THE GLIDER GUIDANCE SYSTEM (GGS): A MODEL-GUIDED PATH-PLANNING PROGRAM FOR ADVANCING GLIDER OPERATIONS THROUGH DEPTH-AVERAGED OCEAN CURRENTS

by

SALVATORE THOMAS FRICANO

A thesis submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Master of Science

Graduate Program in Oceanography

Written under the direction of

SCOTT GLENN & OSCAR SCHOFIELD

And approved by

New Brunswick, New Jersey

October 2024

ACKNOWLEDGEMENTS

“Work hard, have fun, change the world.” This quote from Doug Webb, which hangs proudly on the walls of the Department of Marine and Coastal Sciences, has been a guiding principle for me throughout my academic journey at Rutgers.

My undergraduate advisor and Introduction to Oceanography professor Gary Taghon is someone I am forever grateful for having known. Well known for his dedication to student learning, Gary would often take the time to stay after class with me and discuss my burning scientific curiosities. Our conversations drove me to become an active and involved student which strongly supported my learning experience as a Rutgers student.

Scott Glenn, Oscar Schofield, and Josh Kohut are three professors who I have worked very closely with throughout my years at Rutgers - particularly from taking their Topics in Marine Science course for 7 semesters out of pure admiration for the learning environment they created. They have taught me to approach the world with enthusiasm and take pride in being an ocean explorer. From classrooms to conferences, I have carried their words of wisdom in high regard and will continue to do so into my future.

Additionally, I would like to thank Thomas “Motz” Grothues for his incredible mentorship as a teacher and advisor during my time as an intern at the Rutgers University Marine Field Station. Being hands-on in the field and working on impactful research with Motz helped me become confident in my skills as an oceanographer and follow my curiosities.

Working towards my Masters in Operational Oceanography has been one of the most fulfilling experiences of my academic career, and I could not have navigated its challenges without the support of our program advisor Alex Lopez. Deadlines and high-pressure workloads would have been much more daunting without his guidance and encouragement. Beyond the scientific knowledge I have gained from him, I will always remember Alex’s grounding perspective when approaching hard problems.

Scott and Oscar, who acted as co-advisors for this thesis, have truly become some of the most influential role-models in my life. Despite being established leaders in the world of oceanography and frequently traveling around the world to advocate for the advancement of ocean science, they always found time to meet with me and support my work. Every meeting I have had with them has left me more curious, inspired, and eager to pursue new knowledge and apply my skills out in the world.

Lastly, I would like to thank my incredible family for their unwavering support and unconditional love as I have worked towards my future. To my parents, Roberta and Mark, thank you for always encouraging me to reach for the stars and having confidence in everything I do. To my partner, Tanya, and our two fluffy cats, Pookie and Captain Kibbles, thank you for making our home such a bright and happy place every day.

TABLE OF CONTENTS

TABLE OF FIGURES

I. INTRODUCTION

The Global Ocean Observing System (GOOS) has identified Ocean Gliders, a class of buoyancy-driven uncrewed underwater vehicle (UUV), as a mature technology for providing a sustained sampling presence underwater.[1,2] Gliders are designed to conduct long-duration missions at sea under the control of shoreside operators.[3] Gliders utilize pilot-programmed missions to sample target areas of the ocean as they maneuver in a sawtooth-pattern beneath the waves, typically down to depths of 1000 meters. At sea, gliders are subject to advection by ocean currents that can impact their flight path. Given that buoyancy-driven flight mechanisms yield slow horizontal speeds relative to the water (approximately 20 cm/s), strong currents can be particularly problematic.[3,4] Therefore, knowing ocean currents conditions around an operating environment is vital for the success of glider operations. Surface currents alone do not represent the full advection pattern a glider may be subjected to, as it will experience the water column average current.[3] Due to a limited number of existing glider flight tools, developing modern operational methods to provide operational forecasts of depth-averaged ocean currents would be of great value.

To improve glider operational confidence, we have developed the “Glider Guidance System” (GGS) - a Python-based program that computes 2-dimensional depth-averaged current (DAC) datasets from existing 3-dimensional ocean physics models and subsequently provides navigational support products.[5] Currently, GGS utilizes the Real-Time Oceanographic Forecast System (RTOFS) model from the National Oceanic and Atmospheric Administration, the Copernicus Marine Environment Monitoring Service (CMEMS) model from the European Union, and the Global Oceanographic Forecast System (GOFS) model from the United States Navy.[6,7,8]

GGS automatically computes DAC data products according to a user-constructed configuration file which defines information about the glider mission (such as maximum working depth, mission extent, and deployment timeline). GGS offers multiple DAC visualization products including contour maps, vector field plots, and a pathfinding algorithm to support various end-user needs.

Using multiple ocean models within GGS further allows for the critical assessment of uncertainty. Inherent limitations associated with oceanographic sampling introduces the possibility for models (all driven by the same remote sensing and in-situ data) to represent conditions differently, as shown in Frederick C. Fuglisters 1957-1958 "Atlantic Ocean Atlas" (Figure A.1).[9] Disagreement serves as a proxy for areas that may contain less reliable model data which lacks in-situ confirmation and thus is a factor of consideration for pilots. Specifically, uncertainty in currents can be used to derive a range of estimates for advection, speed, and transit time which can be used to increase operational confidence in glider operations.

II. METHODOLOGY

A. AN OPEN-SOURCE PYTHON-BASED PLATFORM

Ensuring that GGS remains open access for user groups across the world was a key research and development consideration. For this reason, GGS was developed in Python using reliable and well-maintained industry-standard packages that safeguard its stability over time. Accessibility to the program is also maintained through its light-weight hardware requirements which allow most computers capable of running Python to execute GGS. Additionally, all scripts, packages, and documentation related to GGS can be accessed or publicly contributed to through the main GitHub repository.[5]

B. USER-CONFIGURATION SETUP

When executed, GGS automatically runs in accordance with a JavaScript Object Notation (JSON) configuration file. GGS contains a template JSON file which is filled out by the user to define the deployment details and desired product outputs. Within the configuration file, there are 5 object categories: MISSION, MODEL, PRODUCT, DATA, and ADVANCED. Each object has a general scope for the various keys under them. A comprehensive outline of the current keys and associated objects within the GGS configuration file are described further in Appendix B and on GitHub.[5]

C. DEPTH-AVERAGING CALCULATIONS

Dynamical ocean models are sophisticated computational tools which simulate and forecast oceanographic conditions based on numerical approximations to the physical equations of motion.[10] Ocean physics equations depend upon the horizontal and vertical gradients created by 4-dimensional fields of velocity, pressure and density.[11] These gradients are influenced by variations in temperature, salinity, and pressure across space and drive dynamic processes in the ocean such as currents.[11]

Ocean models such as RTOFS, CMEMS, and GOFS calculate data at strong gradient interfaces which results in unevenly spaced grid points over depth. Stratified upper layers which experience atmospheric interactions and river plume inputs have a comparatively higher concentration of data points to the deep ocean where conditions are more uniform, and gradients are less pronounced.[12]

To correctly calculate the depth-average values for grid points in ocean model datasets, GGS utilizes a 1-meter resolution linear interpolation calculation. Interpolation is a common and practical approach for calculating intermediate data between two known points and, in the context of ocean models, allows for a properly weighted depth-average to be efficiently calculated. Programmatically, interpolation calculations can be applied to grid points in different ways. Utilizing a vectorized process which simultaneously applies the interpolation model equations to each grid point within chunked subsets (batches of data processed as one unit to avoid hardware overloading) ensures that GGS runs the depth-averaging calculations in an extremely time-efficient manner.

For each of the models toggled on for processing in the configuration file, GGS acquires data for the mission extent area and subsets it between 0 meters and ‘Z’ meters, where ‘Z’ is the maximum working depth of the glider. In cases where ‘Z’ is an intermediate depth value between two depth indices at which an ocean model calculates data points, GGS will extend the depth subset down to the next-deepest index. For the valid subset range, GGS performs a vectorized interpolation operation to the ocean model variables ‘u’ (the eastward current component) and ‘v’ (the northward current component) across the entire model grid. For each 1-meter bin, the ‘u’ and ‘v’ values are additionally used to calculate the magnitude and direction of the current for each bin. After performing the interpolation and calculating data points for each 1-meter position in the water column, the interpolation model checks if the range of interpolated data exceeds the maximum working depth (resultant from the case where ‘Z’ is an intermediate depth value in relation to the ocean models depth indices) and clips the bins to the operational range of the glider. Once the DAC bins have been calculated and checked, a depth-average is calculated from the bins to yield the depth-average (Figure 1).

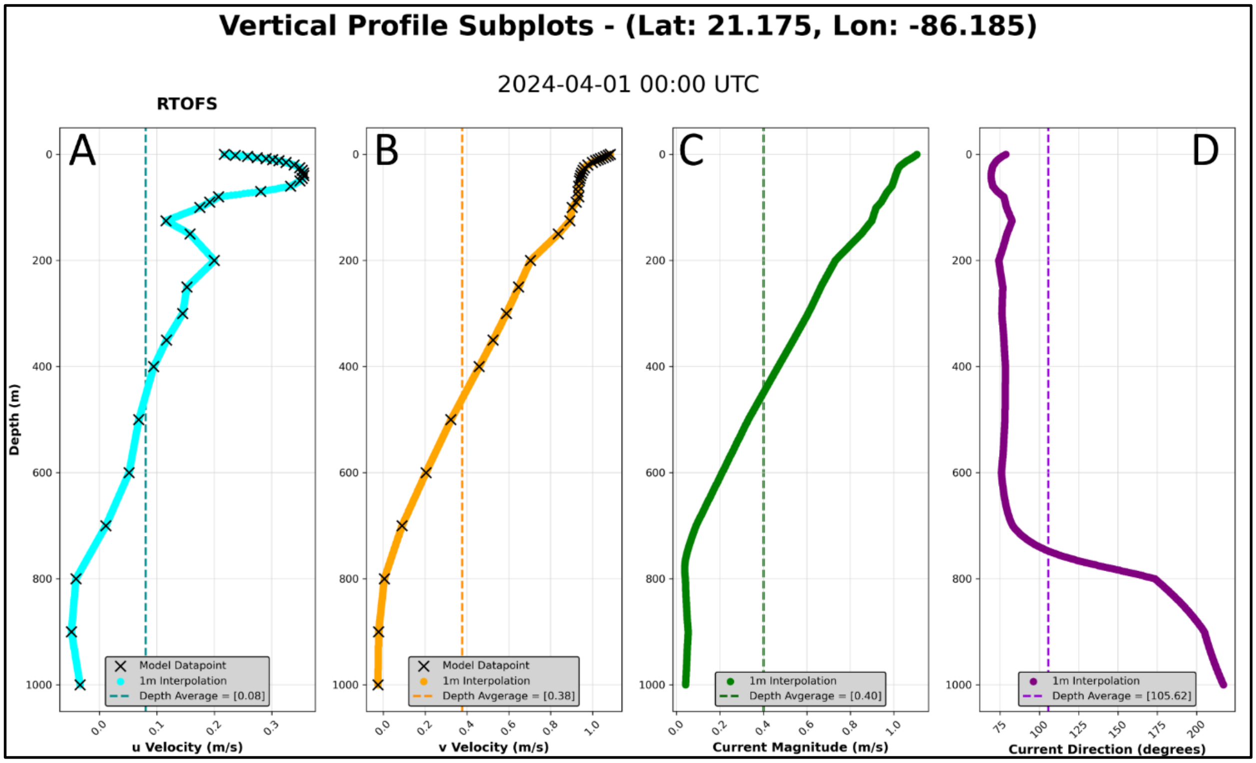


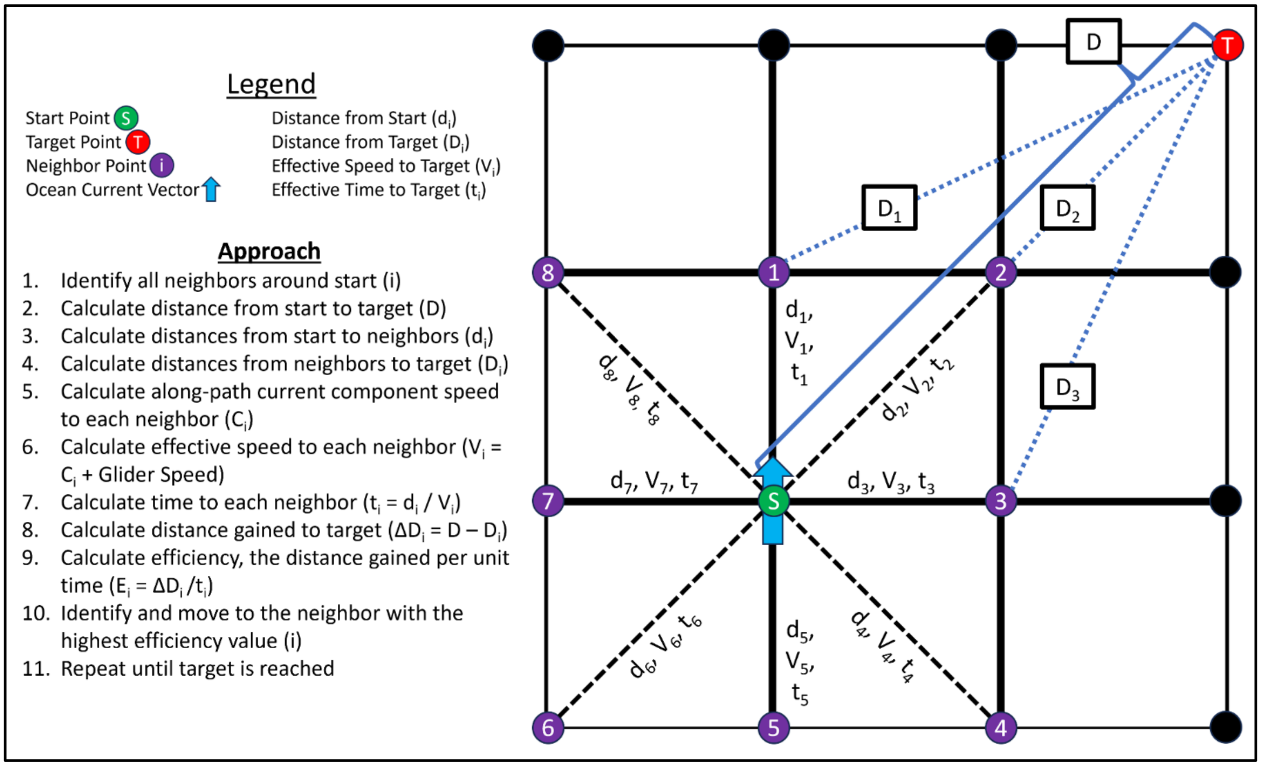
Figure 1: Vertical profile scatter plot output from GGS depicting RTOFS model and DAC data. Subplots A and B depict the variables ‘u’ and ‘v’, respectively. Black “X” markers indicate the positions of RTOFS ocean model data points, colored scatter points (appearing as a connected line due to the data density) indicate the 1-meter interpolated bins, and vertical dashed lines indicate the value of the depth-average. Subplots C and D depict the variables ‘magnitude’ and ‘direction’, respectively, and show colored scatter points for the 1-meter interpolated bins and vertical dashed lines for the value of the depth-average.

D. PRODUCT OUTPUT PROCESSING

For each ocean model processed, GGS outputs the raw ocean model data, the 3-dimensional 1-meter resolution interpolated bin dataset, and the 2-dimensional DAC dataset. Based on the configuration file, GGS utilizes these datasets to create navigational support products. Currently, the DAC product suite offered by GGS includes: 1) a Python contour “Streamplot” map (Figure 3A, below), 2) a Python threshold zone contour Streamplot map (Figure 3B, below), 3) vertical profile scatter plots (Figure 1, above), 4) geographic information system (GIS) GeoPackage files, and 5) a pathfinding algorithm which can identify time-efficient routes for a glider to take between waypoints (Figure 2, below). All products within GGS can be used to support glider missions which are actively deployed or being planned for future deployment, allowing them to address different case uses. For missions which are underway, GGS can acquire live glider data through Environmental Research Division Data Access Program (ERDDAP) glider data server using its dataset ID to obtain glider sensor values and overlay tracklines.[13]

E. PATHFINDING ALGORITHM METHODS

Conceptually, the pathfinding algorithm within GGS is a modified version of the A* search algorithm – an established algorithm framework with applications for finding efficient paths between nodes in a 2-D velocity grid.[14] A* algorithms are designed to compute a path which minimizes the sum of a cost variable (such as time) when evaluating step-wise movements in a grid.[14] Within GGS, the pathfinding algorithm has been adapted to incorporate DAC data grids into its analysis (which has known latitude, longitude, depth-averaged ‘u’, and depth-averaged ‘v’ values at each grid point) identifying time-efficient route(s) between mission waypoints.



Legend

Start Point S
Target Point T
Neighbor Point i
Ocean Current Vector ⬆

Distance from Start (d_i)	
Distance from Target (D_i)	
Effective Speed to Target (V_i)	
Effective Time to Target (t_i)	

Approach

1. Identify all neighbors around start (i)
2. Calculate distance from start to target (D)
3. Calculate distances from start to neighbors (d_i)
4. Calculate distances from neighbors to target (D_i)
5. Calculate along-path current component speed to each neighbor (C_i)
6. Calculate effective speed to each neighbor (V_i = C_i + Glider Speed)
7. Calculate time to each neighbor ($t_i = d_i / V_i$)
8. Calculate distance gained to target ($\Delta D_i = D - D_i$)
9. Calculate efficiency, the distance gained per unit time ($E_i = \Delta D_i / t_i$)
10. Identify and move to the neighbor with the highest efficiency value (i)
11. Repeat until target is reached

Figure 2: Diagram example for a single-step calculation in the GGS pathfinding algorithm depicting the calculations and decision-making process to find the optimal approach path to a Target Point. In practice, the Target Point location may be many more grid points away than in this compact illustration.

Starting from an initial node (a grid point in the DAC dataset), the algorithm first identifies all neighboring nodes in the grid as possible movement locations. For each neighbor, the algorithm calculates: 1) the distance traveled from starting point (d_i), 2) the remaining distance to target point (D_i), 3) the along-track current component (C_i), 4) the effective glider speed (V_i), and 5) the estimated transit time to the neighbor (t_i). Ultimately, this information is used to identify the neighbor node which holds the highest 'efficiency rating' (E_i, the greatest distance gained per unit time). A visual diagram of this incremental process and its associated equations can be found in Figure 2 and the GitHub algorithm documentation file, respectively.[5]

III. RESULTS & APPLICATIONS

A. SUPPORTING THE NATIONAL ACADEMIES

GGS was used to support active glider operations contributing to the National Academies Understanding Gulf Ocean Systems (UGOS) Initiative from January to April of 2024. At the start of the UGOS project, DAC magnitude contour Streamplots were the only two navigational support products available in GGS (Figure 3A). These products were produced daily for each 6-hour forecast model dataset made available by RTOFS, CMEMS, and GOFS.

The goal of these support products was to assist in the identification of zones which were optimal to fly gliders through in real-time. Static plots were hosted online for pilots across the UGOS partnership to reference, and DAC netCDF files were pushed to the Woods Hole Group's MetOcean Mapper which allowed for layered viewing of DAC fields against other data stream sources.[15,16] Over the course of the project, demand for a map product which highlighted strong current zones had become prevalent. This interest among users developed into the threshold zones contour plot product (Figure 3B). After developing the threshold zones plot product, we found that it not only aided in highlighting zones where the DAC magnitudes began to approach and surpass the gliders baseline speed but further highlighted zones of uncertainty where different ocean models had strong disagreement.

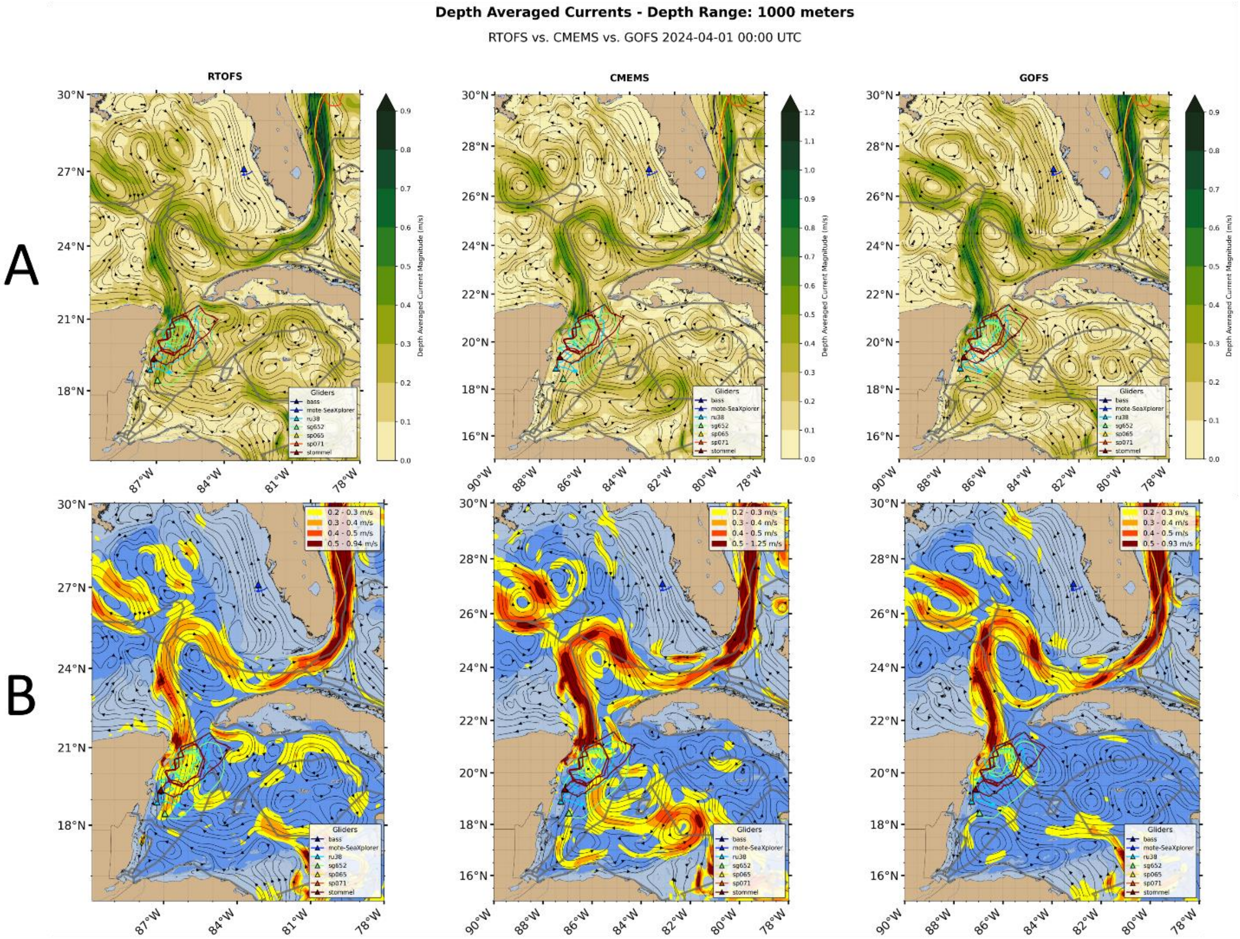
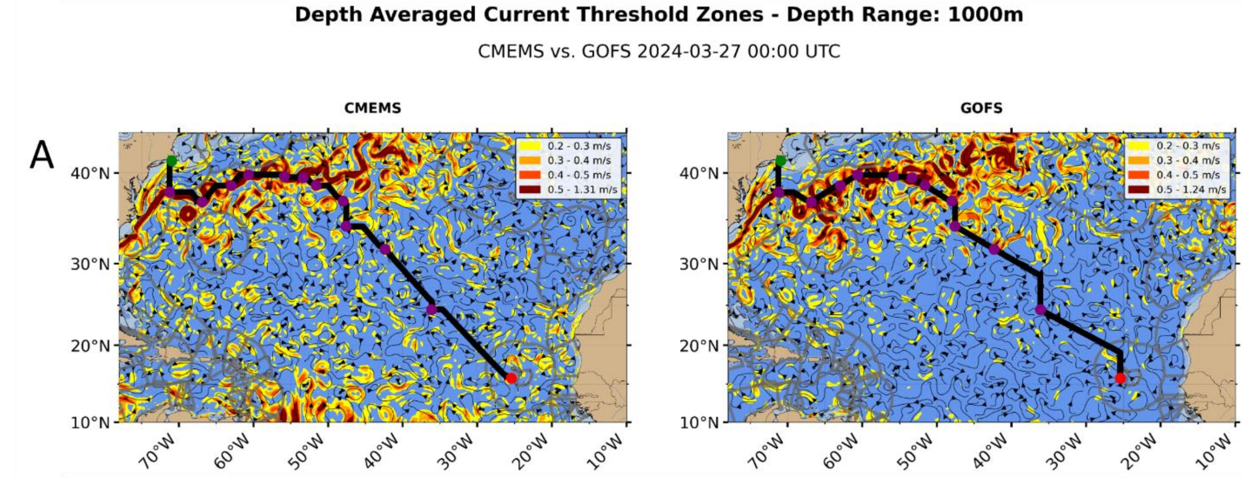


Figure 3: Threshold zones contour plots (top row, A) and magnitude contour plots (bottom row, B) output from GGS showing DAC for RTOFS (left column) CMEMS (middle column) and GOFS (right column). Threshold zone contour plots depict regions of increasing DAC magnitude from 0.2 m/s to >0.5m/s at 0.1 m/s level intervals. Magnitude contour plots depict DAC magnitude for increasing levels, with color bars adjusted per-model based on range.

B. ADVANCING PATH PLANNING METHODS FOR THE SENTINEL MISSION

Beyond UGOS, GGS was used to identify the optimal route for the Sentinel Mission - the first global circumnavigation mission by an underwater glider, championed by “Redwing”.[17] In cooperation with undergraduate students from the Rutgers University Department of Marine and Coastal Sciences ‘Topics in Marine Science’ research course, DAC maps for different ocean basins along the route were produced and analyzed. Primarily, CMEMS and GOFS were used to process DAC for the Sentinel Mission as RTOFS selectively converts the global binary ocean model outputs into different netCDF files for US-centric regions. From CMEMS and GOFS, DAC magnitude contour plots and threshold zone contour plots were used to highlight strong ocean current zones (termed “ocean highways”) which flow along the general heading between waypoints (Figure 4A). After evaluating the DAC fields within each operational leg of the circumnavigation, advantageous waypoints that positioned Redwing to follow favorable ocean highways were needed. Because visually extracting waypoint coordinates from the static map products (which only have XY axis tick marks to reference) would be inexact, GGS was expanded to provide GIS GeoPackage data files which allow the DAC data to be evaluated precisely in a spatial frame of reference. Utilizing this capability, a list of waypoints was collaboratively determined for the mission. These waypoints target major ocean highway features (such as the Gulf Stream (Figure 4A) and Antarctic Circumpolar Current) which CMEMS and GOFS showed approximate agreement on.

Calculating estimations for the duration of the circumnavigation was a critical step in the logistics planning for the Sentinel Mission. To make effective calculations which consider the effect of currents during transit, we developed and integrated the GGS pathfinding algorithm to identify efficient routes between waypoints and calculate approximations for transit time.



B

CMEMS	**LEG 1**	**LEG 2**	**LEG 3**	**LEG 4**	**LEG 5**	**LEG 6**	**LEG 7**	**TOTALS**
Time (days)	90	217	238	122	216	85	297	1265
Distance (kilometers)	5933	8111	9734	7551	10405	4489	13038	59261
GOFS	**LEG 1**	**LEG 2**	**LEG 3**	**LEG 4**	**LEG 5**	**LEG 6**	**LEG 7**	**TOTALS**
Time (days)	97	236	221	159	267	125	301	1406
Distance (kilometers)	5920	8108	9748	7541	10399	4491	13032	59239

Figure 4: Threshold zone contour plots (A) output by GGS for the first leg of the Sentinel Mission, from Cape Cod (green point) to Cabo Verde (red point). Intermediate purple points on the map depict the collaboratively determined waypoints for the mission based on advantageous current fields, such as the Gulf Stream. Black interconnecting lines between waypoints depict the optimal route calculated by the pathfinding algorithm within GGS. Table B displays the pathfinding algorithms calculated time and distance estimations for each leg of the mission.

Disagreement between ocean models proved to be a critical component of planning the Sentinel Mission route. Transit times output by the algorithm for CMEMS and GOFS across a multi-day data composite were used to estimate the expected duration for each leg of the Sentinel Mission with associated error ranges to compensate for their inherent differences. We ultimately calculated that CMEMS predicted a circumnavigation time of 3.4 years whereas GOFS estimated 3.9 years (Figure 4B).

IV. DISCUSSION

A. OUTCOMES AND LIMITATIONS

Applying GGS to the UGOS initiative and the Sentinel Mission showed that GGS satisfied two major use cases for supporting underwater glider missions. The first use case, as seen during UGOS, is supporting glider operations that are underway and providing DAC navigational support products to make informed piloting decisions. The second use case, as seen in the Sentinel Mission, is planning future missions around DAC fields for an operational area to evaluate advantageous and/or disadvantageous regions of ocean currents.

Co-designing GGS alongside users drove many of the new features and functionalities that were implemented into the program. UGOS largely contributed to the growth of additional DAC map products which prioritized user-readability and understanding for where strong currents relative to the glider's speed are in the operational area. Further, the Sentinel Mission highlighted the need for algorithmic and analytical tools such as the pathfinding algorithm and its associated transit time estimations. Across both projects, GGS was iteratively made more efficient. Specifically, the use of vectorized calculations has reduced the per-model DAC calculation time to only a matter of minutes on most hardware.

Time constraints developing GGS led to some limitations in the scope and functionality and products in the program. One constraint in GGS is the availability of the RTOFS model data in operational zones beyond existing regional netCDF file subsets. While this did not impact the UGOS pilots who were operating around the Yucatan peninsula, it did impact the Sentinel Mission by preventing RTOFS data from being evaluated along with CMEMS and GOFS data. Furthermore, limitations in the complexity of the pathfinding algorithm make it sub-optimal for some real-world expectations. Currently, the pathfinding algorithm only utilizes the directional component of the current which is parallel to the gliders heading along grid-based movements to compute the effective speed of the glider. Resulting from this, strong cross-track components of DAC vectors are not regarded which limits the algorithm's ability to gauge off-course advection. However, most glider operations are planned around the utilization of current fields which follow the glider's intended direction of travel. Therefore, the algorithms disconnect from cross-track current components is negligible for most mission plans and only hinders the accuracy of paths which intentionally cross strong current fields.

B. FUTURE EXPANSIONS

Creating an averaged/blended product from the RTOFS, CMEMS, and GOFS datasets is a target for future work. By interpolating each of the models onto a common grid to comprehensively display the average DAC value across each model, additional products which directly highlight zones of disagreement can be made. Taking this concept a step further, using live ERDDAP sensor values from glider data to feed GGS reference DAC values would allow for a dynamic adjustment in the weight of each model’s contribution to a blended DAC product.

Addressing the cross-track component in the pathfinding algorithm is another point of growth for improving the accuracy of automated flight decisions that can be made with navigational tools from GGS. Applying this type of operational consideration for how a glider would need to fly into strong currents by slightly offsetting its heading to cancel advection forces would be an effective way for the algorithm to adjust the optimal path and further increase the accuracy of its effective speed and time estimations.[18]

REFERENCES

1. Intergovernmental Oceanographic Commission, OceanOPS, GOOS Observations Coordination Group, United Nations Environment Programme, World Meteorological Organization, & International Science Council. (2023). Ocean Observing System report card 2023 (IOC/BRO/2023/5). https://www.ocean-ops.org/
2. Global Ocean Observing Networks – Global Ocean Observing System. (n.d.). https://goosocean.org/who-we-are/observations-coordination-group/global-ocean-observing-networks/
3. Schofield, O., J. Kohut, D. Aragon, L. Creed, J. Graver, C. Haldeman, J. Kerfoot, H. Roarty, C. Jones, D. Webb and S. Glenn (2007). "Slocum Gliders: Robust and ready." J. Field Robotics 24: 473-485.
4. Ramos, A. G., V. J. García-Garrido, A. M. Mancho, S. Wiggins, J. Coca, S. Glenn, O. Schofield, J. Kohut, D. Aragon, J. Kerfoot, T. Haskins, T. Miles, C. Haldeman, N. Strandskov, B. Allsup, C. Jones and J. Shapiro (2018). "Lagrangian coherent structure assisted path planning for transoceanic autonomous underwater vehicle missions." Scientific Reports 8(1): 4575.
5. Fricano, S. Glider Guidance System (GGS) [Python program]. https://github.com/STF33/Glider-Guidance-System/
6. RTOFS HYCOM - Global - US East Coast. Regional Volume - 1/12 Degree High Vertical Resolution. National Oceanic and Atmospheric Administration (NOAA). Real-Time Oceanographic Forecast System (RTOFS). URL: https://www.nco.ncep.noaa.gov/pmb/products/rtofs/#RTOFS
7. Global Ocean Physics Analysis and Forecast. E.U. Copernicus Marine Service Information (CMEMS). Marine Data Store (MDS). DOI: 10.48670/moi-00016
8. Global Ocean Forecasting System (GOFS) 3.1: 41-layer HYCOM + NCODA Global 1/12° Analysis. Naval Research Laboratory: Ocean Dynamics and Prediction Branch. URL: https://www.hycom.org/dataserver/gofs-3pt1/analysis

9. Fuglister, F. C. (1960). Atlantic Ocean atlas of temperature and salinity profiles and data from the International Geophysical Year of 1957-1958.
10. Pinardi, N., L. Cavaleri, G. Coppini, P. De Mey-Frémaux, C. Fratianni, J. Huthnance, P. Lermusiaux, A. Navarra, R. Preller and S. Tibaldi (2017). "From weather to ocean predictions: An historical viewpoint." Journal of Marine Research 75: 103-159.
11. Knauss, J. A., & Garfield, N. (2016). Introduction to Physical Oceanography (3rd ed.). Waveland Press.
12. Munk, W. and C. Wunsch (1998). "Abyssal recipes II: energetics of tidal and wind mixing." Deep Sea Research Part I: Oceanographic Research Papers 45(12): 1977-2010.
13. ERDDAP Integrated Ocean Observing System (IOOS) Glider Data Server. U.S. Integrated Ocean Observing System. Available online: https://data.ioos.us/gliders/erddap.
14. Ferguson, D., M. Likhachev and A. Stentz (2005). A Guide to Heuristic-based Path Planning.
15. Glider Guidance System, Yucatan Mission product access page. URL: https://rucool.marine.rutgers.edu/hurricane/GGS_Yucatan/
16. MetOcean Mapper, EddyWatch. Woods Hole Group, Inc., A CLS Company. Available online: http://www.woodsholegroup.com.
17. Teledyne Marine. "The Sentinel Mission: Charting New Paths in Ocean Exploration." Teledyne Marine. Accessed 6/1/2024. https://www.teledynemarine.com/en-us/news/Pages/The-Sentinel-Mission-Charting-New-Paths-in-Ocean-Exploration.aspx.
18. Boeing. (2021). 737 NG Flight Crew Training Manual (Rev. 20). The Boeing Company. Document No. FCT 737 NG (TM).

Appendix A

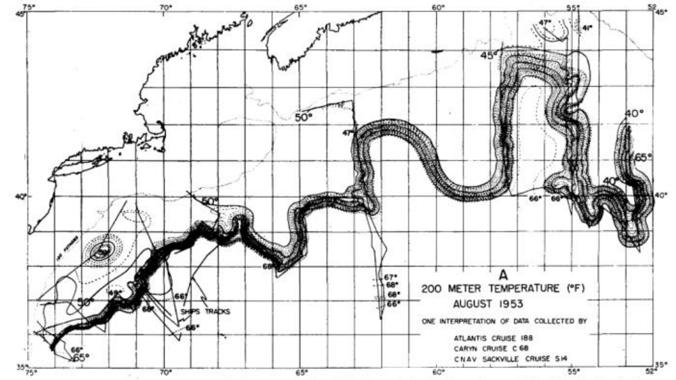


Chart 3a. An analysis of the temperature field (degrees F) at a depth of 200 metres in the Gulf Stream between 75° and 52° West longitude.

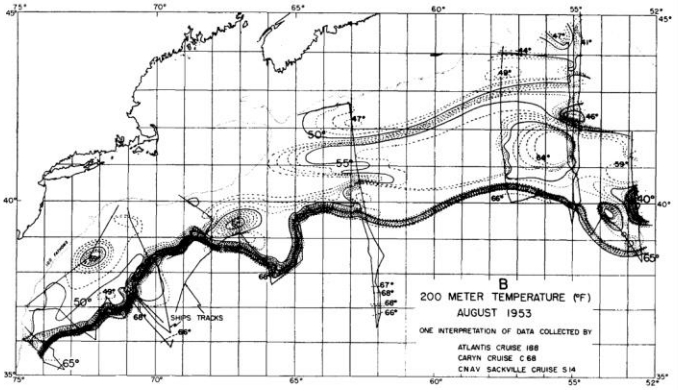


Chart 3b. An analysis of the temperature field (degrees F), at a depth of 200 metres in the Gulf Stream between 75° and 52° West longitude.

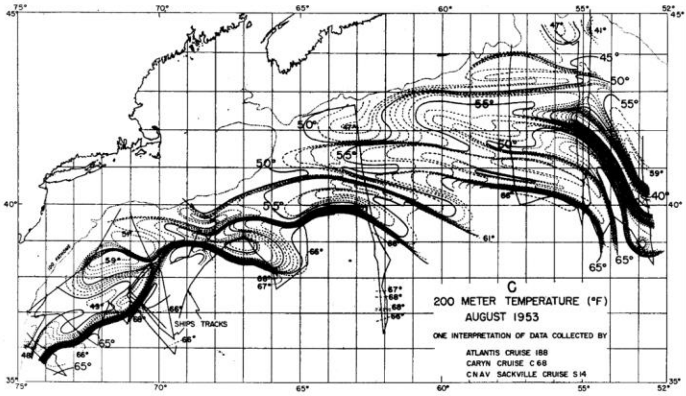


Chart 3c. An analysis of the temperature field (degrees F), at a depth of 200 metres in the Gulf Stream between 75° and 52° West longitude.

Figure A.1: Charts 3a, 3b, and 3c from Frederick C. Fuglister's book "Atlantic Ocean atlas of temperature and salinity profiles and data from the International Geophysical Year of 1957-1958". These charts show different equally representative ways in which temperature profile data (taken during a 1953 *research cruise in the Gulf Stream) can be interpreted, highlighting the disagreement possible in the various methods that we still see today (See Figures 2 & 4).*

Appendix B

Glider Guidance System (GGS) Configuration Guide

MISSION object (7 keys)

- mission_name: (String) The name of the mission.
- target_date: (String) The target date for the mission data. Use null for the current date and time, or specify a date in the format "YYYY-MM-DD HH:MM:SS".
- max_depth: (Integer) The maximum depth for the mission in meters.
- extent: (Array of Arrays) Geographic bounding box for the mission, specified as [[Min Lat, Min Lon], [Max Lat, Max Lon]].
- GPS_coords: (Array of Arrays) Specific GPS coordinates of interest, specified as [[Lat 1, Lon 1], [Lat 2, Lon 2], ...]. Use null for None.
- glider_id: (String) The ERDDAP glider ID to track. Use null for None. Note: Setting a target glider will override the extent with one created around the last position of the target glider.
- glider_buffer: (Float) The buffer value in decimal degrees used to create the extent around the target glider. Use null for None.

MODEL object (8 keys)

- single_datetime: (Boolean) Set to true to process a single date-time, false otherwise.
- enable_rtofs: (Boolean) Set to true to enable the RTOFS model, false to disable.
- enable_cmems: (Boolean) Set to true to enable the CMEMS model, false to disable.
- enable_gofs: (Boolean) Set to true to enable the GOFS model, false to disable.
- chunk: (Boolean) Set to true to enable data chunking for increased performance, false otherwise.
- save_model_data: (Boolean) Set to true to save acquired model data, false otherwise.
- save_depth_average: (Boolean) Set to true to save computed depth-average data, false otherwise.
- save_bin_average: (Boolean) Set to true to save computed bin-average data, false otherwise.

PRODUCT object (15 keys)

- create_magnitude_plot: (Boolean) Set to true to create magnitude plots, false otherwise.
- create_threshold_plot: (Boolean) Set to true to create threshold zone plots, false otherwise.
- create_advantage_plot: (Boolean) Set to true to create advantage zone plots, false otherwise.
- create_profile_plot: (Boolean) Set to true to create profile plots, false otherwise.
- create_gpkg_file: (Boolean) Set to true to create GeoPackage files, false otherwise.
- latitude_qc: (Float) Latitude for quality control plotting.
- longitude_qc: (Float) Longitude for quality control plotting.
- density: (Integer) Density of the streamplot.
- mag1 - mag5: (Float) Thresholds for magnitude levels in the plot.
- tolerance: (Float) Advantage zone tolerance in degrees.
- show_gliders: (Boolean) Set to true to show gliders on the plot, false otherwise.
- show_route: (Boolean) Set to true to show the glider route, false otherwise.
- show_eez: (Boolean) Set to true to show Exclusive Economic Zones (EEZ), false otherwise.
- show_qc: (Boolean) Set to true to show quality control markers, false otherwise.
- manual_extent: (Array of Arrays) Manual specification of plot extent, specified as [[Min Lat, Min Lon], [Max Lat, Max Lon]]. Use null for automatic.

DATA object (2 keys)

- bathymetry_path: (String) Path to the bathymetry data file.
- eez_path: (String) Path to the Exclusive Economic Zones (EEZ) shapefile.

ADVANCED object (1 key)

- reprocess: (Boolean) Set to true the reprocessing of netCDF files in the local '/data/reprocess' folder, false otherwise.