshow
numpy.random.multivariate_normal
```

For non-Python suggestions, reach out to your classmates on Piazza.

Part II: Theoretical Considerations (optional for all)

Problem 5.3: Gaussian processes. It is recommended to complete this problem by November 29, 2021 as a way to review course material and prepare for Quiz II, which will be on December 1, 2021.

(a) Different kernels or covariance functions can have different properties for Gaussian process regression. Consider the (noise-free) squared exponential/RBF covariance function:

$$\kappa(x_i, x_j) = \sigma^2 \exp\left(\frac{-(x_i - x_j)^2}{2\ell^2}\right)$$

What is the effect of changing the signal variance (σ^2) and the lengthscale (ℓ)?

(b) Prediction with Gaussian processes can become computationally expensive when we have a huge number of observations. Why? Explain in your own words.

(c) How could you make this computation more efficient? Suggest a solution and an algorithm. Include pseudo-code to explain your approach.

Hint 1: There are many possibilities, including (i) shrinking the data or (ii) using the Sherman-Morrison-Woodbury formula (see details below for a numerical linear algebra complement). If you follow any of these two suggestions or your own idea, say how exactly you would do it and why, and explain how you would ensure that you do not lose too much prediction quality. Indeed, your predictions will be approximations to predictions obtained using the full, expensive Gaussian process. You may use the form of prediction with noisy observations.

Hint 2: If you want to use the aforementioned Sherman-Morrison-Woodbury matrix inversion lemma (from numerical linear algebra), here is a form that is useful for this problem:

$$(Z + UWV^\top)^{-1} = Z^{-1} - Z^{-1}U(W^{-1} + V^\top Z^{-1}U)V^\top Z^{-1},$$

where $Z \in \mathbb{R}^{n\times n}$, and $U, V \in \mathbb{R}^{n\times m}$.