Problem Set 5
Statistics, Computation and Applications

Felipe del Canto

November, 2021

Problem 5.1: Flows and Correlations

For parts (a) and (b), the first step is to compute the average flow over time. The grid used implies that each coordinate covers a 3km grid. Additionally, all flow values are expressed in cm/second (cmps). In order to gain a better sense of the strength of the currents, two different average speeds are computed. First, the speed of the average flow. This is, compute the speed *after* averaging the flow vectors. Second, the average of all speed values over time. In other words, the speed is *before* averaging. In Figure 1a it is presented the first of these two speed representations. The fastest flow runs from east to west on the southern end of the Archipielago (around 900 km east of the initial coordinate). There is also a strong current around the islands to the southwest (around 300 km north of the initial coordinate) Other important flows run on the east, mostly going south. There are lesser important but prevalent currents running almost in circular patterns at the center and the west and at the center-north, on the east. In contrast, in Figure 1b is presented the second of the speed representations. In this case, the speed of the flow at each time is computed and then averaged over time. The patters shown in the previous picture still hold. The most strong currents are present in the southern part of the Archipielago, and in the islands at the south west (around 900 km to the north of the initial coordinates).

For part (c), the main goal is to understand if a Gaussian Process (GP) might be a good model for making predictions about currents in the Archipielago. For this, spatial correlations must be understood. To this end, two coordinates are considered and the flow and speed correlations with their surroundings are studied. The two coordinates are (420 km, 345 km) and (1200 km, 1200 km). Moreover, the correlations are computed for three variables: the horizontal flow, the vertical flow and the speed. In Figure 2 are presented the results for both coordinates. In particular, panels (a) to (c) show the correlations for the first coordinate. These figures show that horizontal flow is more diverse in terms of correlation, with zones close to the point of reference that have high and low correlation values. On the opposite side, the vertical component of the flow is in general more positively correlated with the reference point. In particular, the previous observation suggests that a GP would not be a reasonable model for the horizontal component of the flow. This, because the red zones are roughly at the same distance that other blue zones. If a GP with covariances following a kernel modelled these zones, then both should have the same color. Additionally, the zones south of the reference point do not seem correlated with it, mounting more evidence against a GP model. In contrast, the vertical component and speed images do have a more stable behavior in terms of the neighbors of the reference point. However, the same concern regarding zones more to the south suggests that GP is not a good model. Now, we turn to the surroundings of the second coordinate, which are shown in Figures 2d to 2f. These panels show a completely different story. Around the reference point, the correlations are spurious. In particular, the zones closer to the point have almost zero correlation. In that sense, it could be argued that in this case, GP are a good model. However, the small correlations would imply a very poor predictive power, making the model useless nonetheless. In the other hand, the caveat is that even if the surroundings of the reference point do follow a distance-driven covariance pattern, points further away have nonzero correlations. If a GP were to model this data, the correlation between the reference point and these far away points should be smaller than those closer, which is not the case. All

in all, the analysis of the previous two coordinates suggests that a GP would not be a good model for this data. The reason might be that currents form a complex system, with difficult rules. If these rules are nonlinear, then correlations, that are more suited for linear dependencies, fail to capture the underlying phenomena.

Problem 5.2: Predicting Trajectories

For part (a), the goal is to implement a procedure to track the position and movement of a particle moving through the ocean flow given by the data. The following approach is considered. Given initial coordinates (x_0, y_0), the movement of the particle is simulated in a second-to-second basis. This is, the new position of the particle is computed every second. The flow that governs the particle's movement is given by the flow of the grid the particle is in, and the current timestamp $T \in \{1, \ldots, 100\}$. Note that computing the movement in this fashion induces a minor estimation error. However, this is negligible given the size of the grid (3 km) and the timeframe considered (300 hours). In Figure 3 are presented four simulations with different starting points. The red dot presents the initial point, while the blue lines represent the trajectory.

For part (b),

Problem 5.3: Gaussian Processes

For part (a), consider the squared exponential/RBF covariance function

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

where σ^2 is the signal variance and ℓ is the lengthscale. If the signal variance is increased, then all things equal, points will have a higher covariance. In other words, the covariance curve (as a function of distance $|x_i - x_j|$) is higher, the higher σ^2 is. This is shown in Figure 4a. Moreover, the image shows the decay is similar among all three functions, reaching values close to zero when the distance is close to 3. On the other hand, varying the lengthscale has different effects on the covariance function. As shown in Figure 4b, changing the lengthscale drastically affects the decaying rate of the function. When ℓ is small, the decay is faster than when it is larger.

For parts (b) and (c), suppose we have data points $\{(x_i, y_i)\}_{i=1}^N$, where x_i are locations and y_i are given outcomes. Additionally, we have a new observation x_* for which we wish to estimate y_*. A Gaussian Process (GP) model assumes each y_i is drawn from Y_i and that $Y := (Y_1, \ldots, Y_N, Y_*)$ follows a multivariate normal distribution. Let μ and Σ be the mean and covariance matrix of that distribution and assume $\mu = 0$. Let σ_* be the variance of Y_*, σ_N be the $N \times N$ top left block of Σ, and let k^* be the vector whose entries are $k(x_*, x_i)$, where k is the kernel function that is used to model the covariances. Then, the parameters for the distribution of Y_* are:

$$\mu_{*\,|\,1:N} = k_*^T K_N^{-1} y_{1:N}$$
$$\sigma_{*\,|\,1:N} = \sigma_* - k_*^T K_N^{-1} k_*$$

Observe that computing both parameters implies that the matrix K_N has to be inverted. If N is large, this operation can take large amounts of time or even be impossible to do due to memory restrictions. In order to do this computation more efficient, an option is to compute the singular value decomposition (SVD) of K_N. However, the problem persists unless we are willing to approximate the SVD. For this, we can take the algorithm proposed by Halko et al.[1] In their framework, given K_N, the idea is to find an orthonormal $N \times \ell$ matrix Q with $\ell \leqslant N$ columns such that $QQ^T K_N \approx K_N$. Then, setting $B := Q^T K_N$, compute the SVD of B, $B = \overline{U}\Sigma V^T$ and produce a low-rank approximation of K_N by setting $U = Q^T \overline{U}$ and obtaining $K_N = U\Sigma V^T$. To find Q, the algorithm tries to obtain it by setting a tolerance ϵ and finding Q such that

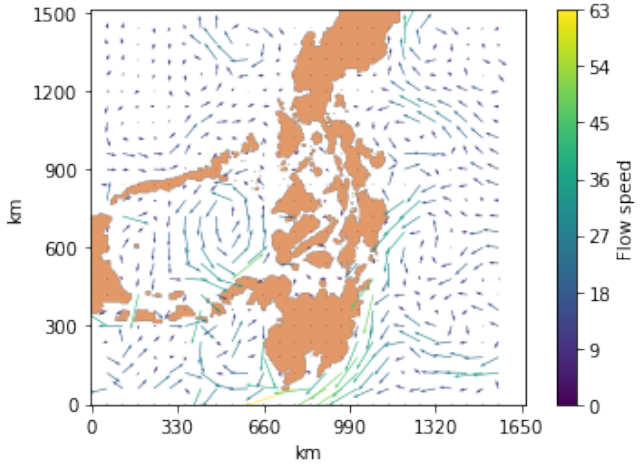
$$\|A - QQ^* A\| \leqslant \epsilon$$

This can be done by building Q incrementally in similar fashion as the Gram-Schmidt process. This last step involves drawing random gaussian vectors that help mapping the range of K_N. Since the SVD is computed in a smaller matrix than K_N the process is faster. The hyperparameter ℓ determines both the quality of the final approximation and how much the process is speeded.

[1] Halko, N., Martinsson, P.-G., & Tropp, J. A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. SIAM Review, 53(2), 217288.

Figure 1: Average flow and speed flow for a zone of the Philippines Archipielago. In panel (a), the rows represent the average flow over time. The colors on the arrows represent the speed of the flow (i.e., the size of the arrow) in cmps. In panel (b), the colors represent the average speed over time, that is, averaging the speed of the flow at each time.

(a) Average flow and speed



(b) Average speed over time

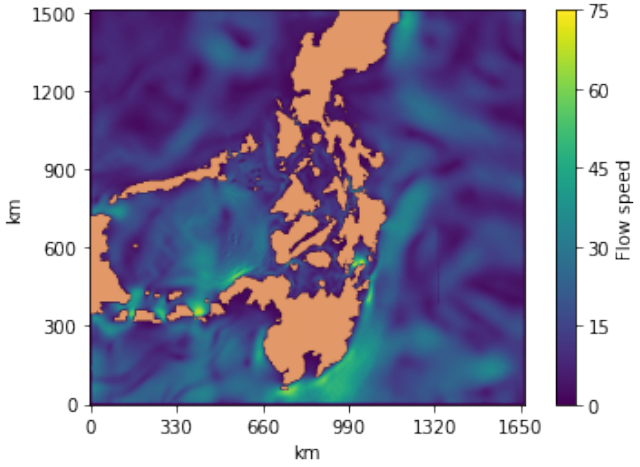
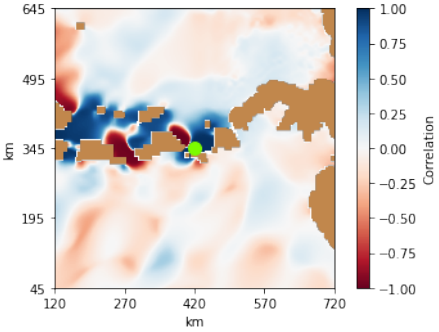
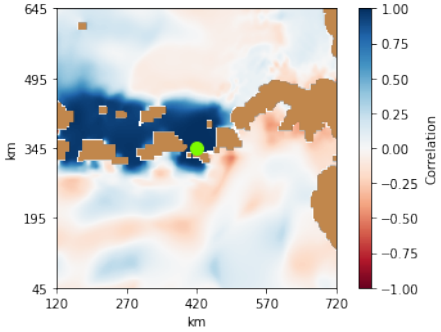


Figure 2: Spatial correlations for coordinates (420 km, 345 km) and (1200 km, 1200 km). Panels (a)-(c) show the correlations of the first coordinate and its surroundings for horizontal flow, vertical flow, and speed, respectively. Analogously, panels (d)-(f) show the same for the second coordinate. In all panels, the green dot represents the corresponding coordinate.

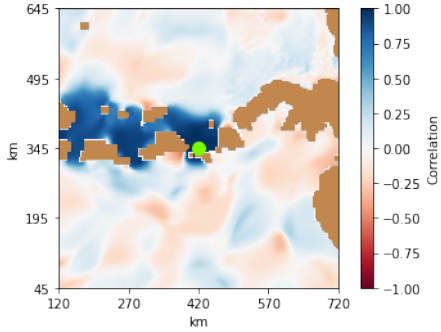
(a) Horizontal flow.



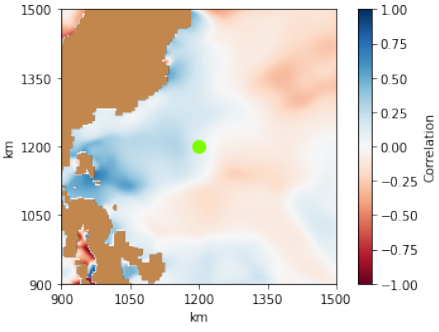
(b) Vertical flow.



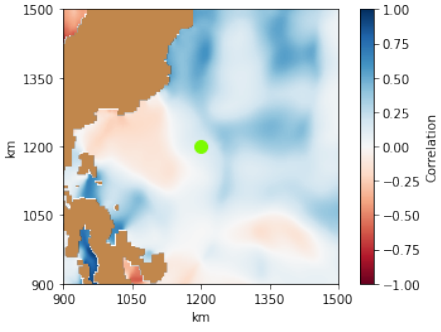
(c) Speed.



(d) Horizontal flow.



(e) Vertical flow.



(f) Speed.

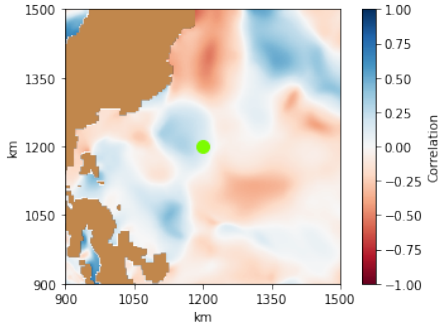
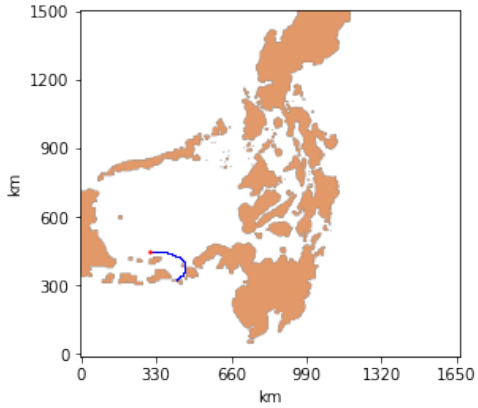
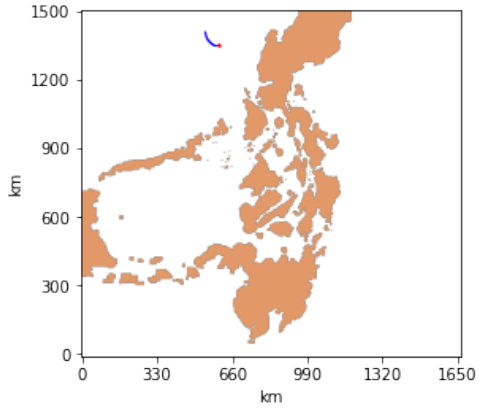


Figure 3: Simulation of trajectories of a particle subject to the ocean flow, from a given initial coordinate. The red dot represents the starting point and the blue line represents the journey of the particle.

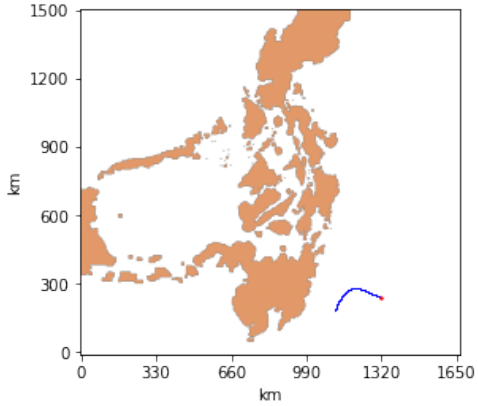
(a) (300, 462)



(b) (600, 1362)



(c) (1320, 252)



(d) (1320, 1212)

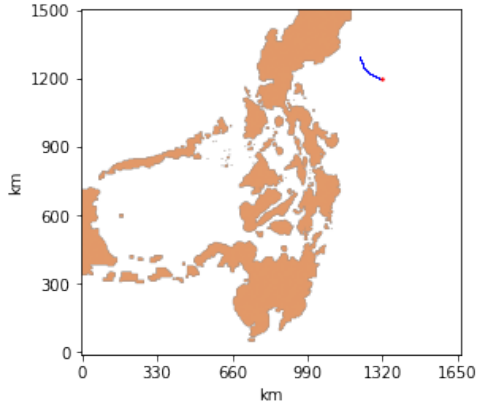
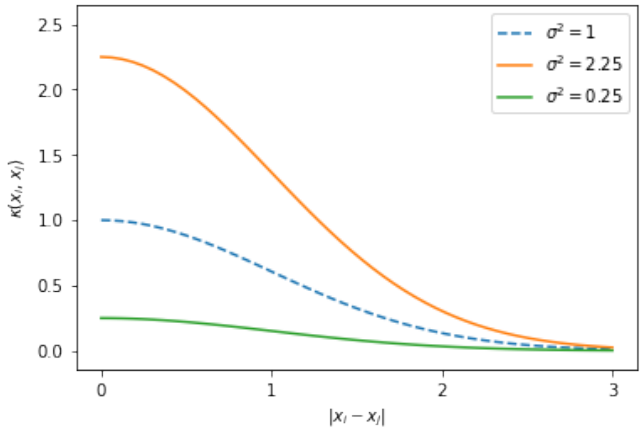


Figure 4: Effect of σ^2 and ℓ on the RBF covariance function $\kappa(x_i, x_j)$. Panel (a) shows the effect of varying σ^2, when holding $\ell = 1$ constant. In panel (b) it is shown the effect of varying ℓ, holding $\sigma^2 = 1$ constant.

(a) $\ell = 1$



(b) $\sigma^2 = 1$