Print ISSN: 0976-2876 Online ISSN: 2250-0138

Available online at: http://www.ijsr.in

INDIAN JOURNAL OF SCIENTIFIC RESEARCH

DOI:10.32606/IJSR.V14.I1.00005

Received: 28-05-2023 Accepted: 29-07-2023 Publication: 31-08-2023

Indian J.Sci.Res. 14 (1): 29-36, 2023 Original Research Article

USING NUMERICAL METHODS TO TRACK OCEAN PLASTICS IN THE NORTH INDIAN OCEAN

KARAN CHAKRAVARTHY[1]

Department of Environmental Sciences, American International School, Chennai, India

ABSTRACT

As oceans get progressively more polluted and garbage patches are discovered across the globe, tracking the movement of waste across the oceans becomes increasingly important. Modeling this movement makes identifying ocean gyres (and possible garbage patches) easier and could serve as an educational tool to demonstrate the effects of discarding trash in the ocean. To create such a model, we represented ocean currents as a vector field, and the fourth-order Runge Kutta method is used along with three different interpolation methods as bilinear, biquadratic, and bicubic interpolation. Bicubic interpolation was chosen as the reference solution, and the error associated with bilinear and biquadratic interpolation was compared. Quantitative results show that there is no clear superiority of the biquadratic model over the bilinear model. Qualitative results show that the model performs as expected based on studies of ocean currents in the North Indian Ocean. This paper demonstrates how to implement such a model, a method for evaluation, and recommendations for further study.

KEYWORDS: Ocean, Biquadratic Model, Bilinear

Our oceans have progressively become more and more polluted with plastics, and much of that plastic comes from rivers. (Ritchie H., 2021) Moreover, with the discovery of the Great Pacific garbage patch, the Indian Ocean garbage patch, and others across the oceans, ocean current analysis has shown that trash entering the oceans tends to be caught within the ocean gyres and remain at sea. Studies are also showing that much of the trash expelled by rivers wash up on beaches (Egger M., 2023).

While there exists a handful of studies to understand how currents affect the movement of plastics across the oceans, most focus on the development of theoretical models that track the motion of particles in a vector field. Part of the challenge comes from the fact that there is little ground truth available to validate any model that is developed. Nonetheless, as the research develops and further empirical evidence is gathered, the background work done to develop and evaluate these models will be valuable.

The purpose of this work is to create a model using numerical methods to track the movement of plastics across the oceans, with a focus on the North Indian Ocean. The fourth-order Runge Kutta model was used to determine the trajectory of a point within a vector field of ocean currents. Three interpolation methods – bilinear, biquadratic, and bicubic – were implemented and evaluated, and the results presented.

DATASET AND REPRESENTATION

We decided to model the ocean currents as a two-dimensional vector field and plastics moving through the ocean as a particle's trajectory through this field. The vector field is a two-dimensional array of vectors sampled at equally spaced points across the Earth's surface.

We used data from NASA's Ocean Surface Current Analyses Real-time (OSCAR) global surface current database. OSCAR data provides current velocities for the ocean's mixed layer (also known as the surface layer). This topmost layer of the ocean is of almost uniform density and lies above the pycnocline (Figure 1). OSCAR data is derived from satellite measurements of sea surface heights, ocean winds, and sea surface temperatures, and the dataset is updated approximately every five days. For the work described in this paper, the dataset used was from January 1, 2023.

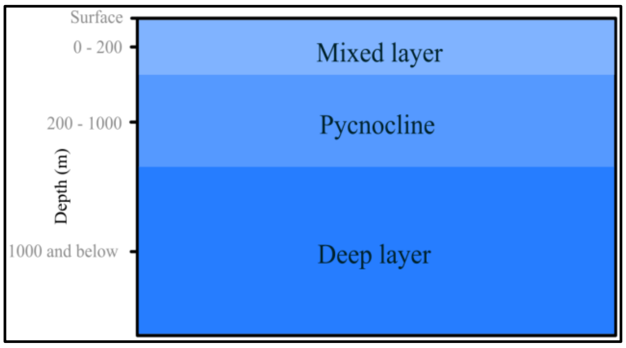


Figure 1: Diagram of the ocean layers

[1]**Corresponding author**

Four fields were extracted from the OSCAR dataset: u, v, latitude, longitude. The u and v fields represent the two components of the vector current: the zonal current component (east-west currents along the lines of latitude) and the meridional current component (north-south currents along the lines of longitude) respectively with units of meters per second. The latitude field is a set of Float64 values representing the latitudes from -80N to +80N divided into 480 values, with each value separated evenly by 0.333 degrees. The longitude field represents the longitudes from 20E to 420E divided into 1200 values, with each value similarly separated evenly by 0.333 degrees. Together, this data results in a grid of 480 x 1200 with each grid point associated with a latitude, longitude, and 〈u, v〉 vector. Each vector represents the surface currents across approximately 32 square kilometers.

Instead of using latitudinal and longitudinal values to calculate the trajectory of a particle, the latitude and longitude values were mapped to a 2D array of (x, y) index values with x values ranging from 1 to 480 and y values ranging from 1 to 1200. The latitude and longitude data were overlaid in such a way that $(x, y) = (1, 1)$ represented a latitude of 80N and a longitude of 20E while $(x, y) = (480, 1200)$ represented a latitude of -80N and a longitude of 420E.

METHODS AND IMPLEMENTATION

Runge Kutta Method

After consideration of various numerical methods to trace a particle's path through a vector field, we chose the fourth order Runge Kutta algorithm. The fourth-order Runge Kutta algorithm is a commonly used predictor-corrector method and uses a weighted average of four nearby vectors to determine the next step in a particle's path.

Given a starting point p_0, the first vector $\vec{k}_0$ is the vector at point p_0. The point p_1 is then calculated as one half of the step size along the vector $\vec{k}_0$ from the point p_0. The second vector $\vec{k}_1$ is the vector at p_1, and p_2 is then calculated as one half the step size along the vector $\vec{k}_1$ again from the point p_0. The third vector $\vec{k}_2$ is the vector at p_2, and p_3 is calculated as the step size along the vector $\vec{k}_2$ from point p_0. Finally, the fourth vector $\vec{k}_3$ is the vector at point p_3. The equations for each of these vectors are given below where $f(p)$ is the vector at the point p_0 and Δt is a constant step size.

$$\vec{k}_0 = f(p_0)$$

$$p_1 = p_0 + \frac{1}{2}\Delta t \cdot \vec{k}_0$$

$$\vec{k}_1 = f(p_1)$$

$$p_2 = p_0 + \frac{1}{2}\Delta t \cdot \vec{k}_1$$

$$\vec{k}_2 = f(p_2)$$

$$p_3 = p_0 + \Delta t \cdot \vec{k}_2$$

$$\vec{k}_3 = f(p_3)$$

The final position vector k is created as a weighted average of the four vectors k_0, k_1, k_2, k_3 according to the following equation.

$$\vec{k} = \frac{1}{6}\vec{k}_0 + \frac{1}{3}\vec{k}_1 + \frac{1}{3}\vec{k}_2 + \frac{1}{6}\vec{k}_3$$

With the final position vector calculated, the next point in the trajectory is given by

$$p_{n+1} = p_n + \vec{k}$$

This is depicted pictorially as shown in Figure 2.

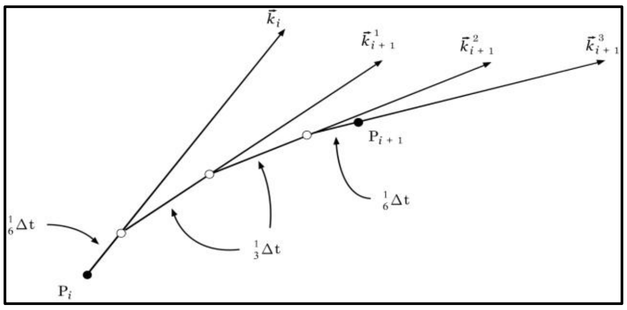


Figure 2: Diagram of weighted vectors in the RK4 algorithm

Bilinear Interpolation

Often, a vector field in $\mathbb{R}^2$ is described as the gradient of a function $f(x, y)$ so a vector can be determined at any point in the field. However, because ocean currents are represented by a sampling of points across a vector field, it is necessary to interpolate the vectors that lie between the discrete samples. To do so, we considered three methods to interpolate vectors between the discrete samples: bilinear interpolation, biquadratic interpolation, and bicubic interpolation. These methods are variations of linear, quadratic, and cubic interpolation applied to a 2D space.

Bilinear interpolation is the simplest of these three and is commonly used in computer graphics and image processing to estimate values between grid points or other applications where the dataset is relatively uniform. It involves determining the value of an unknown point based on the values of the 4 neighboring points.

Assume a point P to be interpolated with location (x, y) lies within a rectilinear cell of four points (x_1, y_1), (x_2, y_1), (x_1, y_2), and (x_2, y_2) each of which is associated with a vector as shown in Figures 3 and 4.

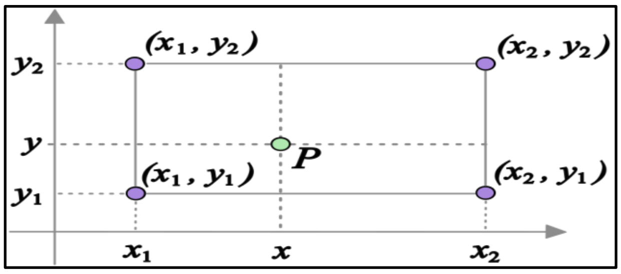


Figure 3: Visual representation to find percentage distance from closest corner (x_1, y_1) to P

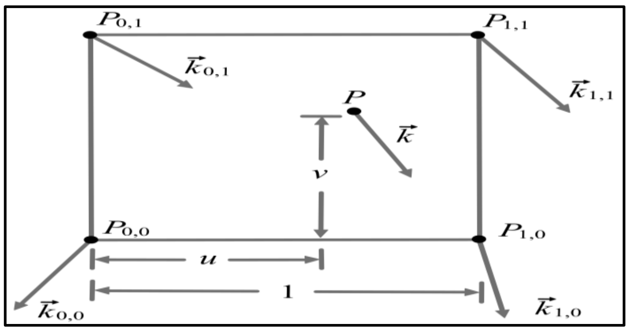


Figure 4: Rectilinear cell of four vectors with calculate vector at point P

The vector $\vec{k}$ at point P is calculated as follows:

$$\vec{k} = \frac{y_2 - y}{y_2 - y_1}\vec{k}_1 + \frac{y - y_1}{y_2 - y_1}\vec{k}_2$$

where

$$\vec{k}_1 = \frac{x_2 - x}{x_2 - x_1}\vec{k}_{11} + \frac{x - x_1}{x_2 - x_1}\vec{k}_{12}$$

$$\vec{k}_2 = \frac{x_2 - x}{x_2 - x_1}\vec{k}_{21} + \frac{x - x_1}{x_2 - x_1}\vec{k}_{22}$$

Biquadratic Interpolation

While bilinear interpolation can provide a suitable representation of the vector field, biquadratic interpolation provides a smoother curve between the discrete points. This technique involves fitting a smooth curve between data points using a piecewise-defined quadratic polynomial. Unlike bilinear interpolation, the biquadratic interpolation method takes nine data points surrounding a central point instead of four. This allows for a greater representation of the data as more data points are included when determining the interpolation.

With linear interpolation, one point is chosen on either side of the point to be interpolated. With quadratic interpolation, three points are considered. However, the odd number of points leaves two points on one side and one point on the other, and the interpolation would not be centered about the point to be interpolated.

To address this, we shifted the interpolation by half a unit. This was done by creating two virtual control points half way between the first and second points and the second and third points. The point to be interpolated is designed to be between these two virtual control points. The values for each of these virtual control points are determined by linearly interpolating between the two corresponding points. When applied to a 2D space, this results in 4 virtual control points as shown in Figure 5.

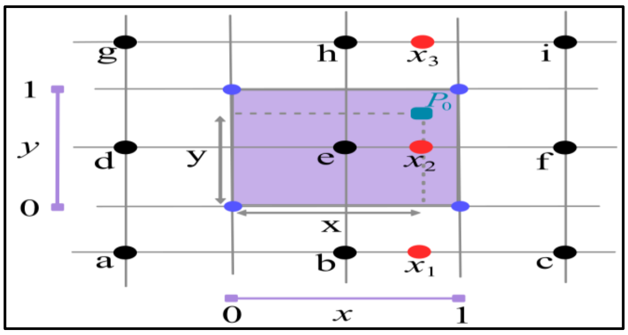


Figure 5: Biquadratic grid of points used to interpolate the data

Virtual control points shown in blue and interpolated points shown in red.

With the linear interpolation included as described above, the general equation used for quadratic interpolation across three points p_1, p_2, and p_3 is given as follows.

$$f_{quad}(p_1, p_2, p_3, x) = \frac{p_1 + p_2}{2} + 2\left(p_2 - \frac{p_1 + p_2}{2}\right)x + \left(\frac{p_2 + p_3}{2} - 2p_2 + \frac{p_1 + p_2}{2}\right)x^2$$

where x and y are normalized to be between 0 and 1 between the virtual control points.

When extended to biquadratic interpolation, the quadratic function is first interpolated three times along the x-axis as follows.

$$x_1 = f_{quad}(a, b, c, x)$$

$$x_2 = f_{quad}(d, e, f, x)$$

$$x_3 = f_{quad}(g, h, i, x)$$

Once the values for x_1, x_2, and x_3 are determined, the quadratic interpolation is performed once more along the y-axis to determine the value for P_0, the point to be interpolated.

$$p_0 = f_{quad}(x_1, x_2, x_3, y)$$

Bicubic Interpolation

Bicubic provides an even higher degree of smoothness between grid points, using a cubic function to fit a curve between the discrete points. Like bilinear interpolation, bicubic interpolation uses an even number of points so the creation of virtual control points is not needed.

Bicubic interpolation works by taking a group of 16 points in a 4 x 4 grid of data points surrounding the point to be interpolated. Like both of the other interpolation methods, the interpolation is first run across the x-axis to calculate the points x_1, x_2, x_3, and x_4, and then run once along the y-axis to determine the value at p_0. As earlier, x and y are normalized to be between 0 and 1. This is illustrated in Figure 6.

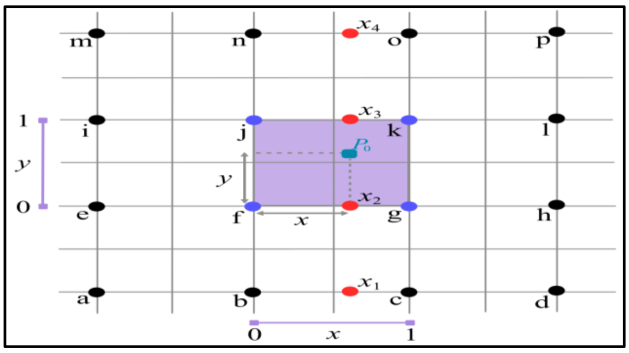


Figure 6: Grid of data points used for bicubic interpolation. Interpolated x coordinate points shown in red

The equation for cubic interpolation across the x-axis is defined below.

$$f_{cubic}(p_0,p_1,p_2,p_3,x) = \left(-\frac{1}{2}p_0 + \frac{3}{2}p_1 - \frac{3}{2}p_2 + \frac{1}{2}p_3\right)x^3 + \left(p_0 - \frac{5}{2}p_1 + 2p_2 - \frac{1}{2}p_3\right)x^2 + (-\frac{1}{2}p_0 + \frac{1}{2}p_2)x + p_1$$

This equation is applied four times across the points along the x-axis to determine the values for points at $x_1, x_2, x_3,$ and x_4.

$x_1 = f_{cubic}(a, b, c, , d, x)$

$x_2 = f_{cubic}(e, f, g, h, x)$

$x_3 = f_{cubic}(i, j, k, l, x)$

$x_4 = f_{cubic}(m, n, o, p, x)$

The value at p_0 is finally determined through one more application of the equation for cubic interpolation, but this time across the four interpolated points $x_1, x_2, x_3,$ and x_4 along the y-axis.

$p_0 = f_{cubic}(x_1, x_2, x_3, x_4, y)$

Implementation

With the ocean currents represented as a 2D vector field, a model was implemented in the Wolfram language using the fourth order Runge-Kutta algorithm and the three interpolation methods described above. Inputs to the model are the starting point, step size, and number of iterations. Each run of the model created three trajectories — one each for the linear, quadratic and cubic interpolation methods. For all runs, a step size of 0.5 was used with 800 iterations.

Because there is no known ground truth with which to evaluate the models, the bicubic interpolation was used as a reference by which to evaluate the linear and quadratic models as it represents the most mathematically accurate solution of the three. To calculate the error between the linear and quadratic trajectories with the cubic trajectory, the Euclidean distance E is first calculated between corresponding points along the two trajectories as follows:

$$Euclidean\ Distance\colon\ E(P,Q) = \sqrt{(Q_x - P_x)^2 + \left(Q_y - P_y\right)^2}$$

where P and Q are points in the 2D space.

The values for a given trajectory are then combined as the square root of the sum of squares. This is known as the lock-step Euclidean distance and is described as follows:

$$Lock\text{–}step\ Euclidean\ Distance\colon\ L(M,N) = \sqrt{\sum_{i=1}^{n} E^2(m_i, n_i)}$$

where M and N are two vectors of points in the 2D space and i ranges from 1 to the number of elements in the vector. In our case, M corresponds to the trajectory associated with the bicubic model and N corresponds to the trajectory associated with either the bilinear or biquadratic model. This approach allows us to quickly compare the bilinear and biquadratic trajectories.

This process was repeated for four 5 x 5 grids of GPS coordinates in the North Indian Ocean the Arabian Sea, the Bay of Bengal, northeast of Seychelles and southwest of Sri Lanka as shown in Figure 7, and the total errors across the trajectories were compared.

In addition, 16 GPS coordinates were evaluated qualitatively by observing the trajectories. Eight of these were chosen along the coast of India to be near river outlets into the sea, and eight were spread across the Bay of Bengal and the Arabian Sea. The processing times for each of these runs was also recorded.

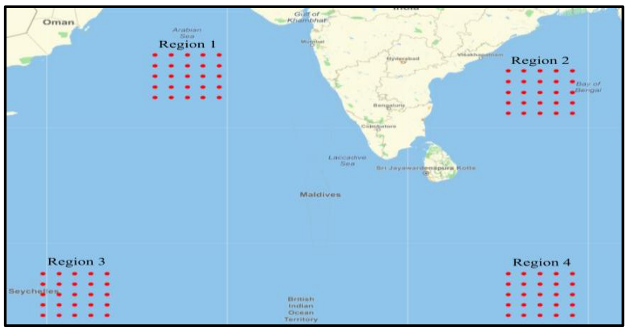


Figure 7: The four regions of 5 x 5 grids of coordinates on which the runs were tested

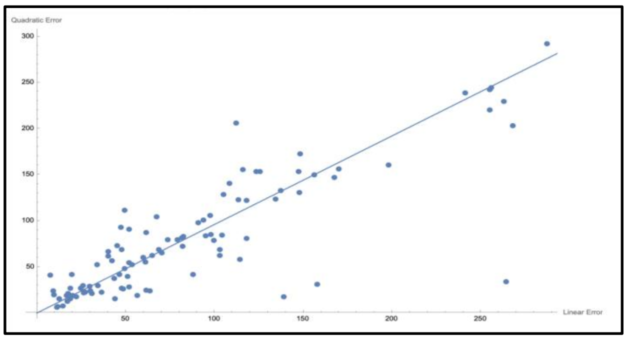


Figure 8: Bilinear and biquadratic error for each trajectory along with a best fit line

RESULTS AND DISCUSSION

Figure 8 illustrates a graph of the total error associated with both the bilinear and biquadratic models when compared to bicubic for all of the points tested. For a given trajectory, the bilinear error is represented on the x-axis, and the biquadratic error is represented on the y-axis. A determination of the line of best fit yields the line $y = 0.9609x$ and is also overlayed on the plot. Because the slope of the line is nearly one, points under the line generally indicate samples where the biquadratic model performed with less error than the bilinear model, and points above the line indicate the contrary.

As can be seen from the figure, the points appear to cluster around the line $y = 0.9609x$, suggesting the biquadratic model performs similar in accuracy to the bilinear model, if not slightly better. Table 1 also shows the count of trajectories for each region where the bilinear error is greater than the biquadratic error and vice versa. The biquadratic model performs better in 57% of the trajectories.

Table 2 shows the time taken to calculate several trajectories with each of the interpolation methods. The results are consistent across the runs, regardless of the starting points of the trajectories. Bilinear and biquadratic interpolation are similar in the absolute times taken with bilinear interpolation being slightly faster than biquadratic. However, bicubic interpolation results in approximately a 10x increase in computation time over both bilinear and biquadratic.

Table 1: Comparison of bilinear and biquadratic error across trajectories by region

Region	Bilinear error > Biquadratic error	Biquadratic error > Bilinear error
I	9	16
II	13	12
III	12	13
IV	9	16

Table 2: Time taken in seconds to generate the trajectories for the bilinear, biquadratic, and bicubic methods

Latitude	Longitude	Bilinear	Biquadratic	Bicubic
19	72.4	0.578843	0.612795	5.88402
13	80.5	0.566436	0.593552	5.88056
15.4	73.7	0.573438	0.607441	5.83616
12.9	74.6	0.570757	0.613081	5.87552
9.95	76.2	0.574994	0.600569	5.8592
8.64	78.15	0.58338	0.602804	5.93379
11.36	79.85	0.566767	0.60121	5.84195
15.66	80.98	0.572612	0.599231	5.88178
20.08	86.77	0.563314	0.609077	5.92286
21.4	91.02	0.562457	0.593427	5.86548
13.9	84.1	0.580862	0.609787	5.90423
13.9	88.1	0.580396	0.620646	5.99402
13.9	92.1	0.582617	0.619004	5.95934
19	70	0.583237	0.629257	5.87745
14	70	0.581369	0.612338	5.89972
9	70	0.581801	0.615364	5.94343

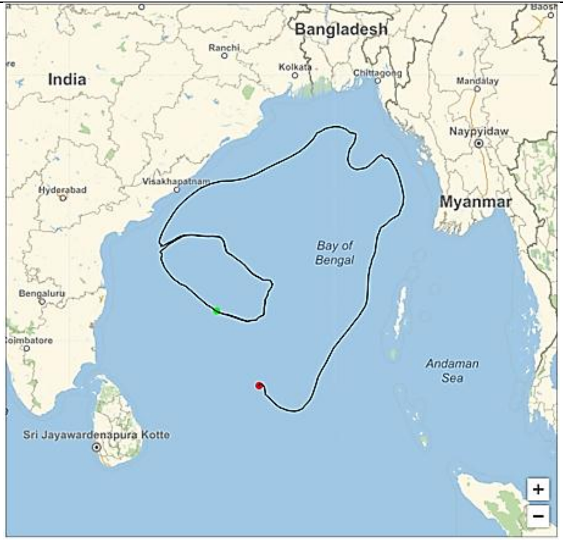
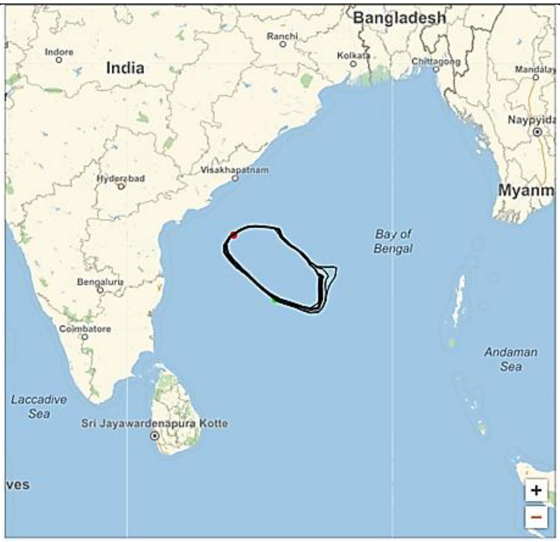
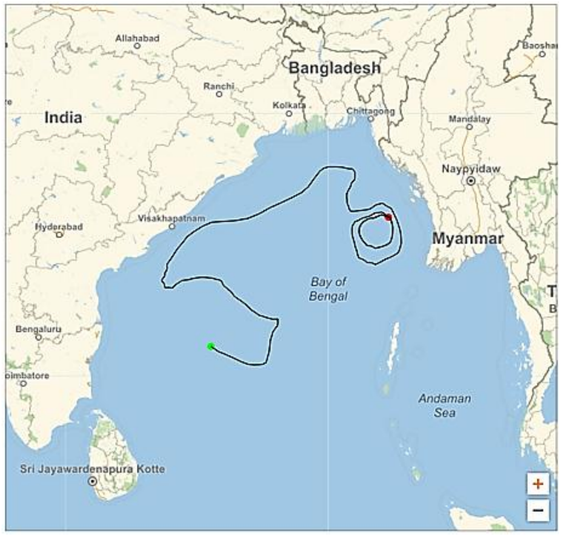
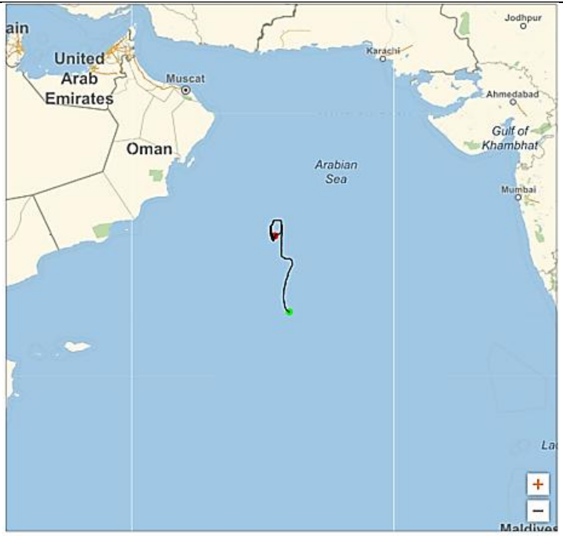
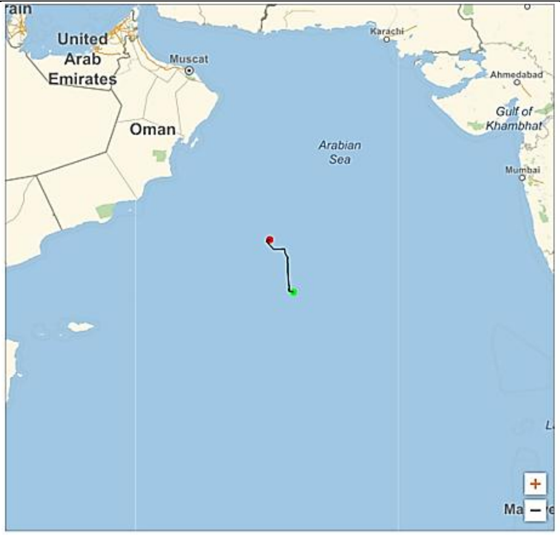
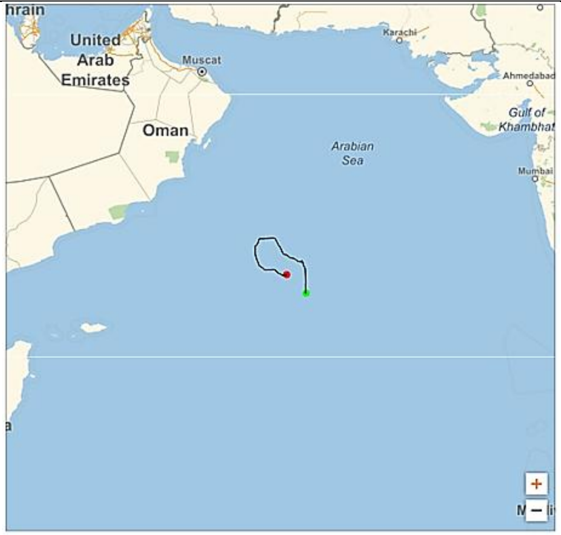
Bilinear Model	**Biquadratic Model**	**Bicubic Model**
	Starting Point: (12.5, 85)	
		
	Starting Point: (14, 63)	
		

Figure 9: Trajectories associated with the 3 interpolation methods with a GPS starting point (13.9, 88.1)

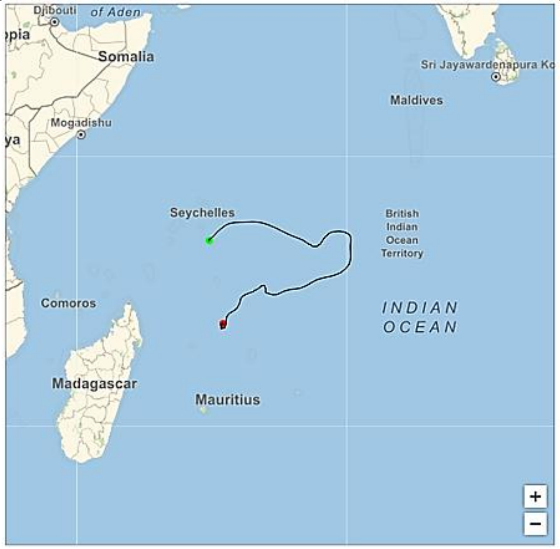
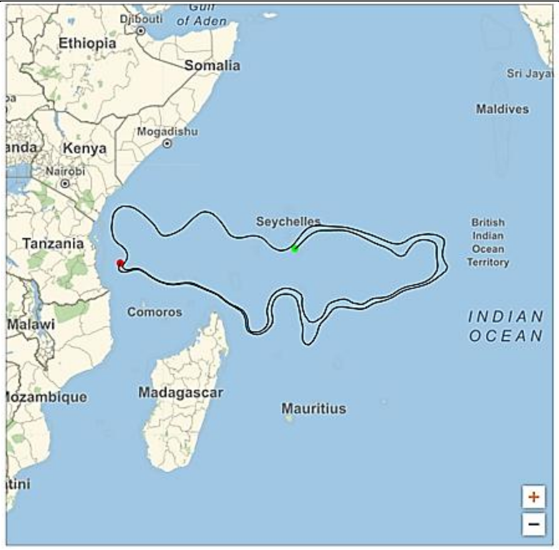
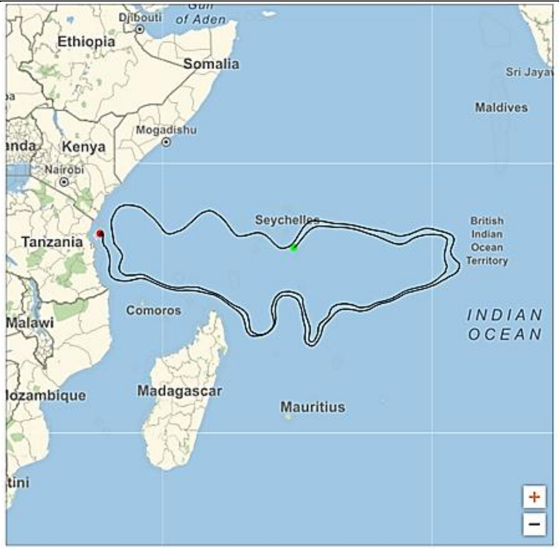
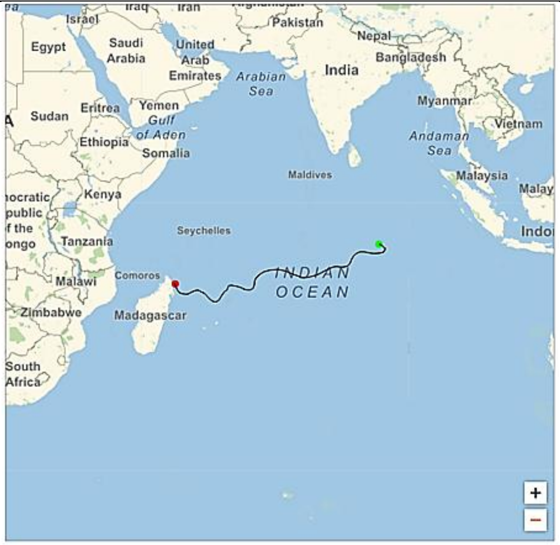
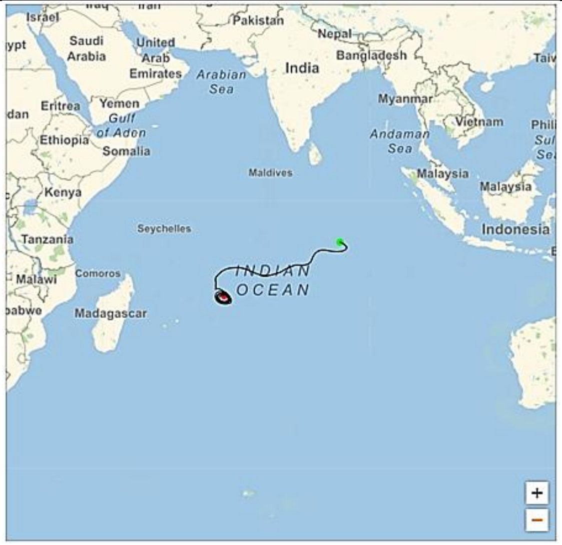
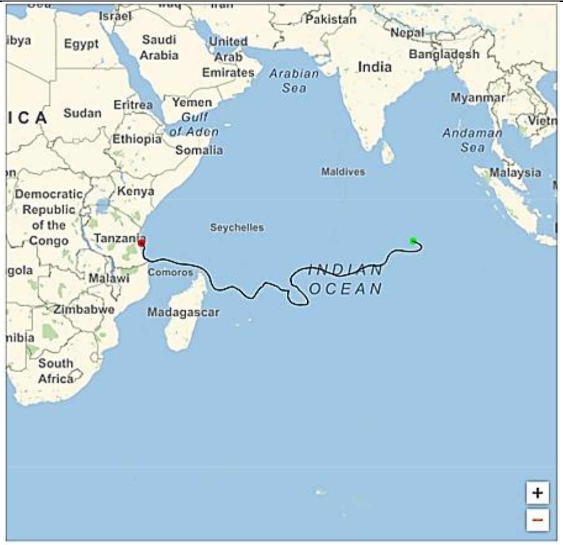
Bilinear Model	Biquadratic Model	Bicubic Model
Starting Point: (-7, 56)		
		
Starting Point: (-7, 85)		
		

Figure 9: Trajectories associated with the 3 interpolation methods with a GPS starting point (13.9, 88.1)

These results indicate there is no significant benefit in a model with biquadratic interpolation over bilinear interpolation for this application under the current conditions.

In addition to the quantitative evaluation above, the trajectories were evaluated visually. All trajectories are not shown here for conciseness, but Figure 9 shows the trajectories associated with the first GPS coordinate for each region. In most cases, the trajectories starting near the shore result in endpoints along a nearby coastline. This points to the hypothesis that most plastic entering the ocean from points along the Indian coastline result in beaching along the Indian coastline or in fewer cases along other countries in the North Indian Ocean (eg. Sri Lanka, Maldives) (van der Mheen *et al.*, 2020).

Trajectories originating further from land were taken to be at least 90 km from the nearest point on the Indian coastline. This ensured that no points in the first iteration of the cubic interpolation calculation would be 0. While this set produced more variation in the trajectories than the set of trajectories starting near land, all trajectories starting in the Bay of Bengal ended in the Bay of Bengal, while all trajectories starting in the Arabian Sea ended in the Arabian Sea. This again agrees with the hypothesis that most debris originating in the Northern Indian Ocean remains in the Northern Indian Ocean.

CONCLUSION AND AVENUES FOR FURTHER STUDY

The aim of this paper is to explore the use of numerical methods to understand the movement of plastics in the ocean. With the ocean modeled as a vector field, the fourth order Runge-Kutta algorithm was used to track the movement of a particle across that vector field, with bilinear, biquadratic, and bicubic interpolation used to interpolate points between discrete samples. The computation times for each model were recorded, and with the bicubic model designated as the reference solution, the error was calculated for both the bilinear and biquadratic models. In addition, trajectories created by each of the models were assessed qualitatively.

Across a set of 100 trajectories taken from starting points in the Northern Indian Ocean, the error associated with the bilinear model appears to be similar to that of the biquadratic model, indicating there is no clear superiority of the biquadratic model as might have been expected for this scenario. The time taken to execute the biquadratic model is marginally longer than that of the bilinear model, so there also there is no clear advantage of one model over the other with regards to computation time.

There are numerous avenues for further study including variation of the step size and more granular ocean current data, as both of these could show more differentiation in performance between bilinear and biquadratic models.

REFERENCES

Egger M., 2023. “The Other Source: Where Does Plastic in the Great Pacific Garbage Patch Come From?” The Ocean Cleanup, 17 July 2023, www.theoceancleanup.com/updates/the-other-source-where-does-plastic-in-the-great-pacific-garbage-patch-come-from/.

Ritchie H., 2021. “Where Does the Plastic in Our Oceans Come From?” Our World in Data, 1 May 2021, www.ourworldindata.org/ocean-plastics.

van der Mheen M., van Sebille E. and Pattiaratchi C., 2020. Beaching patterns of plastic debris along the Indian Ocean rim. Ocean Sci., **16**: 1317–1336, https://doi.org/10.5194/os-16-1317-2020.