First conclusions of a probabilistic approach to finding airplane debris at sea

Monday, February 9, 2015

Contents

1 Introduction

The disappearance of the Malaysian Airlines flight 370 (MH370) in March 2014 is tragic not only for the great loss of life but also for the failure of any trace of the aircraft debris to turn up in searches for it across the ocean. The mystery surrounding MH370's disappearance is magnified by the unending list of unknown variables search teams have attempted to overcome: the flight trajectory of the aircraft, the location of the aircraft's impact with the ocean, and the day-to-day changes in the ocean's currents are three such examples making precise searching nigh impossible. Finding MH370, and other tragedies like it, is thus equivalent to finding a needle in a sea-sized haystack.

This paper seeks to begin to answer some of the questions posed by the search problem. Once the debris of the aircraft crashes into the ocean, it is subject to a host of forces that are, due to the size of the system the ocean comprises, characteristically non-deterministic. We will thus set forth in establishing a model by which we can predict the trajectory of debris in the ocean from a probabalistic approach incorporating known data regarding ocean currents. We also will outline a method for determining the path a search aircraft ought to take in order to find the debris after considering extensive simulation of the trajectory of the debris.

We proceed as follows: Section 2 will describe the model, its assumptions, and core computational procedures. In Section 3, we will apply the model to publically-available datasets of the ocean's surface velocities and interpret our findings. Finally, in Section 4, we will discuss the evidence provided and seek to contextualize the results of our simulations within the broader problem, reflect on the modeling choices made throughout the process, and suggest areas in which our approach can be improved with future work.

2 Model

2.1 Modeling debris trajectory

The model we present is designed to provide insight into the trajectory of a missing aircraft by simulating its motion in the ocean. Modeling ocean dynamics from a first principles approach is both conceptually difficult and computationally expensive, compounded by the existence of freely-available large datasets detailing the surface velocity of points in the ocean over extensive periods of time. Thus, our model is built around an existing dataset which gives zonal and meridional velocities for latitudinal and longitudinal coordinates evenly spaced around the Earth. Our high-level approach is to treat the Earth's oceans as a vector-field and aircraft debris as a particle subject to its flow.

We begin with an overview of the notation that we will use throughout the paper. Latitudinal and longitudinal coordinates are denoted ϕ and λ, respectively. We will colloquially call any pair (ϕ, λ) an *Earth-coordinate* pair. The zonal (that is, West-East) velocity at a given Earth-coordinate pair is denoted u, while meridional (North-South) velocity at a given Earth-coordinate pair is denoted v. If $u > 0$, the zonal direction is oriented towards East, and if $v > 0$, the meridional direction is oriented towards North. We denote the mass of a piece of debris by m. Both zonal and meridional velocities are given in units of kilometers per day.

The model we present relies of an interpretation of the ocean's currents as a vector field. In this respect, we make two key assumptions: debris, though endowed with mass, are treated as points on the vector field; and the velocity of the debris is wholly determined by the surface velocities of the ocean. In addition, numerous computations performed by the model assume that the immediate surroundings of each Earth-coordinate pair is locally Euclidean, so that changes in coordinates computed from kinematics on a Cartesian grid map one-to-one with points on the Earth's surface.

2.1.1 Local approximation of changes in earth coordinates

One fundamental aspect of our model is to update a particle's coordinates properly when it is subject to a velocity field. To compute changes in the debris' coordinates, we incorporate a means for converting (locally-Euclidean) distances given in kilometers to latitudes and longitudes. Let the radius of the Earth be given by R_E. The bearing from North, denoted θ, is approximated by considering the Euclidean triangle formed by the zonal and meridional velocities at any Earth-coordinate, so it follows that

$$\theta = \tan^{-1}\left(\frac{u}{v}\right). \tag{1}$$

Given the Earth-coordinate (ϕ_1, λ_1) along with a velocity (u, v), we can approximate the distance d traveled over the period of one day by

$$d = \sqrt{u^2 + v^2}. \tag{2}$$

Then, to compute the new Earth-coordinates (ϕ_2, λ_2), the literature [5] provides the following formulae:

$$\phi_2 = \sin^{-1}\left(\sin\phi_1 \cos\left(\frac{d}{R_E}\right) + \cos\phi_1 \sin\left(\frac{d}{R_E}\right)\cos\theta\right) \tag{3}$$

$$\lambda_2 = \lambda_1 + \tan_2^{-1}\left(\sin\theta \sin\left(\frac{d}{R_E}\right)\cos\phi_1, \cos\left(\frac{d}{R_E}\right) - \sin\phi_1 \sin\phi_2\right). \tag{4}$$

Here, $\tan_2^{-1}$ is a sign-sensitive generalized version of $\tan^{-1}$, which returns the angle in the proper quadrant from which the tangent was computed. We also note that Earth-coordinates are given in degrees, so proper attention must be given to ensure that they are converted properly.

2.1.2 Algorithms

The model is entirely reliant on a starting Earth-coordinate pair, the aircraft debris' mass, and a probability-control parameter $q \in [0, 1]$. The algorithm in Alg. 1 serves as the general execution framework for our model. Given an Earth-coordinate pair (ϕ, λ), the model infers the zonal and meridional velocities at that location by computing a bilinear interpolation of the velocities found in the box formed by the Earth-coordinates nearest to (ϕ, λ) in every direction based on the distances between (ϕ, λ) and its neighbors. Additionally, due to the fact that the dataset contains velocity data over time (in an update interval of five days), we use Alg. 1 to incorporate the most recent time data.

To obtain a velocity pair (u, v) at a point not featured in the dataset, we estimate the velocity at a given Earth-coordinate by considering the the four Earth-coordinates which form a box that surrounds it. The Earth-coordinates forming the box are calculated in Alg. 2 by determining the nearest latitudinal and longitudinal coordinates North, South, East, and West of the Earth-coordinates. We note that the Round routine used in Alg. 2 is the usual function one would expect.

After the box is calculated, the values are bilinearly interpolated, which we provide in Alg. 3. Bilinear interpolation generalizes the concept of linear interpolation on the line between two points by extending the intuition for preferencing closest-neighbors to four points in a plane [3]. Now, because the dataset, which this model incorporates, collects velocity data physically separated by latitude and longitude coordinates one degree apart, the algorithm for bilinear interpolation presented in Alg. 3 is a simplified version of the typical computation.

Once the model computes the zonal and meridional velocities (u, v) at the Earth-coordinate (ϕ, λ), the model then calculates how far the debris moves in that direction, which is described in Alg. 4. This component is dependent on the mass of the debris and also the probability-control parameter. When the model computes the new direction of motion, it finds the bearing of the debris (that is, the angle between North and the direction of motion), $\bar{\theta}$, by

$$\bar{\theta} = \tan^{-1}\left(\frac{u}{v}\right) \tag{5}$$

Importantly, there is a crucial assumption that the bearing can be estimated by treating the immediate surroundings of (ϕ, λ) as locally-Euclidean, which makes the computation tractable.

The probabalistic component of the model becomes relevant in determining the final angle to which the debris' coordinates update. In Alg. 5 we present a ProbAngle procedure which allows the debris to deviate from the thusfar deterministic trajectory. The probability-control parameter $q \in [0, 1]$, which is selected at the outset of the simulation, represents the likelihood that the debris continues in the previously computed path. ProbAngle also keeps track of the previous bearing, θ, so we can model the effect of massive debris' inertia to changing angles. Specifically, we update the new bearing $\bar{\theta}$ to be

$$\bar{\theta} = \theta + \frac{1}{\sqrt{m}}(\bar{\theta} - \theta) \tag{6}$$

This update step has several desirable features: first, if $m = 1$, the updated bearing is unchanged by inertia; second, as the debris becomes more massive it becomes increasingly unlikely for large, consistent variations in the bearing to occur; third, incorporating $\sqrt{\cdot}$ gives diminished inertia to scale, allowing some likelihood that even massive bits of debris are moved by the ocean current.

The model generates a random number $p \in [0, 1]$ and then configures the final bearing $\bar{\theta}$ according to the function

$$\bar{\theta}(p) = \begin{cases} \bar{\theta} & p < q \\ \bar{\theta} + \frac{2}{3}\theta & q \leq p < q + \frac{1-q}{2} \\ \bar{\theta} - \frac{2}{3}\theta & q + \frac{1-q}{2} \leq p \leq 1 \end{cases} \tag{7}$$

We note that the choice of $\frac{2}{3}$ as the coefficient for θ is arbitrary, which we discuss in Section 4. Proceeding with a final value for $\bar{\theta}$, the model calculates the distance traveled in one day by making use of Eq. (2), and computing the new Earth-coordinates by using Eq. (3) and Eq. (4). The updated coordinates are stored, and the model simulation repeats this process until it reaches a terminal number of iterations.

2.2 Formulation of a search solution

The model for simulating the trajectory of aircraft debris provides us with the data to find an optimal path for the search plane to follow in order to maximize the likelihood that it discovers the debris. We assume that the search plane is confined to a linear path, that it has an arbitrary quantity of fuel, that it can begin and end its search at any Earth-coordinate, and that it can properly identify any debris around its Earth-coordinates. With these assumptions in mind, we present a least-squares optimization for determining a search path.

Given that the downed aircraft crashes in the Earth-coordinates (ϕ, λ), we use the debris trajectory model to obtain N simulations for T periods (days). Each simulation n comprises two time series $x_n = (\phi_1, \phi_2, \cdots, \phi_T)$ and $y_n = (\lambda_1, \lambda_2, \cdots, \lambda_T)$. Let x_μ and y_μ be the unweighted average of each time series x_n and y_n, i.e. let

$$x_\mu = \frac{1}{N}\sum_{n=1}^{N} x_n \tag{8}$$

$$y_\mu = \frac{1}{N}\sum_{n=1}^{N} y_n, \tag{9}$$

with addition defined component-wise. Let $t = (1, 2, \cdots, T)$. We can estimate an optimal path by constructing an Ordinary Least Squares regression on both x_μ and y_μ, that is, by estimating $\bar{x}$ and $\bar{y}$ that minimizes the differences between the x_μ and y_μ and the line [1]. Thus, we have that

$$\bar{x} = \alpha_x t + \beta_x$$
$$\bar{y} = \alpha_y t + \beta_y$$

which will serve as our optimal line through which a search plane could try to identify debris.

3 Empirical Results

To model the surface-level velocity of the ocean's currents at a given time, we used the National Oceanic and Atmospheric Administration's Ocean Surface Current Analysis - Real time (OSCAR) ocean dataset (Courtesy of the OSCAR Project Office) [2]. This dataset contains zonal and meridional velocities which vary with respect to latitudinal and longitudinal position as well as time. The velocities are given in meters per second, so must be converted to kilometers per day by applying a scaling factor of 86.4. The OSCAR data is visualized in Fig. 1.

We give an example simulation in Fig. 2. Here we have a starting location of $(\phi_0, \lambda_0) = (34.5, 124.5)$. The probability-control parameter q is set to 0.4. The Ordinary Least Squares solution is the set of equations

$$\bar{x} = -0.0001t + 124.5009 \tag{10}$$

$$\bar{y} = 0.0494t + 33.4213 \tag{11}$$

which are overlayed along with (x_μ, y_μ) on the data.

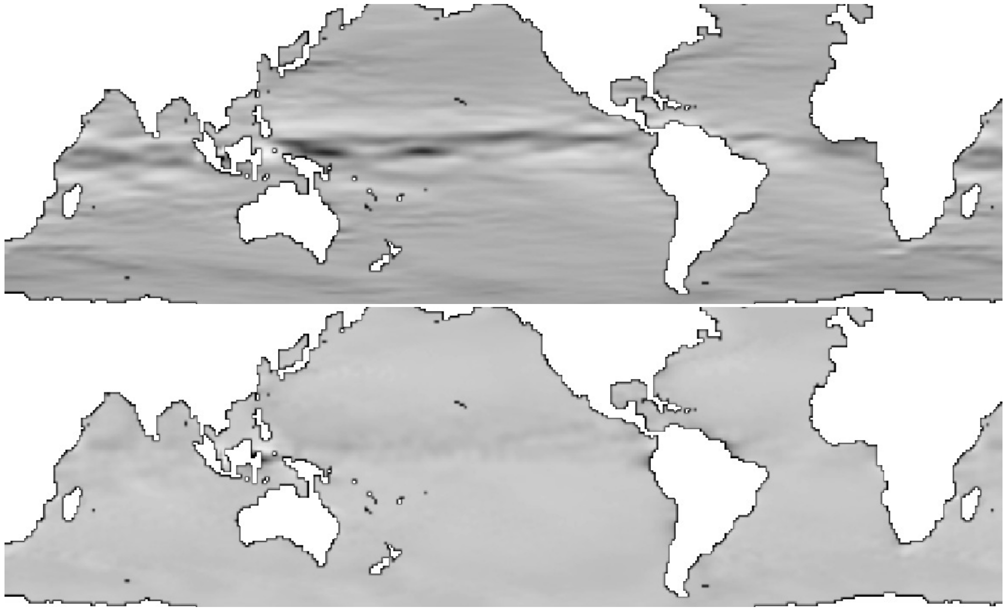

Figure 1: OSCAR dataset comprising zonal (top) and meridional (bottom) velocities. Darker colors on the zonal plot represent Eastward currents, and darker colors on the meridional plot represent Northward currents.

4 Discussion

Our model tends to behave according to reasonable assumptions. That the trajectories of more massive debris are more clustered than lighter debris makes intuitive sense, so it is a good sign that this phenonmenon is displayed in the data. Additionally, decreasing the probability-control parameter, which should theoretically increase the variability of the simulated data, does in practice behave in fashion. The least-squares approach to finding an optimal path for the search plane, from viewing the data anectodally, seems to provide the important features that we desired: namely, that while it is constrained to a linear path it tries to find with modest success a route that maximizes the likelihood that it finds any debris. Potential bugs in the implementation of our proposed simulation notwithstanding, these point to some merit in the model we have presented.

However, it is critical that we understand which questions our model does not answer. From a high-level perspective, our model does not take into account the depth-direction of the ocean currents. This is a limiting factor because dense debris will probably sink, not float, from the surface. Second, because we assume that each piece of debris can be modeled as a point-mass, we most certainly understate the effect of rotation on the trajectories we have simulated. We also need to investigate the error caused by assuming that the distances on which our model operates can be linearized, as such error accumulates throughout a simulation, which can drastically detriment the accuracy we need. Finally, we completely omit the primary problem surrounding MH370: we assumed throughout that the location of the plane when it crashed was known. In reality this is not the case.

Several modeling choices made at a lower level also need consideration. Bilinear interpo-

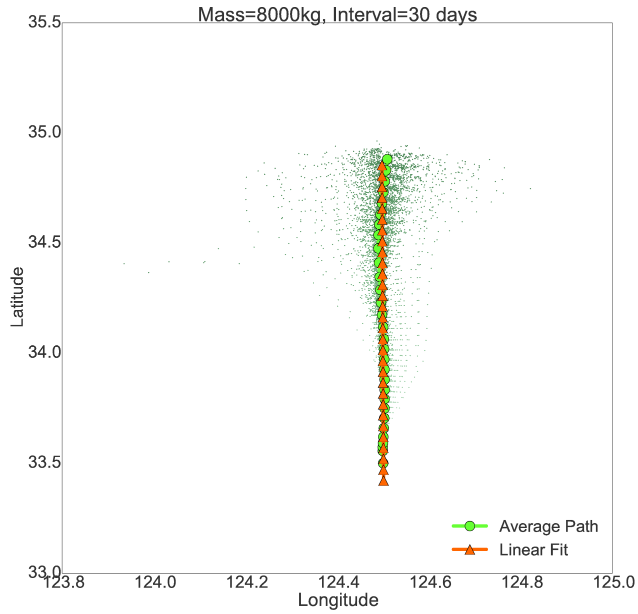

Figure 2: Simulated data starting from the Earth-coordinate location $(\phi_0, \lambda_0) = (34.5, 124.5)$, which is located in the East China Sea. The probability-control parameter q is set to 0.4. The trajectory tends towards North. The plot of the average path (x_μ, y_μ) and the parametrized regression line $(\bar{x}, \bar{y})$ are overlayed.

lation, while convenient to implement, may not be an accurate weighting scheme, especially at the distances we are interested in. It could well be that some sort of inverse-square law holds for approximating ocean current velocities. If that is the case then our model has significantly skewed the importance of far-away velocities. Our probability model is essentially contrived, with no evidence to support our method of altering the bearing. Moreover, while the way our model incorporates the mass of the debris in Eq. (6) behaves according to basic intution, a more sophisticated model would attempt to estimate this better through statistical inference. That $\frac{2}{3}$ is the offset designed in Eq. (7) is not supported by any evidence. The probability-control parameter q is similarly arbitrary. Linear regression, which provides a solution to our search problem, does not guarantee that it is optimal, because the objectives of linear regression do not necessarily match the ones posed by the search problem. An alternative would be to explore other forms of optimization, perhaps with linear programming.

One way of validating components of our modeling choices would be to analyze how robust the system to small changes in the underlying parameters. Here our model provides good metrics for testing. One could reasonably expect the linear regression to remain stable over small changes in the starting Earth-coordinates, and this hypothesis could be tested simply by choosing origins in a neighborhood around some fixed location. Similarly, the linear regression line could be used to test changes in the arbitrarily-chosen parameters previously mentioned. Finally, to validate this approach, more testing is required over much larger swathes of ocean to accumulate a representative sample of origins.

Following sensitivity analysis, an important goal for future work would be to attempt

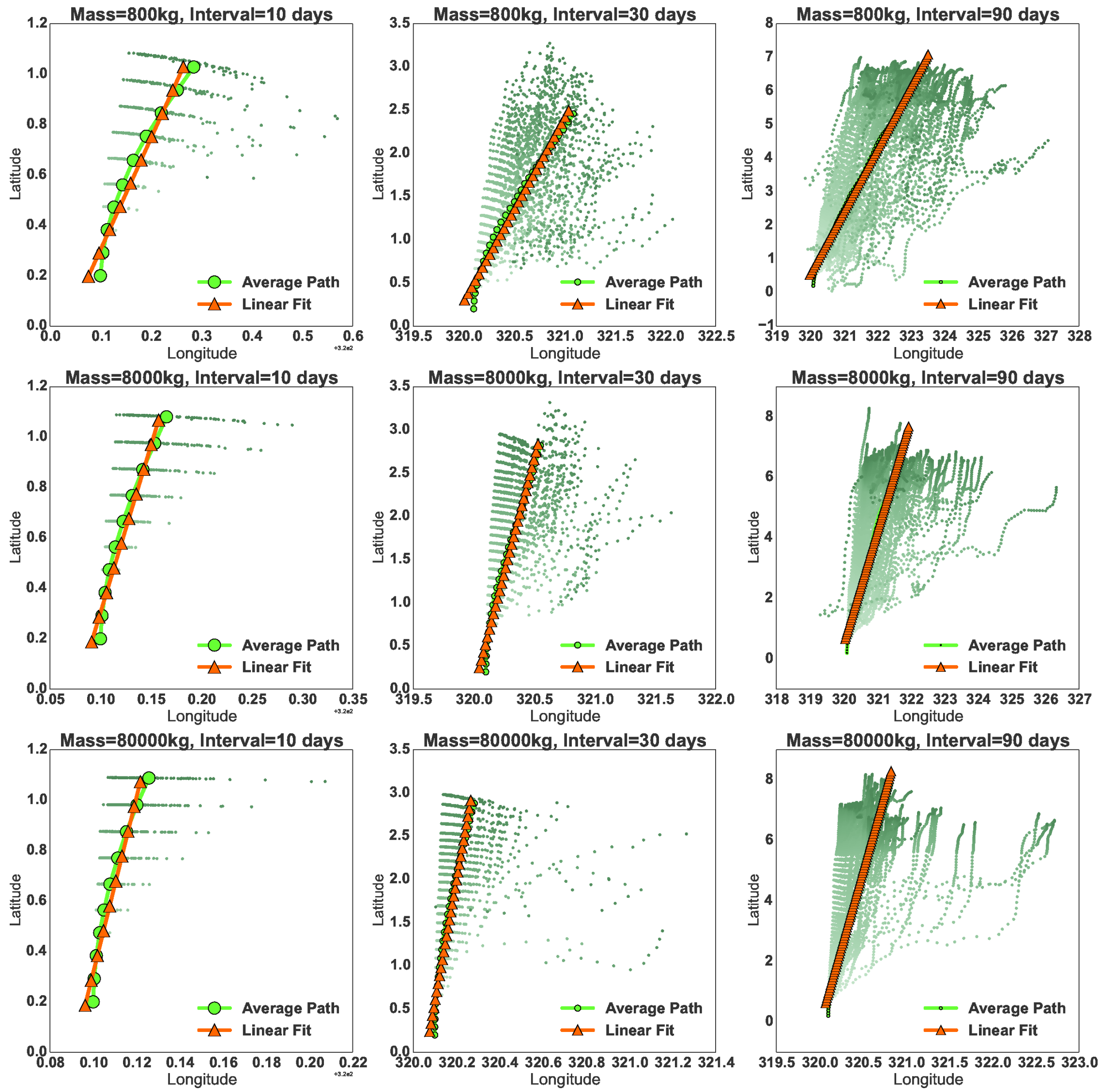

Figure 3: Simulations of debris trajectory starting from the Earth-coordinate location $(\phi_0, \lambda_0) = (0.2°, 320.1°)$. The probability-control parameter q is set to 0.4. The trajectories tend towards the North-East corner of each grid. The plot of (x_μ, y_μ) along with the parametrized regression line are overlayed.

to relax the high-level assumptions outlined above. The most challenging of these is the inclusion of ocean depth into the simulation, as the OSCAR dataset does not provide information of this type. Incorporating shape, density, and material with the mass of debris and modeling them from a "laws of bature" paradigm would be fruitful as well. Plenty of room exists for backtesting the model on old data to see if it has real predictive power over any time interval. Finally, it is worth adding the challenge of unknown crash coordinates to the model to see if it can infer useful patterns of debris trajectory that are invariant of the precise initial Earth-coordinate.

References

[1] "Ordinary Least Squares." Wikipedia. Wikimedia Foundation, 18 Jan. 2015. Web. 09 Feb. 2015. <http://en.wikipedia.org/wiki/Ordinary_least_squares>

[2] "OSCAR Near Realtime Global Ocean Surface Currents." OSCAR Near Realtime Global Ocean Surface Currents. N.p., n.d. Web. 07 Feb. 2015. <http://www.oscar.noaa.gov/>.

[3] Raymond, Jimmy. "Bilinear Interpolation Equation Formula Calculator— Double Interpolator" AJ Design, 2012. Web. 09 Feb. 2015. <http://www.ajdesigner.com/phpinterpolation/bilinear_interpolation_equation.php>.

[4] Waskom, Michael. "Seaborn: Statistical Data Visualization." Seaborn: Statistical Data Visualization Seaborn 0.5.1 Documentation. Stanford University, 2014. Web. 09 Feb. 2015. <http://stanford.edu/ mwaskom/software/seaborn/>.

[5] Veness, Chris. "Movable Type Scripts." Calculate Distance and Bearing between Two Latitude/Longitude Points Using Haversine Formula in JavaScript. Movable Type Scripts, Jan. 2015. Web. 08 Feb. 2015. <http://www.movable-type.co.uk/scripts/latlong.html>.

[6] "Unidata." Unidata. UCAR Community Programs, 2014. Web. 09 Feb. 2015. <http://www.unidata.ucar.edu/software/netcdf/>.

[7] Scipy Developers. "SciPy.org." SciPy.org SciPy.org. Enthought, 2015. Web. 09 Feb. 2015. <http://www.scipy.org/>.

[8] Numpy Developers. "NumPy." NumPy Numpy. Scipy Developers, 2013. Web. 09 Feb. 2015. <http://www.numpy.org/>.

5 Airline Press Statement

We are deeply affected and distraught by the tragedy of Malaysian Airlines Flight 370 over the Indian Ocean. Our thoughts and prayers go out to the families of the 239 passengers and crew members. As you know, we lost contact with the plane during its flight on the morning of March 8, 2014. We do not know where it crashed at sea, though search efforts have persisted through the beginning of the new year.

Finding the aircraft requires building systems to predict the likely locations that it may end up on in the ocean. This is an incredibly challenging task, but we are employing large numbers of teams to construct these exact models to aid us in our search for the plane. This infrastructure, had it existed last year, may have helped search operations find the plane. Our only consolation now, which must continue to drive us forward, is that investing in this infrastructure will help bring closure to tragedies like that of MH370, should they—Heavens forbid—happen in the future. With sophisticated models incorporating the latest data on the ocean's movements, we may be able to turn search missions into rescue operations as well.

At present, our researchers have constructed a preliminary solution, based on a mathematical model, that we believe can greatly improve the likelihood of finding planes like MH370 in the crucial days after their disappearances. The model uses state-of-the-art ocean data, allowing us to precisely estimate the motion of ocean currents and lead us to a location of the aircraft. Once we have an idea of the likely point-of-impact between the airplane and the ocean, we can accurately learn the most probable locations of the plane after up to thirty days. Through the use of powerful statistical tools, we can coordinate search missions that maximize our chances of finding the aircraft. Even after several days from the point at which we lose contact, we believe that the techniques our solution employ will help our efforts in finding the plane.

We are a community of organizations dedicated to the proposition that our passengers have not only the right to safe passage through the air, but the moral authority to demand of us our highest commitment to that right. We will continue to ensure that our services far-exceed all standards of safety. If, the highly-unlikely event that a plane go missing in the future, we believe that the solutions provided by our researchers will give importance assistance to coordinated efforts in finding the plane. We want the planes tasked with finding the aircraft at sea to succeed. The solutions presented by our researchers will help them do so.

It is with great sadness that we are forced to consider these possibilities, but it is a responsibility that those tasked with the safety of passengers across the world must embrace. Thank you.

A Pseudocode for Algorithms

Algorithm 1 Path Simulation Algorithm

procedure SimulatePath($\phi_0, \lambda_0, m, q, N$)
  $x \leftarrow [0, \cdots, 0]$ ▷ N-element array
  $y \leftarrow [0, \cdots, 0]$ ▷ N-element array
  $t \leftarrow 0$
  $\theta \leftarrow 0$
  $x_0 \leftarrow \lambda_0$
  $y_0 \leftarrow \phi_0$
  $i \leftarrow 1$
  for $i < N$ **do**
    if $0 \equiv i \bmod 5$ **then**
      $t \leftarrow t + 1$
    end if
    $(u, v) \leftarrow$ InterpolateVelocity($\phi_{i-1}, \lambda_{i-1}, t$)
    $(y_i, x_i, \theta) \leftarrow$ UpdateCoordinates($y_{i-1}, x_{i-1}, u, v, \theta, m, q$)
  end for
  return (x, y)
end procedure

Algorithm 2 Velocity Interpolation from Data

procedure InterpolateVelocity(ϕ, λ, t)
  $(\phi_1, \phi_2) \rightarrow$ (Round(ϕ) $+ \frac{1}{2}$) ▷ The northern points
  $(\lambda_1, \lambda_4) \rightarrow$ (Round(λ) $- \frac{1}{2}$) ▷ The western points
  $(\phi_3, \phi_4) \rightarrow$ (Round(ϕ) $- \frac{1}{2}$) ▷ The southern points
  $(\lambda_2, \lambda_3) \rightarrow$ (Round(λ) $+ \frac{1}{2}$) ▷ The eastern points
  if $\phi_1 = \phi_3$ or $\lambda_1 = \lambda_3$ **then**
    return $(0, 0)$
  end if
  return BiinearInterpolate($\lambda, \phi, \phi_1, \phi_2, \phi_3, \phi_4, \lambda_1, \lambda_2, \lambda_3, \lambda_4$)
end procedure

Algorithm 3 Bilinear Interpolation of Velocity Data

procedure BilinearInterpolate$(\lambda, \phi, \phi_1, \phi_2, \phi_3, \phi_4, \lambda_1, \lambda_2, \lambda_3, \lambda_4)$
  $(u_1, v_1) \leftarrow (u(\phi_1, \lambda_1), v(\phi_1, \lambda_1))$ ▷ Velocities at the Earth-coordinate (ϕ_1, λ_1),
  $(u_2, v_2) \leftarrow (u(\phi_2, \lambda_2), v(\phi_2, \lambda_2))$ ▷ at (ϕ_2, λ_2),
  $(u_3, v_3) \leftarrow (u(\phi_3, \lambda_3), v(\phi_3, \lambda_3))$ ▷ at (ϕ_3, λ_3),
  $(u_4, v_4) \leftarrow (u(\phi_4, \lambda_4), v(\phi_4, \lambda_4))$ ▷ at (ϕ_4, λ_4)
  $u = u_1 \cdot (\lambda_2 - \lambda)(\phi_2 - \phi) + u_2 \cdot (\lambda_2 - \lambda)(\phi - \phi_1) + u_3 \cdot (\lambda - \lambda_1)(\phi_2 - \phi) + u_4 \cdot (\lambda - \lambda_1)(\phi - \phi_1)$
  $v = v_1 \cdot (\lambda_2 - \lambda)(\phi_2 - \phi) + v_2 \cdot (\lambda_2 - \lambda)(\phi - \phi_1) + v_3 \cdot (\lambda - \lambda_1)(\phi_2 - \phi) + v_4 \cdot (\lambda - \lambda_1)(\phi - \phi_1)$
  return (u, v)
end procedure

Algorithm 4 Update Earth-Coordinates on Path

procedure UpdateCoordinates$(y_{i-1}, x_{i-1}, u, v, \theta, m, q)$
  if $v = 0$ **then** $\bar{\theta} \leftarrow \frac{\pi}{2}$
  else
    $\bar{\theta} \leftarrow \tan^{-1}\left(\frac{u}{v}\right)$
  end if
  $\theta \leftarrow$ ProbAngle$(\bar{\theta}, \theta, m, q)$
  $d \leftarrow \sqrt{u^2 + v^2}$
  $y \leftarrow \phi_2$ in Eq. (3)
  $x \leftarrow \lambda_2$ in Eq. (4)
  return (x, y)
end procedure

Algorithm 5 Probability of Deviating from Angle

procedure ProbAngle$(\bar{\theta}, \theta, m, q)$
  $p \in [0, 1]$ ▷ p is a random number
  $\bar{\theta} \leftarrow \theta + \frac{1}{\sqrt{m}}(\bar{\theta} - \theta)$
  if $p < q$ **then**
    return $\bar{\theta}$
  else if $p < q + \frac{1-q}{2}$ **then**
    return $\bar{\theta} + \frac{2}{3}\theta$
  else
    return $\bar{\theta} - \frac{2}{3}\theta$
  end if
end procedure