SIMROUTE USER'S MANUAL

SIMROUTE:
A WEATHER SHIP ROUTING SOFTWARE FOR ACADEMIC PURPOSES

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Table of Contents

List of figures

Supported by THE NIPPON FOUNDATION

By: The SIMROUTE team

Barcelona School of Nautical Studies
(FNB) Universitat Politècnica de Catalunya –
BarcelonaTech Pla de Palau, 18, 08003
Barcelona, Spain
e-mail: simroute@fnb.upc.edu

SIMROUTE INITIATION MANUAL

1 WHAT IS SIMROUTE?

SIMROUTE is software designed for maritime route optimizations which carries out simulations of short and long distances. This software has been developed by a team of researchers at the Universitat Politècnica de Catalunya (UPC-BarcelonaTech) in Matlab language. The spatial scope of SIMROUTE is the Mediterranean Sea, fully or partially selected, depending on the longitudes and latitudes introduced by the user. In order to run the program, there are several parameters to introduce: port of departure, port of arrival, vessel speed and the wave field for the days of the voyage to simulate.

SIMROUTE is a program composed by several scripts and Matlab functions which, when executed sequentially, allow for the definition of a computational mesh according to the selected longitudes and latitudes. Furthermore, it also allows for the definition of start and end nodes in order to carry out the route optimization between these two nodes.

The SIMROUTE's main and most used components are described herein together with the required inputs for each of them.

2 SIMROUTE AS A TEACHING TOOL (STCW FRAMEWORK)

The pedagogic purpose of this software is to provide skills on ship routing optimization, to assess the impact of the meteo-oceanographic variables (such as wind, waves or currents) on ship navigation and to highlight the relevance of ship routing in terms of sailing time, fuel consumption and harmful emissions for the environment. This software is very novel in the framework of teaching innovation in MET institutions.

From an academic point of view, this software deals some specific topics that are part of the syllabi of Maritime Education Training (MET) institutions' programs and that are included in STCW 95/2010 Code (IMO 2011). The STCW Convention 1978 has been amended by the 2010 Manila Amendments and articulates the minimum standards of competence required for seagoing personnel in detail in a series of tables (part A of the Code).

The academic modules that Simroute software includes meet the requirements of the STCW Code and provide schools with a modern way to assess STCW competences. A particular emphasis is done in terms of air emissions and safety of navigation. For instance, this software could be included in existing study programs of the Bachelor's and Master's degrees of Nautical Science and Maritime Transport in subjects such as "Routes and Compasses," "Nautical Meteorology and Oceanographic," or "Bridge Procedures" among others. The software could be also used to assess some specific topics related to Marine Environment Protection.

In a more general context, this software should be considered as a progressive step forward to develop best practices for a more efficient and sustainable maritime industry and to provide maritime industry with better-educated seafarers.

3 SIMROUTE SCRIPTS

The scripts used by SIMROUTE are the following ones, which are listed in the correct order of launching:

a. start.m
b. make_mesh.m
c. make_waves_Copernicus.m
d. find_ports.m
e. simroute_Opt.m
f. simroute_Dmin.m
g. plot_routes_and_waves.m

a. start.m

This is an initializing script that links the required folders which will be used afterwards.

b. make_mesh.m

This script creates the Mediterranean Sea computational mesh defining the maximum and minimum longitudes and latitudes. Furthermore, it can determine the computational mesh resolution results by altering the quantity of nodes as desired.

c. make_waves.m

This script uses the *Copernicus* data and the file created by *make_mesh.m* for creating a .mat file with the waves and their direction which are interpolated at the created mesh. Furthermore, it defines the vessel's time of departure from the selected port of origin. Therefore, it is necessary to define the initial time of the trip and the name of the file (or files) downloaded from *Copernicus*.

When looking at the *make_waves_Copernicus.m* file, the use of the correct wave files has to be ensured. The wave files (in *netcdf* format), downloaded from *Copernicus* website, (see direction in the .m file header), have to be introduced in the *make_waves_Copernicus.m* script. Each file poses the waves conditions (wave height and direction) of one entire day (hourly data). For instance, if the path of the route takes 80 hours and only 3 wave scripts (accounting for 72 hours) have been introduced, the software will reply "error". Once the correct wave files are introduced and the estimated time of departure (ETD) is established, an output wave script is obtained and it is used for running, *simroute_Opt.m* file.

d. find_ports.m

Using *find_ports.m* script, the initial and end nodes (ports) may be easily determined. When the user introduces the latitude and longitude of a point on the map, this function gives the node which represents these coordinates on the mesh. The node initial and the node end represent the port of departure and the port of arrival and will be needed for the simroute_Opt script.

The function *find_ports.m* lets the user know whether the selected node is situated at land or at sea.

e. Simroute_Opt.m

This main script calculates the optimum route taking into account the wave information interpolated from start and end nodes. When *simroute_Opt.m* is executed the initial speed of the vessel (v0) is modified taking into account the wave field of the optimized function cost (i.e. sailing time).

The required inputs for *simroute_Opt.m* are:

- Start node
- End node
- Name of the file generated by *make_waves.m*
- Vessels cruising speed
- Define whether Dijkstra o A* algorithm is going to be used (algorithms description included in SIMROUTE Technical Manual). (A* is selected by default)

f. simroute_Dmin.m

Taking advantage of the files generated by *simroute_Opt.m* and *make_waves_Copernicus.m*, this script carries out the calculations for the minimum distance route to be compared with the optimized routes.

Through the output scripts that *simroute_Opt.m* creates, *simroute_Dmin.m* can be run and the results of the shortest path route with and without waves, in hours, will be obtained.

In addition to the aforementioned scripts, there are 3 scripts which are very useful when there is no wave data in a strait between islands as this sea can be interpolated:

g. make_sea_step1
h. ake_sea_step2
i. make_sea_step3

g. **make_sea_step1.m**

Together with *make_sea_step2.m*, this script is used to open the straits zones where there should be water but there is not because of the interpolation made by *make_waves.m*.

make_sea_setp1.m allows the user to select which part of the Mediterranean Sea is going to be analyzed.

h. **make_sea_step2.m**

After executing former script, the *make_sea_step2.m* allows the user to define the specific rectangle inside zone designated on the first step where the new and more detailed interpolation is going to be done in order to see the wave fields on those points.

The scripts *make_sea_step2_ew.m* and *make_sea_step2_ns.m* can be differentiated by means of the direction of interpolation on the rectangle, being the first one done from east to west and the second one from north to south (that is opening a channel horizontally or vertically).

i. **make_sea_step3.m**

By executing this script, a new map will be obtained showing the strait zone with interpolated Sea.

Furthermore, there are 4 very useful functions for viewing the results:

k. plot_routes
l. plot_routes_and_waves
m. plot_waves_from_nc
n. plot_waves_interpolated

j. plot_routes.m

When the user gives to this script the resulting files from applying *simroute_Opt.m* and *simroute_Dmin.m*, this script carries out the representation of both routes on the zone of the Mediterranean Sea previously defined on *make_mesh.m*. It only represents the coast line and the routes.

k. plot_routes_and_waves.m

This second function represents and saves the wave fields evolution together with the route in an hourly manner until getting to the end node.

l. plot_waves_from_nc.m

This function plots the wave field when a *netcdf* file is given.

m. plot_waves_interpolated.m

This function plots the interpolated wave fields.

There is another function related to the emission of pollutants, described below:

n. Make_Emissions_Delta_2021

This script has been developed to calculate the fuel consumption and the emission of pollutants per trip (SO_2, NO_X, CO_2 and PM).

When the user gives to this script the resulting files from applying *simroute_Opt.m* and *simroute_Dmin.m*, this script carries out the calculation of above parameters.

The user has to introduce a list of variables which are ship specific, as explained in section 4 of Simroute Technical Manual.

4. SIMROUTE FLOW CHART

What has been stated in former paragraphs can be summed up in a flow chart as the one shown below. This flow chart shows the order of execution, together with the names of the scripts and functions which control different actions.

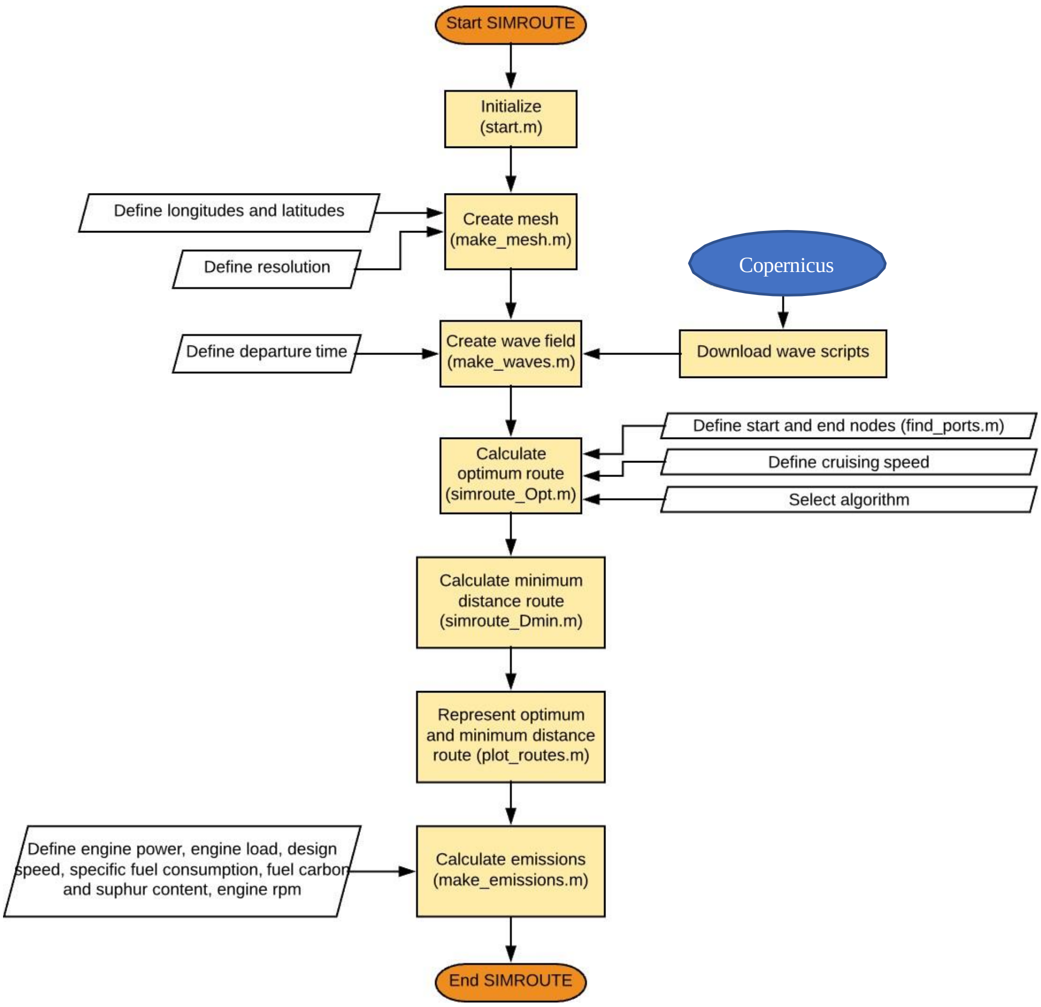


Figure 1: Simroute flow chart

5. USING THE PROGRAM STEP BY STEP

1. Download documents with .m extension from the Drive (or Atenea) copying folders (ie., Folders: IN, OUT, TOOLS_PLOTS, FUNMATLAB) together with the rest of .m files.

   - IN folder: composed by 5 general program input files. In this folder, there is also the .nc files (waves files) which were downloaded from the Copernicus Marine webpage: https://marine.copernicus.eu
   - OUT folder: 4 files which are generated as program outputs.
   - TOOLS_PLOTS folder: 4 files for graphic tools.
   - FUNMATLAB folder: files of functions used by the program.

2. From the Copernicus Marine Services web page, search the wave file to be analyzed that will be used in the simulation.
   See ANNEX I for instructions to download wave files.

3. Open MATLAB. *OPEN start.m* file. This file allows the access to different folders. Click on *RUN* to have access to all the folders. The folders will appear in black on the Current Folder window (before they were gray).
4. Open *make_mesh.m* file. If *RUN* is activated, the program will draw the mesh for further calculations. In this situation, the latitude and longitude points can be changed in order to get different mesh sizes. The variable “inc” (increment) stands for the distance between nodes of the mesh we are creating, measured in nautical miles.
5. Open and run *make_waves_Copernicus.m* in order to open wave files previously saved. At line 10: ARX={'COP-20201001.nc',COP-20201002.nc'}, the user can change the files by introducing the name of the files the user is interested in analyzing.
   In the script *make_waves_Copernicus.m*, there is a variable named Tini_trip; this variable is used to specify the vessel’s time of departure related to the first day interpolated in the wave files. When Tini_trip=0, the vessel’s departure will be at 00:00; if Tini_trip=3, the vessel will leave at 03:00. Only o’clock hours are valid.
6. Open *find_ports* script and introduce the longitude and latitude of the ports of departure and arrival in order to obtain the nodes of the mesh where these ports are located.

7. Open and run *simroute_Opt.m* in order to look for the optimum route taking into account the mesh and the waves that have been introduced. Be aware that the name of the wave file which is being analyzed has to be loaded at line 10.
8. Open and run *simroute_Dmin.m.* In this case, the shortest path is calculated.
9. Open and run *make_emissions_Delta_2021.m*. In this script, there are several ship specific variables which have to be introduced by the user. These variables are the installed engine power, the engine load, the design speed of the vessel, the specific fuel consumption, the engine speed of rotation and the fuel characteristics. Once the variables have been introduced, the script is run and it outputs the fuel consumption and the quantity of pollutants emitted into the atmosphere, both for the minimum distance and for the optimized route.

It is highly recommended to save a folder with all the original files and make a copy of the folders in order to carry out tests on the copy and keep the originals intact.

6. CASE TEST

In order to make what was previously stated more understandable, below is a detailed example of the way to proceed to undertake a simulation.

The area of study is the Western Mediterranean and the type of vessel used is a Ro-Ro vessel sailing from Livorno to Alicante.

In order to run the simulation straightaway, the Copernicus files have already been downloaded and put into IN folder.

Proceed as follows:

1) Open Matlab and call *start.m* (enter) in Command Window or look for it in Current Folder Window and double click on it.

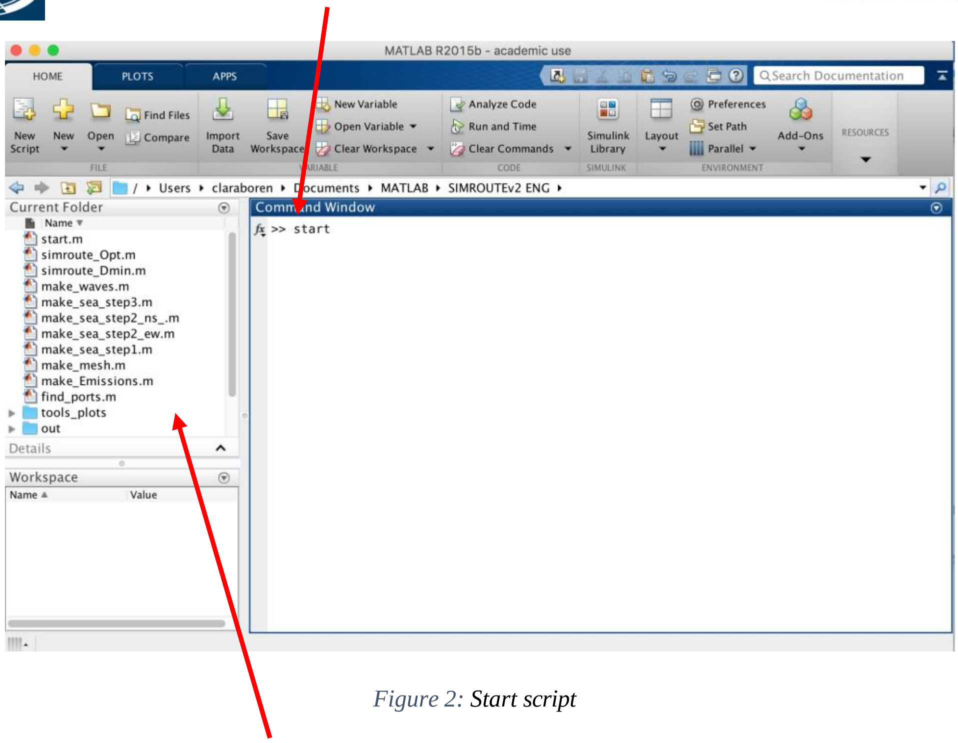


Figure 2: Start script

By doing so, the user will have access to the different folders (shown in grey, “non-active” before and in black, “active”, after).

2) Call *make_mesh* script to open it in Editor window by double clicking on it in Current Folder window in order to introduce the required inputs.

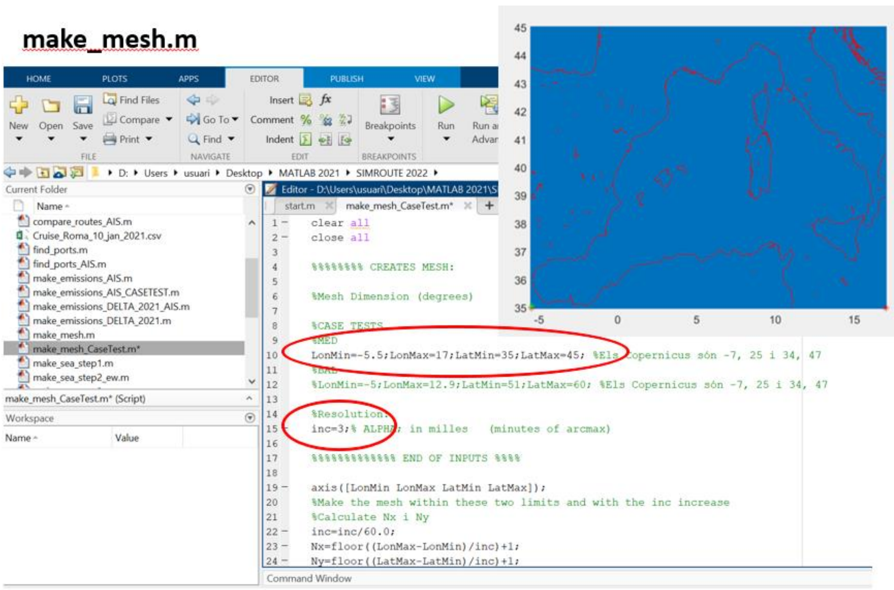


Figure 3: make_mesh script

On the Editor window, the user has to introduce the minimum and maximum longitudes and latitudes of the mesh he/she wants to create. Obviously, this mesh has to enclose latitudes and longitudes of port of departure and port of arrival.

In our example the selected longitudes and latitudes are the following (as seen in figure 3 above):

LonMin=-5.5 LonMax=17 LatMin=35 LatMax=45

The user has to select the mesh interval too (in miles). This means the distance between the computational nodes. As seen in figure 3, the mesh interval in our example is 3 miles (line 15 of the script, inc=3).

At this point,  RUN the script and the mesh will be made.

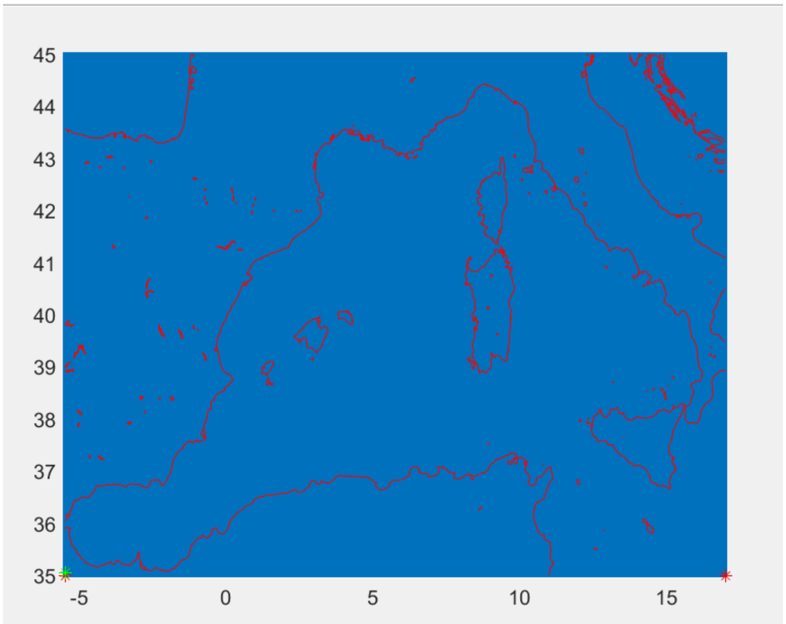


Figure 4: Mesh made by selecting maximum and minimum longitudes and latitudes

3) Call *make_waves_Copernicus.m* script by double clicking on it in the Current Folder window.

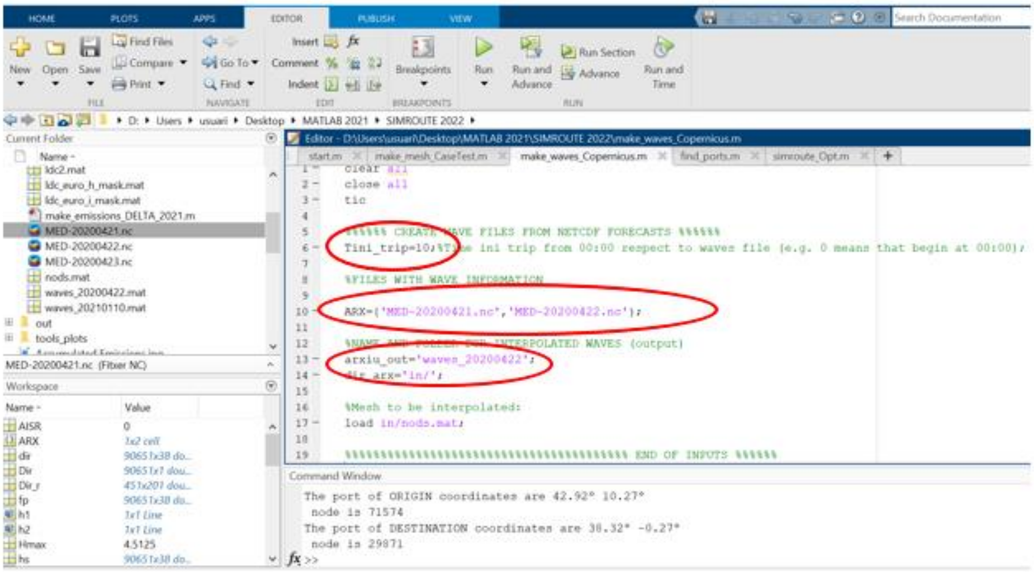


Figure 5: make_waves_Copernicus.m script

In order to interpolate the waves field into the computational mesh, previously the user has to have downloaded the wave files from Copernicus web page (as shown in Annex I of User's Manual).

In the example shown in these pages, the files analyzed where the ones for the 21st and 22ndth of April 2020.

The user has to introduce the starting time of the trip represented by Tini_trip variable in this program (line 6 of *make_waves_Copernicus.m* script, figure 5). This means selecting the start time of the wave field. The user has to bear in mind that giving the value 0 to this variable would mean midnight.

Afterwards, the user has to introduce the name of the wave fields to be interpolated, which would be the following ones in our example (line 10 *make_waves_Copernicus.m* script, figure 5):

ARX={'MED-20200421.nc', 'MED-20200422.nc'} (if there were more files, the user has to introduce them separated by a coma, i.e.: ARX={'MED-20201008.nc','MED-20201009.nc','MED-20201010.nc'})

The user has to give a name to the interpolated output file. In our example, the output file is called 'waves_20200422', as stated in line 13 of *make_waves_Copernicus.m* script of figure 5 (arxiu_out='waves_20200422').

Now the user has to RUN  the script and the interpolated wave field will be generated.

The generated file will be found in the folder IN as per below order:

dir_arx='in/' (line 14 figure 5).

4) Call find_ports script in Editor mode by double clicking on it in the Current Folder Window.

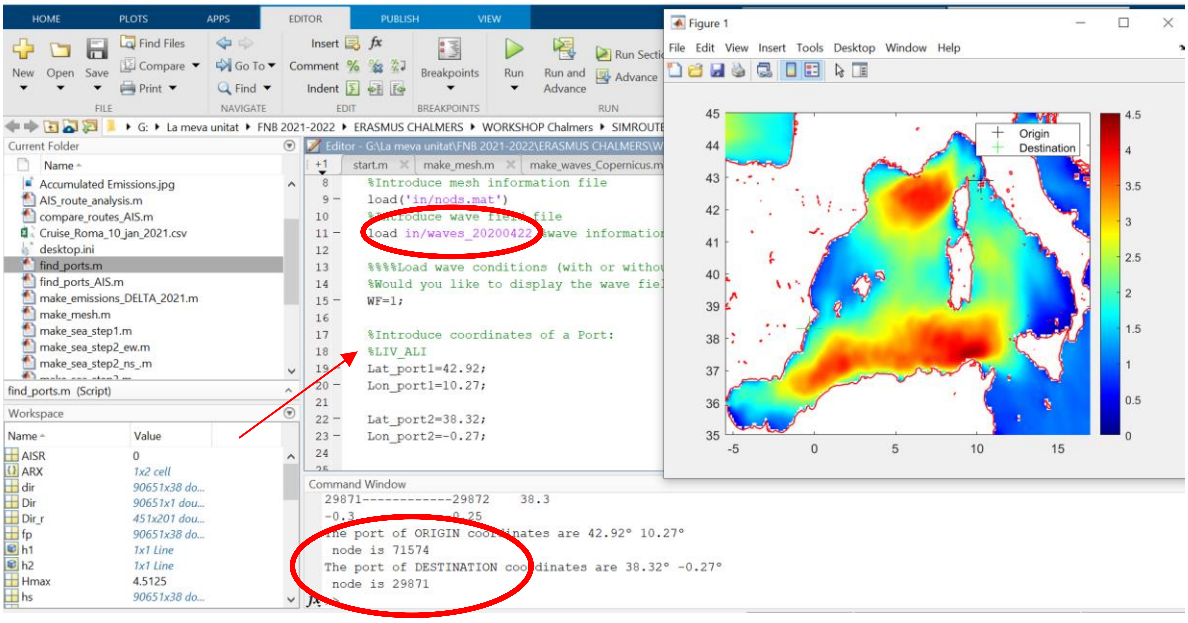


Figure 6: find_portd.m script

In this script, the name of the file with the wave interpolation data has to be introduced in line 11, as per above image.

The coordinates of the ports of departure (Port1) and arrival (Port2) have to be introduced too. Then run the script and a map will appear as an output showing the port of departure (node_ini) with a black cross and the port of arrival (node_end) with a green cross.

On the Command Window, there will appear the node corresponding to port of departure and the node for the port of arrival. In case the coordinates introduced corresponded to a node where there is no wave data, the following sentence would appear: “the node is land”. In this case, the user should select new coordinates the closest to the real ones by doing the following:

In order to get as close as possible to the port, the figure that find_ports.m displays is very useful because the user can zoom in the zone and choose to show the coordinates in the map.

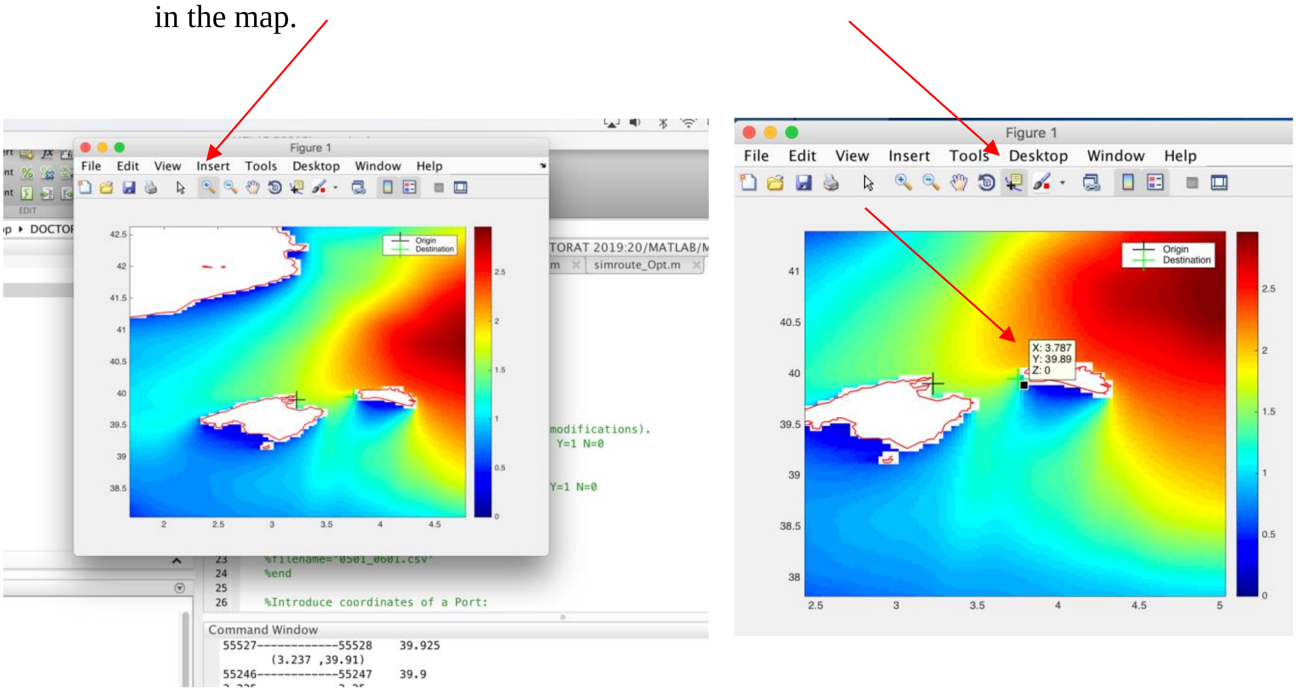


Figure 7: find_ports.m script searching for new coordinates (the figure does not correspond to the default example)

The point selected cannot be in a white zone of the map as this means that there is no sea data available and the program will read it as “land”. The point selected has to be next to the port but in the colored zone. The coordinates displayed are the ones to be introduced as port coordinates in *find_ports.m* and run the script again to get a new node and check it is not land.

5) Call *simroute_Opt.m* script in Editor mode by double clicking on it in the Current Folder window.

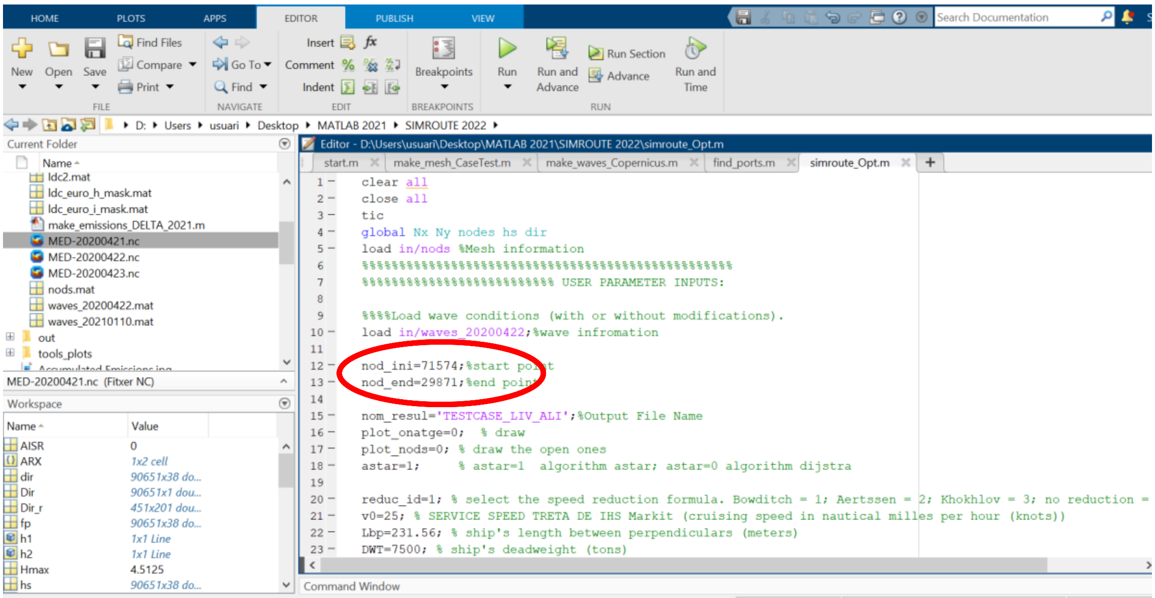


Figure 8: Simroute_Opt.m script

By running *simroute_Opt.m* script, the optimum route can be found taking into account the effect of waves. Nevertheless, before running it, the user has to ask *simroute_Opt.m* script to load the waves interpolated file which is located in IN folder.

Load in/waves_20200422 (see line 10 of figure 8).

In this script, the cruising speed of the vessel can be changed (variable v0, line 21 figure 8).

In the same script, the user has to indicate the start (nod_ini) and end (nod_end) nodes (see line 12 and 13 of figure 8) and he/she also has to give a name to the output file, as shown below:

nom_resul='TESTCASE_LIV_ALI'; (line 15 figure 8)

In order to indicate the start and end nodes, the user has to find out which of the nodes of the mesh he/she has created are situated in the closest position to the Port of Departure and Port of Arrival of the simulated route. These positions can be found out by using the *find_port.m* script, as previously stated. Run the script.

The user can check the value of any variable by double clicking on them in the Workspace window and a new window will be generated, the Variables window (see figure 9).

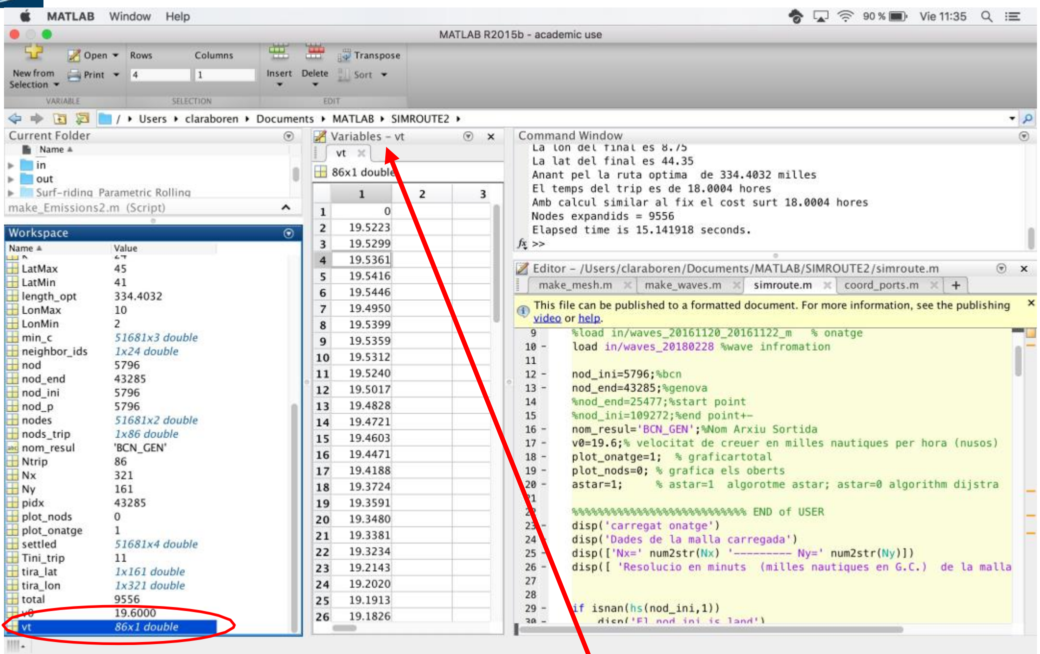


Figure 9: variables shown in Variables window

6) Call *simroute_Dmin.m* in Editor mode by double clicking on it in the Current Folder window.

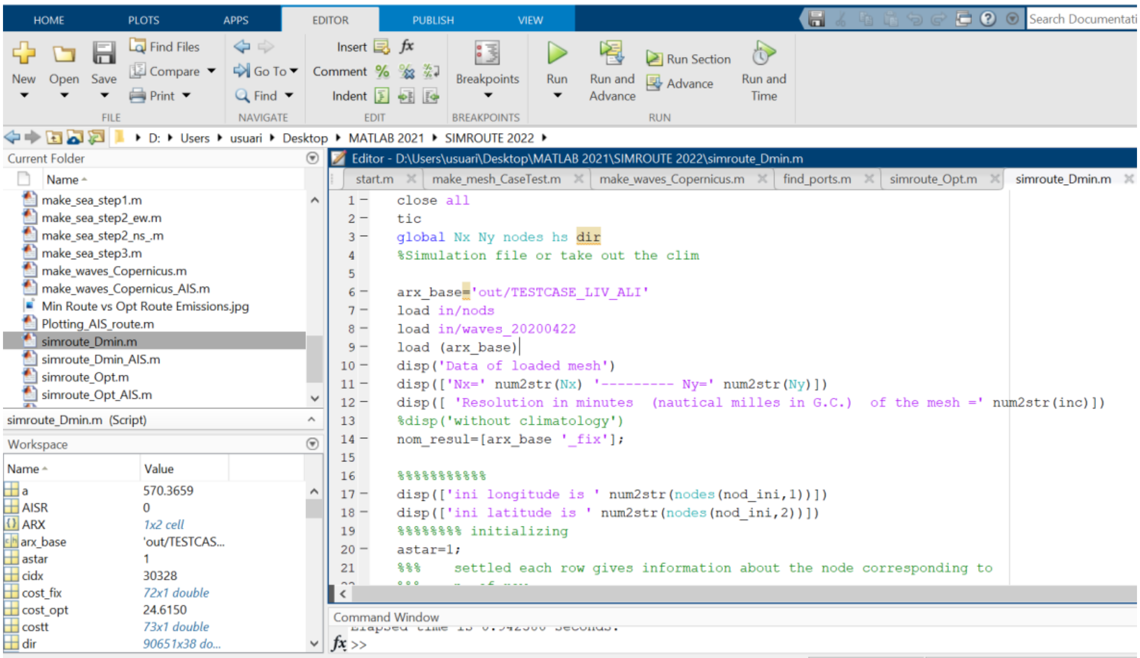


Figure 10: simroute_Dmin script

Simroute_Dmin.m is called in order to calculate the shortest path. In this script, the user has to give a name to the output file and locate it in the OUT folder as shown at line 6 of figure 10.

arx_base='out/TESTCASE_LIV_ALI'

The user has also to ask the script for loading the interpolated waves file (see line 8 of figure 10):

load in/waves_20200422

At this moment, the user can RUN the *simroute_Dmin.m* script.

7) Call *plot_routes_and_waves.m* on Editor mode by double clicking on it in Current Folder window.

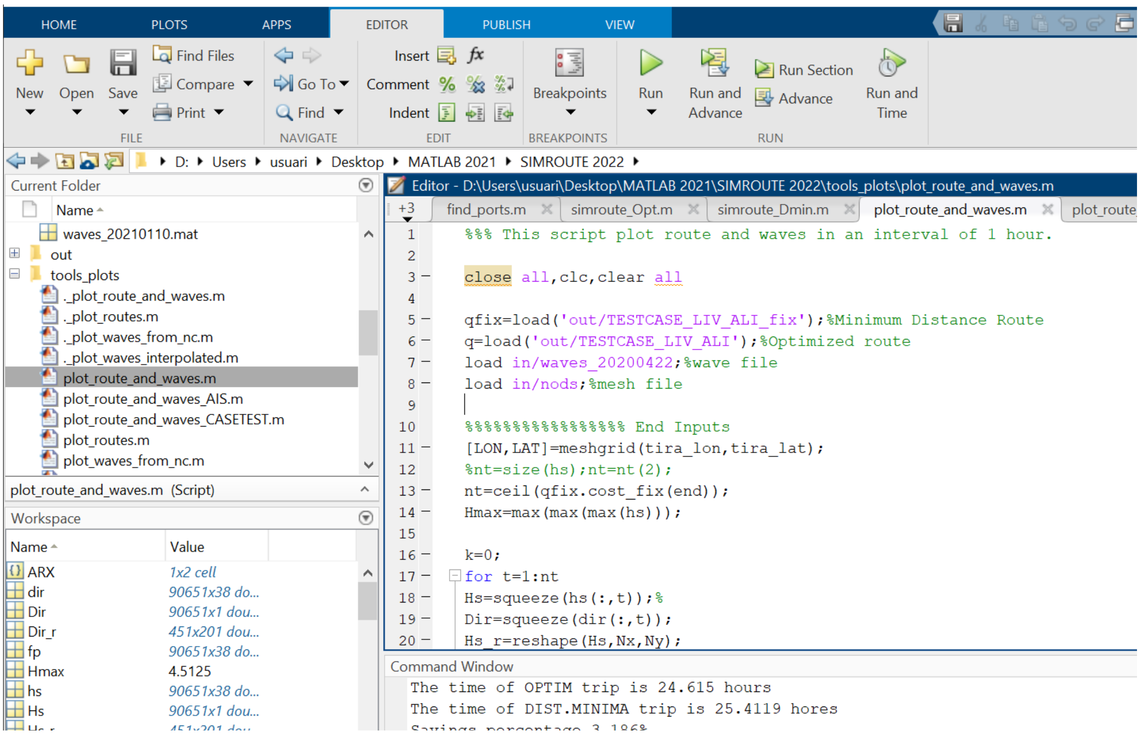


Figure 11: plot_routes_and_waves.m script

In order to see the evolution of the storm and the optimized and the minimum distance route in an hourly manner, the user has to call the function *plot_routes_and_waves*.

On this script, the user has to ask for loading the output files of *simroute_Opt.m* and *simroute_Dmin.m* and the wave field (see lines 5, 6 and 7 of figure 11, respectively):

q_fix=load('out/TEST CASE_LIV_ALI_fix');

q=load('out/TESTCASE_LIV_ALI');

load in/waves_20200422;

Below there is a plot shown of the Test Case as presented above (see figure 12).

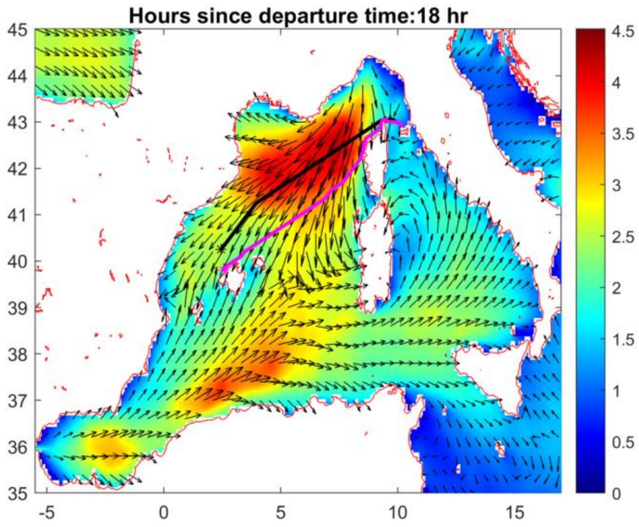


Figure 12: plot_routes_and_waves.m (minimum distance route in black, optimized route in magenta)

The different plots will be saved in folder PLOTS located into the folder OUT.

Before or after plotting, the user can call the module for calculating pollutants emission during the analyzed trip.

8) Call *make_emissions_Delta_2021.m* in Editor mode by double clicking on it on the Current Folder window.

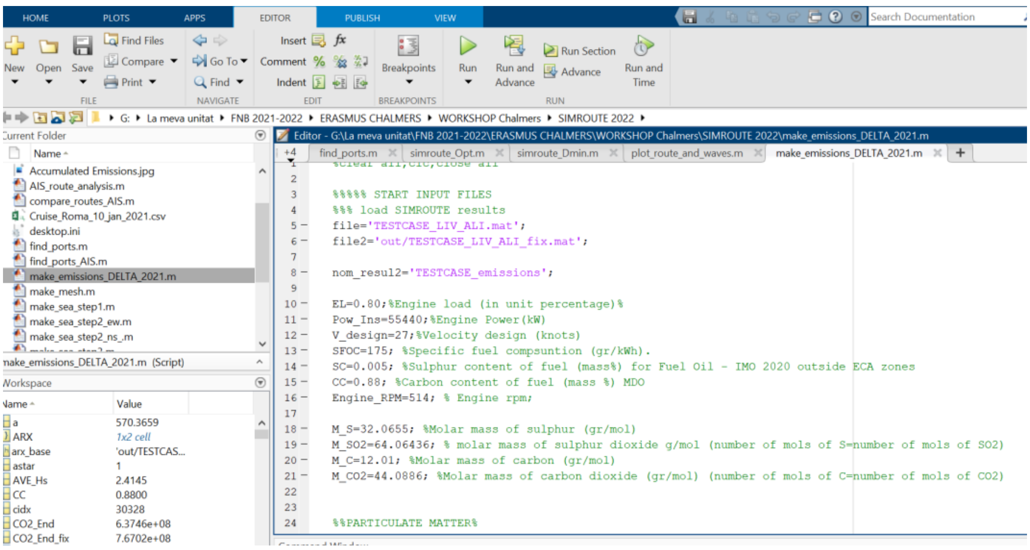


Figure 13: make_Emissions_Delta_2021.m

This script calculates the amount of carbon dioxide (CO_2), sulphur dioxide (SO_2), particulate matter (PM) and nitrogen oxides (NOx) emitted into the atmosphere per engine during the trip analyzed.

The required input variables, which can be changed by the user, are the following:

EL=0.80;	Engine load (in unit percentage)
Pow_Ins=14400:	Engine Power (kW)
V_design=19.6;	Velocity design (knots)
V_safety=0.5;	Safety margin of speed (knots)
SFOC=185;	Specific fuel consumption (gr/kWh).
SC=0.027;	Sulphur content of fuel (mass%) for Fuel Oil
CC=0.85;	Carbon content of fuel (mass %)
Engine_RPM=500;	Engine rpm

Additionally, there are some constants also required:

M_S=32.0655;	Molar mass of sulphur (gr/mol)
M_SO2=64.06436;	Molar mass of sulphur dioxide g/mol (number of mols of S=number of mols of SO2)
M_C=12.01;	Molar mass of carbon (gr/mol)
M_CO2=44.0886;	Molar mass of carbon dioxide (gr/mol) (number of mols of C=number of mols of CO2)
EF_EC=0.08	Emission factor elementary carbon (gr/kWh)
EF_OC=0.2	Emission factor for organic carbon (gr/kWh)
EF_ASH=0.06	Emission factor for ash (gr/kWh)
OC_EL=1.024	Part of organic carbon depending on EL (dimensionless)

The user has to ask the program to load the *simroute_Opt.m* and *simroute_Dmin.m* results and introduce the name of the results file (see line 5, 6 and 8 figure 13 respectively):

file='out/TESTCASE_LIV_ALI.mat';
file2='out/TESTCASE_LIV_ALI_fix.mat';

nom_resul2='TESTCASE_emissions';

Once the user has checked that all variables are well introduced, he/she has to RUN the *make_emissions_Delta_2021.m* script and the quantity of fuel consumed, quantity of pollutants emitted into the atmosphere for the optimized route and for the minimum distance route will appear on the Command Window (see figure 14).

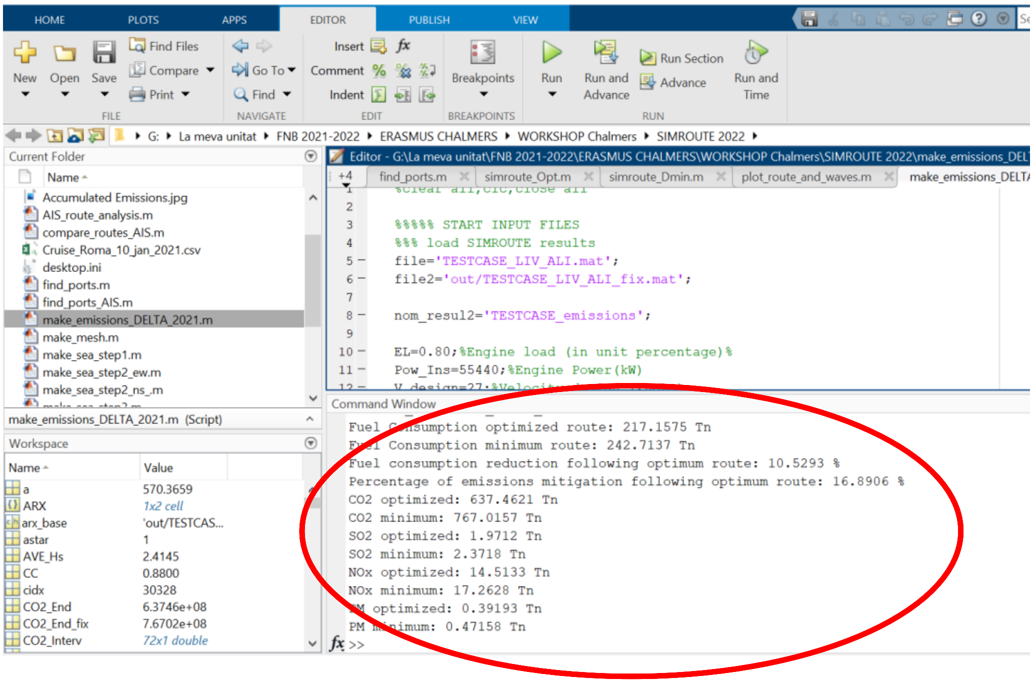


Figure 14: make_emissions_Delta_2021.m results

As already stated above, the user can also click on a variable in order to display the Variable window and see the hourly evolution of that variable, but only when the variable's value changes as a time function (sub index _fix is for the minimum distance route).

This script also shows below figures:

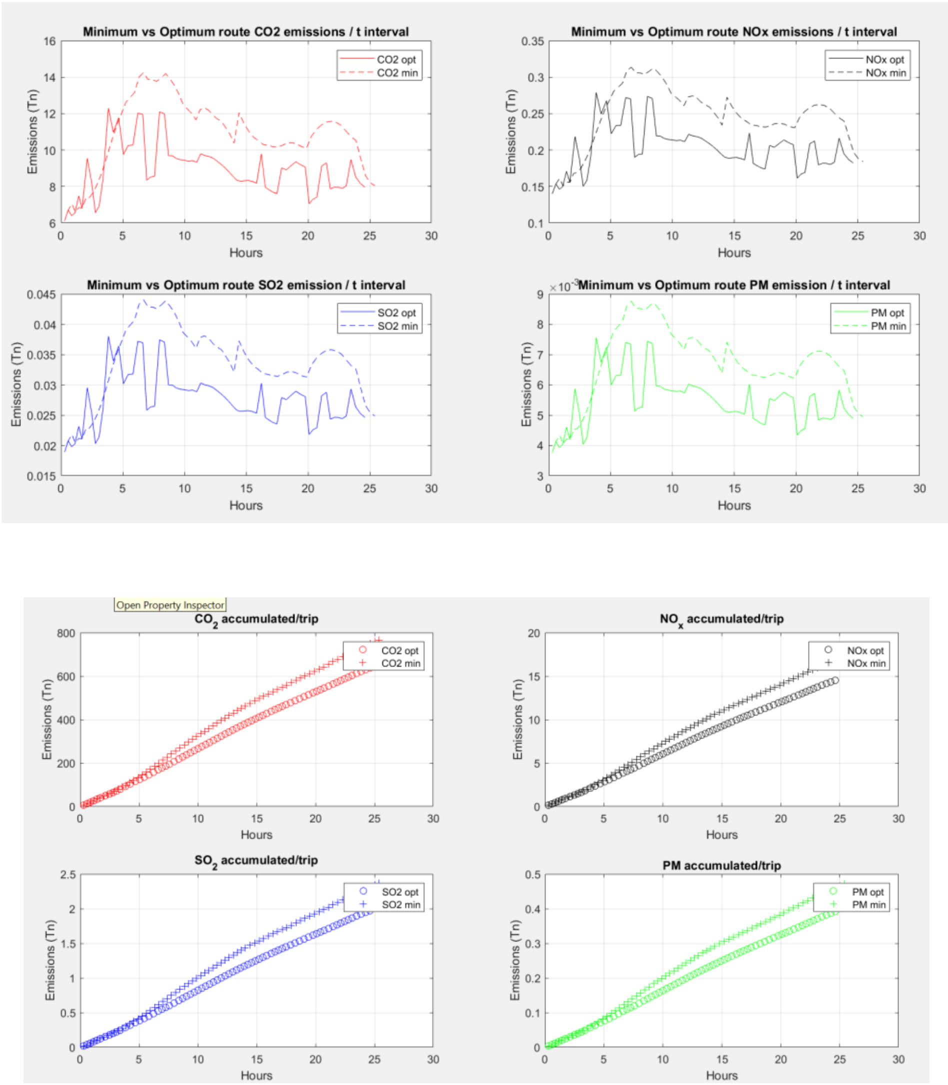


Figure 15: make_emissions_Delta_2021.m graphs

Which represent the amount of pollutants emitted into the atmosphere per interval of time for the optimum and minimum route and the emissions accumulated per trip, respectively.

The results of above Test Case have been obtained using A* algorithm due to the significant reduction of computational time compared to the Dijkstra model. This reduction in time occurs because less nodes are considered in comparison to the Dijkstra algorithm.

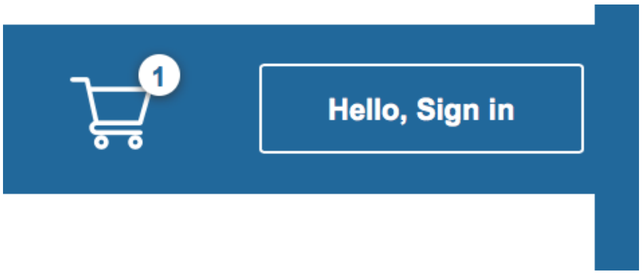
<u>ANNEX I</u>

HOW TO DOWNLOAD WAVE FILES FROM COPERNICUS

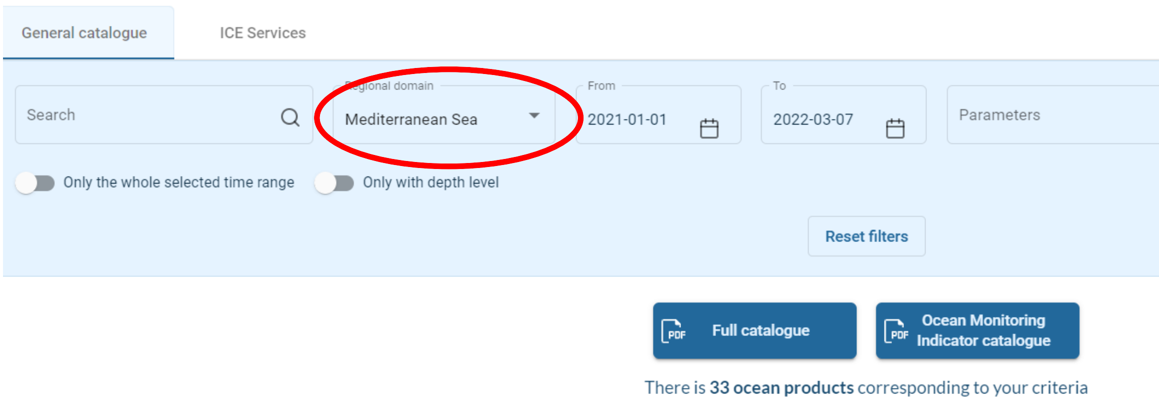
1) To download wave files, one has to go to Copernicus Marine Services web page https://marine.copernicus.eu

2) Go to Ocean Products

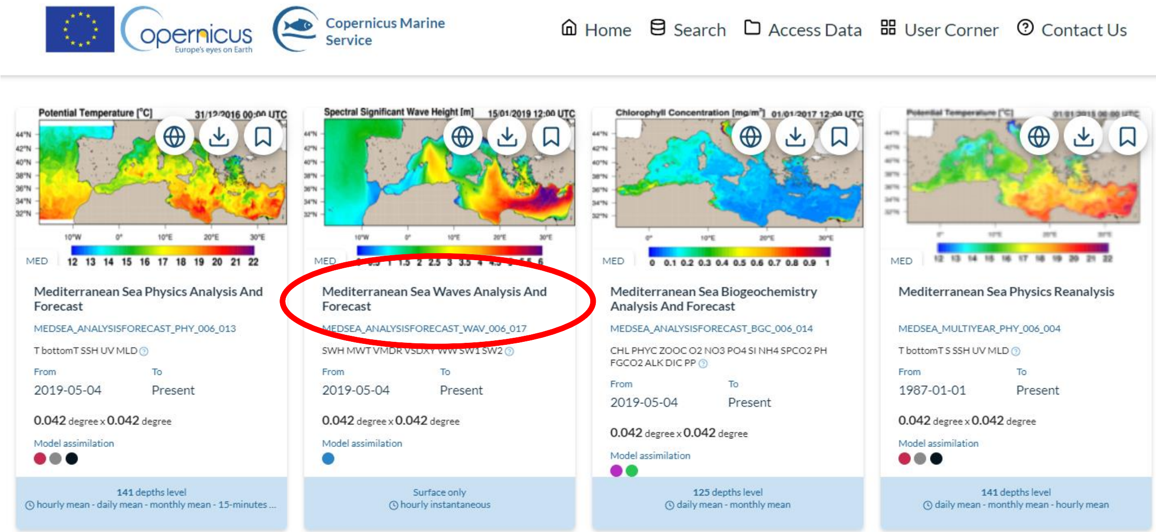
3) Register/sign in, in order to be allowed to download data



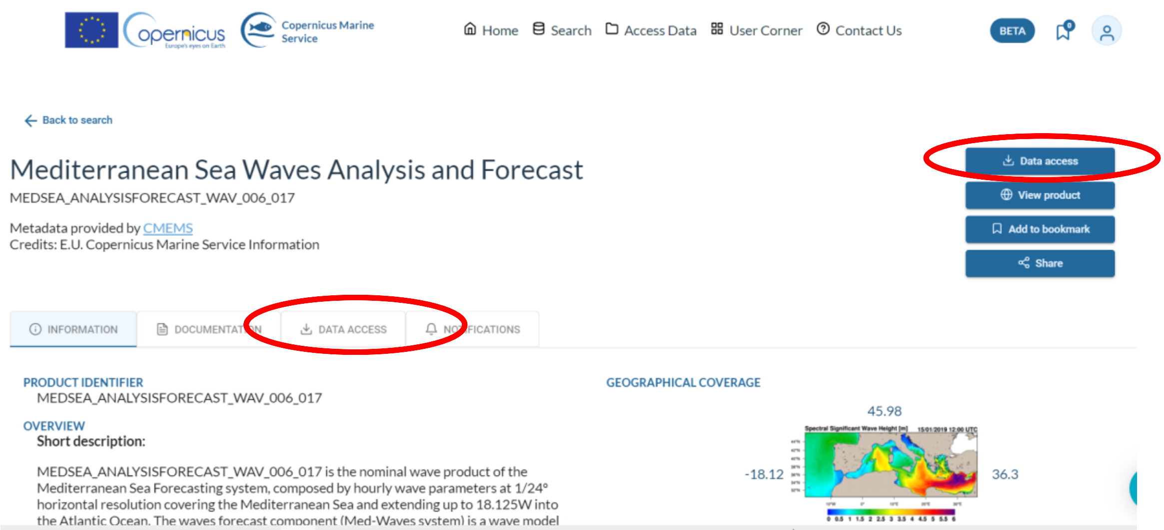
4) Select regional domain (for instance, Mediterranean Sea)

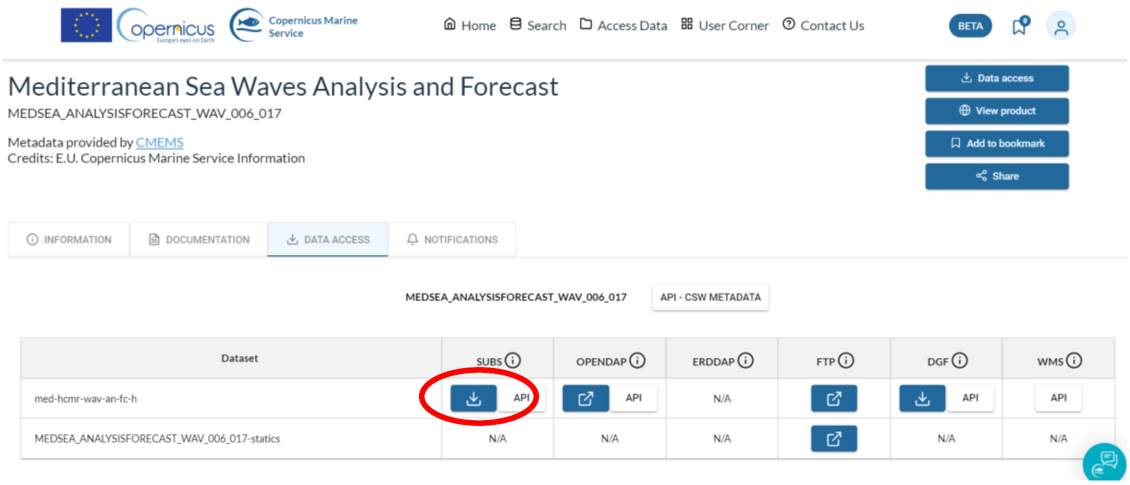


5) Select Mediterranean Sea waves analysis and forecasts product

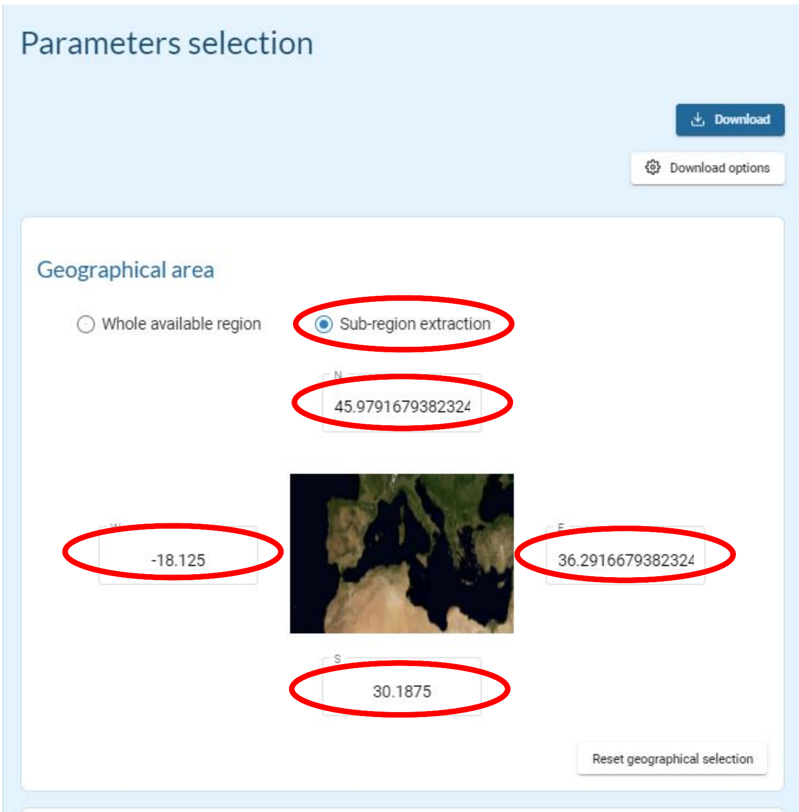


6) Select download products





7) Select wave file limits by choosing minimum and maximum longitudes and latitudes (zone to investigate must fit inside the limits)

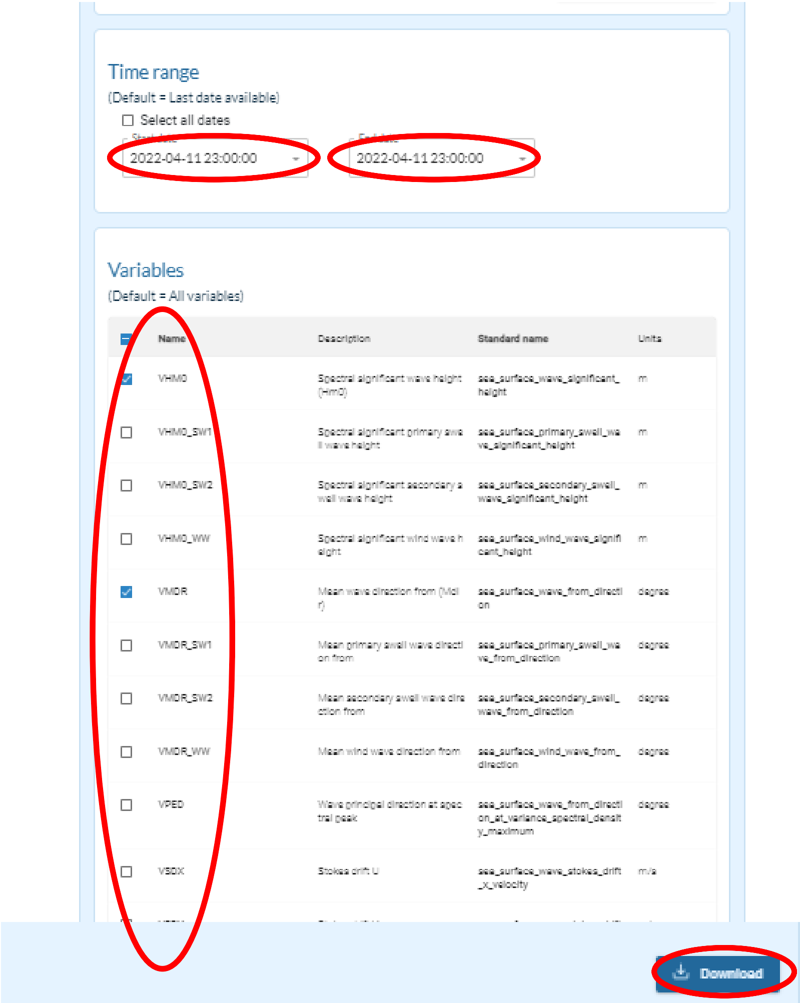


8) Select time range and variables:

VHMO: Sea surface wave significant height (m)

VMDR: Sea surface wave from direction (degrees)

VTM01_SW1: Sea surface primary swell wave mean period



7) Download data and save in IN folder.

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Nàutica de Barcelona

ANNEX II

COMPARING OPTIMUM AND MINIMUM DISTANCE ROUTE WITH AIS DATA

AIS_route_analysis.m

This newly added script presents briefly the characteristics of a downloaded *.csv* file containing a route from AIS sources, in this case, the source of files has been Marine Traffic database using the FNB's profile. The file shall have the data presented in the following way:

1) Header:
   Timestamp,Source,Speed,Latitude,Longitude. (The order is not important as the script calls each variable by its name).
2) Data: date(yyyy-MM-dd HH:mm:SS), speed(xx.xx), course(xxx), latitude(xx.xxxx), longitude(xx.xxxx).

Compare_routes

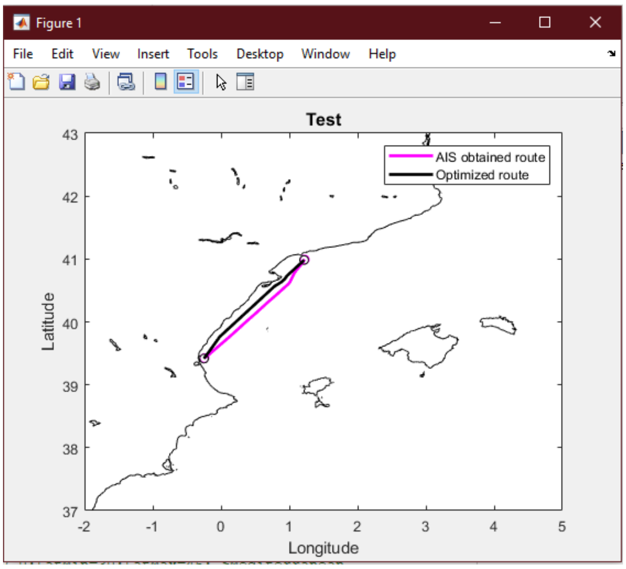
This script makes a comparison between an AIS obtained route and a SIMROUTE obtained route. Mainly, it summarizes the two routes by showing data such as:

- Total time of both routes' journey.
- Total distance sailed by both journeys.
- Average speed of the AIS route.
- Savings in terms of time of the SIMROUTE route.
- Savings in terms of distance of the SIMROUTE route.
- Port of origin.
- Port of destination.

The inputs for this script shall be:

a) A *.csv* file containing the AIS route.
b) A *.mat* file containing the SIMROUTE route.
c) Name of the desired output figure.

The outputs will be:



Total time with optimization	6.8353 hours
Distance with optimization	116.0161 miles
Total time in reality, obtained from AIS data	10.0667 hours
Distance obtained from AIS data	116.4468 miles
Average speed of	11.502 kn
Saved	3.2314 hours
Saved	32.0997 % (h)
Saved	0.43075 miles
Saved	0.36991 % (miles)
Port of origin	39.425º -0.244º
Port of destination	40.993º 1.2268º

Update of SIMROUTE User's guide

Using SIMROUTE's AIS route analysis functionality step-by-step:

1. Download from www.marinetraffic.com a .csv file containing the desired route to analyze. Remember it must have the following data: date and time, speed, course, latitude, and longitude.

2. Open MATLAB. Open start.m and run it.

3. Open make_mesh.m, choose mesh dimension in degrees; define LonMin, LonMax, LatMin and LatMax which stand for minimum and maximum longitude and minimum and maximum latitude. Next, choose mesh resolution in minutes of arcmax. Run it.

4. Open AIS_route_analysis.m, write the name of the .csv file on the variable 'filename'. Run it. This will present the main data for the AIS route which will define the SIMROUTE optimum route. Take notes of the time and dates of departure and arrival, theses will be the days for the wave field forecasts. Also, the time of departure will define the Tini_trip (Initial time) of make_waves.m. There is also the coordinates of the origin and destination and some other useful information such as the total time of the route, the total distance, and the average speed.

5. Download from https://marine.copernicus.eu/ the desired wave field. You will need to sign up if do not have an account, it is free for research. The wave fields can be found on Ocean products (DATA) under the name of 'Mediterranean sea waves analysis and forecast'. Once in there, choose 'MED00-HMCR-WAV-AN-FC-H' from the list, you will, then, have to set the desired geographical area and the dates for which you need the forecast. Make sure to download one-day forecasts as make_waves.m will only accept those. Therefore, you need to set start date for a given date at 00:00 and end date for the same date at 23:00. Download it.

6. Open make_waves.m, set Tini_trip to the time given on departure before. Type the name of the wave field files in ARX and name the output file on 'arxiu_out'. Run make_waves.m. Depending on the resolution and the amount of days introduced it may take some time. Once finished it will display 'Saving the swell' and the time consumed by the script.

7. Open find_ports.m. Now that the wave field is done you need to find out the nodes for start and end. The main use of this script is to find the nodes by loading the wave field and writing the start coordinates on Lon_port1 and Lat_port1 (longitude and latitude) and then the end coordinates on Lon_port2 and Lat_port2. The coordinates must be given in degrees. There is also the option to display the wave field and the AIS route to be able to see how the chosen coordinates compare to the real route that is going to be analyzed. You can do that by changing the variables 'WF' and 'AISR' values to '1' (make sure to load the AIS route in case you want to plot it). Run the script and take notes of the ORIGIN and DESTINATION nodes. If the nodes happen to be out of the wave field the script will display that the given node 'is land'.

8. Open simroute_Opt.m. Write the make_waves.m wave field fila name. Introduce the start and end nodes in the variables 'nod_ini' (start) and 'nod_end' (end). Choose the name for the output file and choose the cruising speed of the vessel in knots. Check that everything has been introduced correctly and run the script. This one may take some time as well for Western to Eastern Mediterranean routes it may take up to 1h30m. Once it has finished it will display some information of the created optimum route.

9. Open compare_routes.m. Type the desired name for the output figure. Write the AIS route file name in the variable 'filename' and the SIMROUTE optimum route in the variable 'q'. Choose the desired dimension of the figure by setting the maximum and minimum latitude and longitude in LonMin, LonMax, LatMin, and LatMax. Run the script. Now, you will obtain information about both routes and a saved figure in the 'out/compare_routes_figures folder of SIMROUTE.

Once the script is run, the output will be, as an example:
departure = datetime

2020-01-29 04:16:50

Total time in reality, obtained from AIS data	7.0931	hours
Distance obtained from AIS data	119.0299	miles
Average speed of	16.5682	kn

Port of origin	39.4189º -0.27819º
Port of destination	41.0164º 1.185º