DOI: 10.5281/zenodo.10205589

cmcc
Centro Euro-Mediterraneo
sui Cambiamenti Climatici

www.cmcc.it

VISIR-2 ship weather routing model: an introduction

Gianandrea Mannarini[1],

Mario Leonardo Salinas[1], *Lorenzo Carelli*[1], *Nicola Petacco*[2],
Josip Orović[3]

1) Fondazione Centro Euro-Mediterraneo sui Cambiamenti Climatici (CMCC)

Ocean Predictions and Applications Division

via Marco Biagi 5, 73100 Lecce (Italy)

2) DITEN, University of Genova (Italy)

3) Maritime Department, University of Zadar (Croatia)

seminar at
MMT-3ME TU-Delft on Nov.23, 2023

TUDelft

last update: 2023-11-25

Outline

Introduction

Numerical features

Vessel performance curves

Computational performance

Validation

Case studies

Operational service

Discussion

Motivation

New (IMO MEPC-80) decarbonisation strategy

- *2030: uptake of low-carbon fuels (5-10%)*
- *2050: a zero-carbon shipping*

EU-ETS for shipping

- *all calls at EU ports included*
- *starts 2024, progressive application*
- *CO2, CH4, N20*

role of weather routing

- *saving money*
- *saving emissions*
- *hardly quantified so far, open models needed*

https://shipnerdnews.com/imo-initiates-new-cii-revision-mepc-80/

IMO INTERNATIONAL MARITIME ORGANIZATION

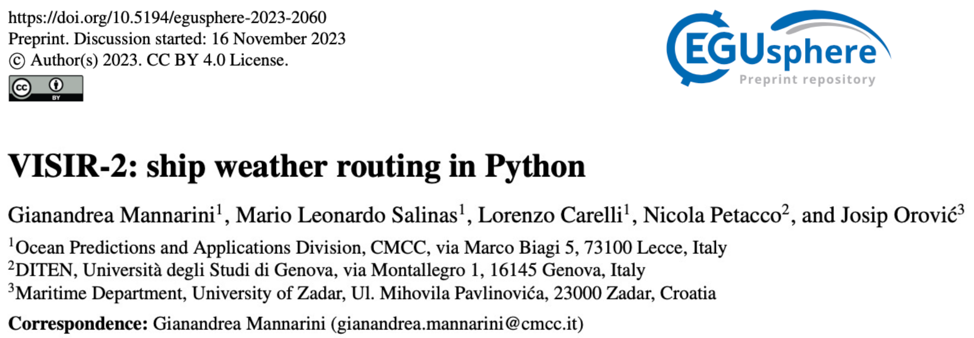
European Commission

[1] https://www.imo.org/en/MediaCentre/PressBriefings/pages/Revised-GHG-reduction-strategy-for-global-shipping-adopted-.aspx
[2] https://climate.ec.europa.eu/eu-action/transport/reducing-emissions-shipping-sector_en#:~:text=Inclusion%20of%20maritime%20emissions%20in,of%20the%20flag%20they%20fly.

VISIR-2 resources

open access – open review manuscript

https://egusphere.copernicus.org/preprints/2023/egusphere-2023-2060/



https://doi.org/10.5194/egusphere-2023-2060
Preprint. Discussion started: 16 November 2023
© Author(s) 2023. CC BY 4.0 License.

VISIR-2: ship weather routing in Python

Gianandrea Mannarini[1], Mario Leonardo Salinas[1], Lorenzo Carelli[1], Nicola Petacco[2], and Josip Orović[3]

[1]Ocean Predictions and Applications Division, CMCC, via Marco Biagi 5, 73100 Lecce, Italy
[2]DITEN, Università degli Studi di Genova, via Montallegro 1, 16145 Genova, Italy
[3]Maritime Department, University of Zadar, Ul. Mihovila Pavlinovića, 23000 Zadar, Croatia

Correspondence: Gianandrea Mannarini (gianandrea.mannarini@cmcc.it)

---

open language - open source model code

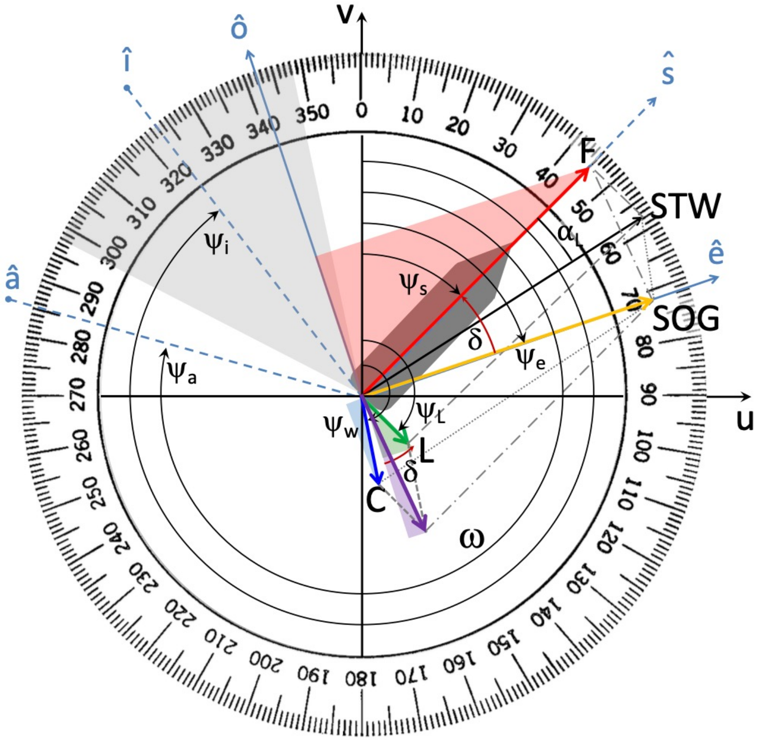
https://zenodo.org/records/8305527

Published September 7, 2023 | Version v1

[VISIR-2 ship weather routing model] source code

Salinas, Mario Leonardo[1]; Carelli, Lorenzo[1]; Mannarini, Gianandrea[1]

Kinematics: geometry



Key hypotheses:

- *linear superposition of velocities (STW, sea currents, leeway)*
- *ship's motion to occur along a graph edge*

→ *angle of attack δ between ship's heading and course*

$$\delta = \psi_s - \psi_e$$

→ *SOG as a vector sum of:*

STW and C (no leeway)

or:

F and ω (general)

$$S_g = F\cos(\delta) + \omega_{\parallel}$$

$$0 = -F\sin(\delta) + \omega_{\perp}$$

F: forward speed (along vessel's heading)

STW: speed through water (differs from F if leeway present)

SOG: speed over ground (along vessel's course – graph edge)

C: ocean current

L: leeway velocity

Kinematics: angle of attack

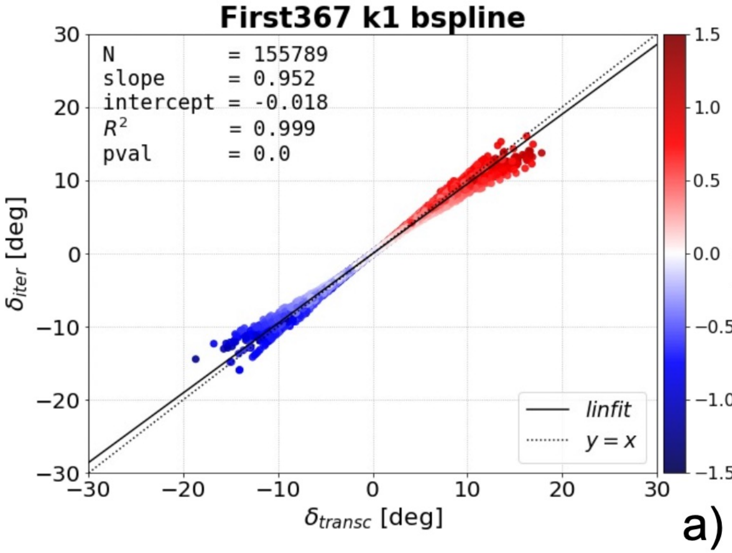
as F depends on environmental field's angle,

need to solve a transcendental equation for angle of attack δ:

$$\sin\delta = \frac{\omega_\perp(\delta, \delta_i(\delta))}{F(|\delta_i(\delta))|, |\delta_a(\delta)|)} \quad \Leftrightarrow \quad F \neq 0$$

---

solve numerically (scipy.optimize.root, not vectorizable) or via iteration (vectorizable):



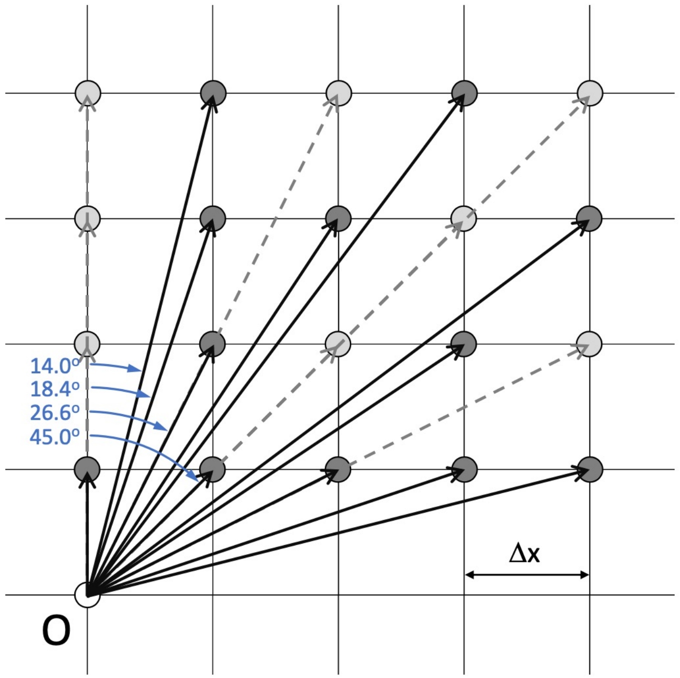
$$\begin{aligned} \delta^{(0)} &= 0 \\ \delta^{(k)} &= h(\delta^{(k-1)}) \quad \text{for} \quad k = 1, 2, \ldots \end{aligned}$$

$$h(x) = \arcsin\left(\frac{\omega_\perp(\delta = x, \delta_i = x - \gamma)}{F(|x - \gamma|)}\right)$$

Graph structure

Graph stencil for connectivity

up to 4th order neighbours ($\nu=4$)



benefits of pruning of collinear edges (dashed) :

- *saving RAM memory*
- *more faithful representation of the environmental fields*

connectivity and number N_{q1} of edges in the first quadrant

ν	$\Delta\theta$ [°]	N_{q1}	$\nu(\nu+1)$
1	45.0	2	2
2	26.6	4	6
3	18.4	8	12
4	14.0	12	20
5	11.3	20	30
6	9.5	24	42
7	8.1	36	56
8	7.1	44	72
9	6.3	56	90
10	5.7	64	110

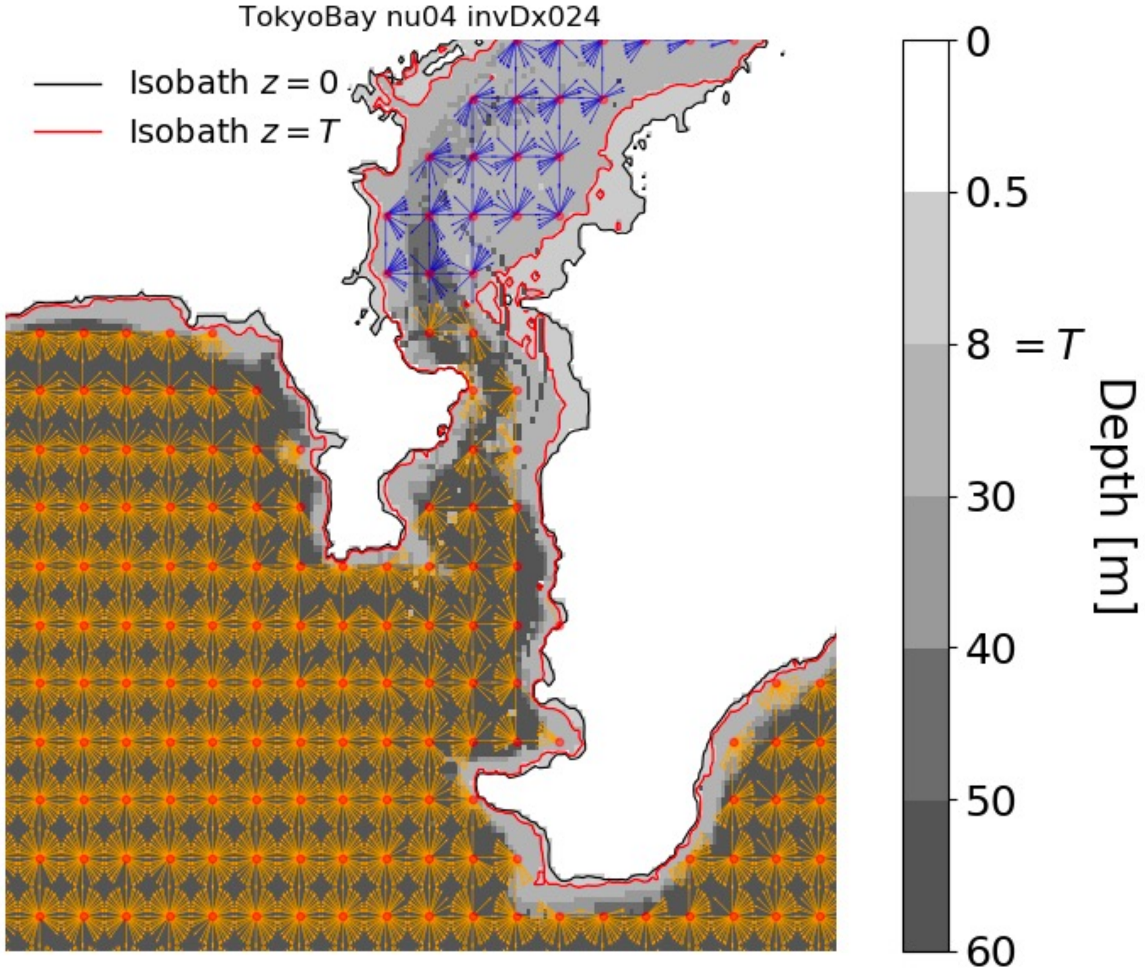
Graph computation

indexing via a K-dimensional Tree

a spatial data structure which can effectively be queried for:

- *nearest neighbours (coast proximity of nodes)*
- *range queries (coast intersection of edges)*

implementation in Python: scipy.spatial.KDTree



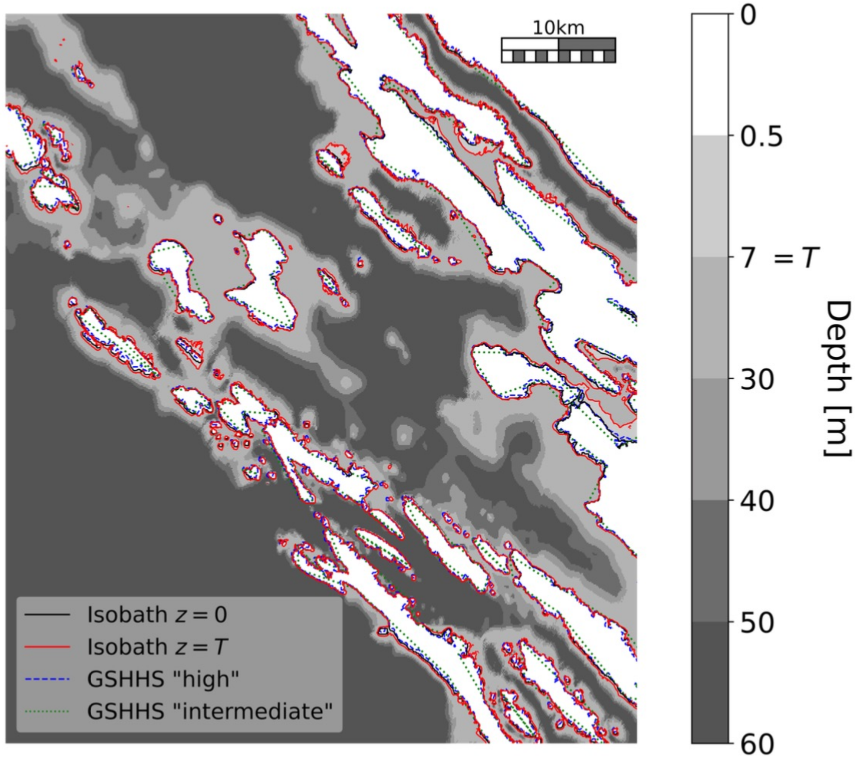
Pseudo-shoreline

pseudo-shoreline: avoiding too shallow water for a given vessel

high resolution bathymetry dataset ➔ compute as a zero contour line of under-keel clearance:

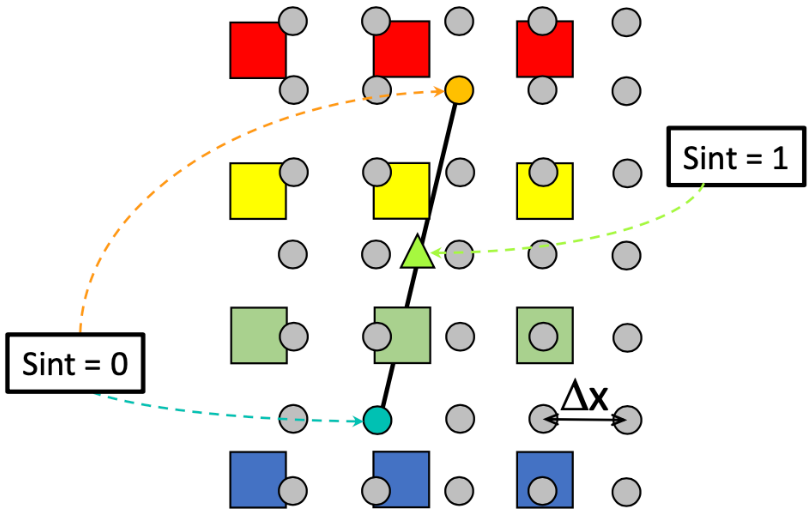
$$UKC = z - T$$

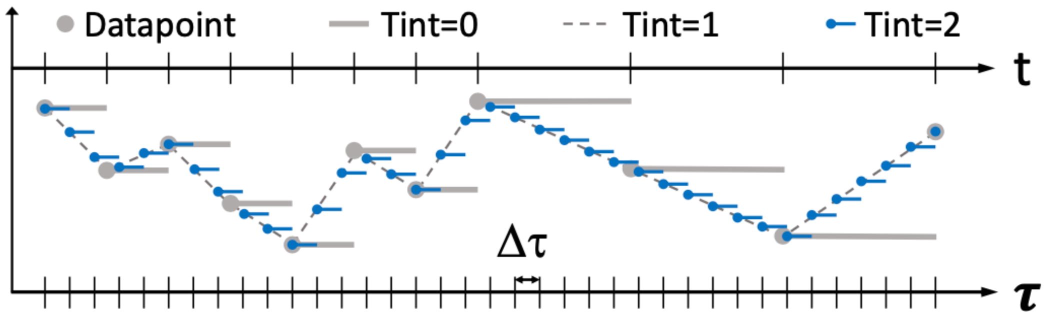
retain in the graph just edges with UKC > 0



- *GSHHG "high" res shoreline: 200m*
- *GEBCO_2022 bathy : 463 m*
- *EMODnet bathy: 116 m*

Space and Time interpolation





Remap environmental fields to the graph grid:

Two options:

- *averaging between the edge head and tail's values ("Sint = 0")*
- *interpolating their values to the 315 edge barycentre ("Sint = 1", default)*

a) *environmental field values (grey dots) interpolated in time on a finer grid with Δτ spacing ("Tint = 2" or blue dots)*

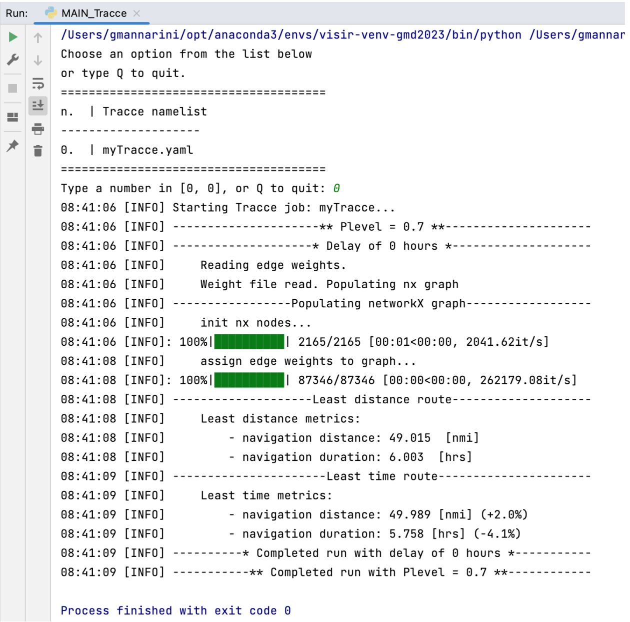
b) *edge weight at the nearest available timestep (floor function used, blue segments) is selected*

VISIR-2 suite modules

Greater modularity with respect to VISIR-1

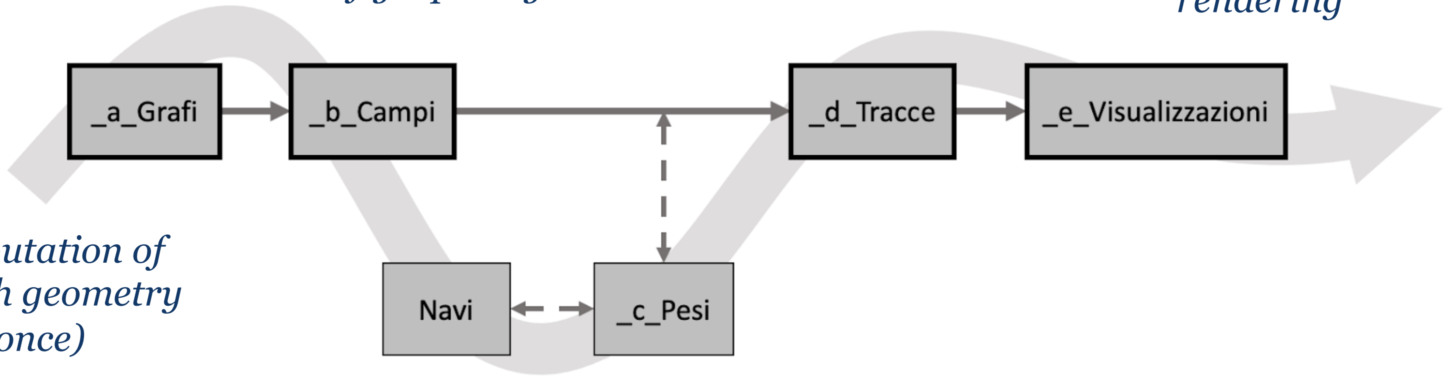
facilitating both R&D and operational applications

conda virtual environment ("visir-venv") for portability



```
/Users/gmannarini/opt/anaconda3/envs/visir-venv-gmd2023/bin/python /Users/gmannar
Choose an option from the list below
or type Q to quit.
======================================
n.  | Tracce namelist
--------------------
0.  | myTracce.yaml
======================================
Type a number in [0, 0], or Q to quit: 0
08:41:06 [INFO] Starting Tracce job: myTracce...
08:41:06 [INFO] ---------------------** Plevel = 0.7 **--------------------
08:41:06 [INFO] --------------------* Delay of 0 hours *--------------------
08:41:06 [INFO]     Reading edge weights.
08:41:06 [INFO]     Weight file read. Populating nx graph
08:41:06 [INFO] -----------------Populating networkX graph------------------
08:41:06 [INFO]     init nx nodes...
08:41:06 [INFO]: 100%|██████████| 2165/2165 [00:01<00:00, 2041.62it/s]
08:41:08 [INFO]     assign edge weights to graph...
08:41:08 [INFO]: 100%|██████████| 87346/87346 [00:00<00:00, 262179.08it/s]
08:41:08 [INFO] --------------------Least distance route--------------------
08:41:08 [INFO]     Least distance metrics:
08:41:08 [INFO]         - navigation distance: 49.015  [nmi]
08:41:08 [INFO]         - navigation duration: 6.003  [hrs]
08:41:09 [INFO] ----------------------Least time route----------------------
08:41:09 [INFO]     Least time metrics:
08:41:09 [INFO]         - navigation distance: 49.989 [nmi] (+2.0%)
08:41:09 [INFO]         - navigation duration: 5.758 [hrs] (-4.1%)
08:41:09 [INFO] ----------* Completed run with delay of 0 hours *-----------
08:41:09 [INFO] -----------** Completed run with Plevel = 0.7 **------------

Process finished with exit code 0
```



space and time interpolation of environmental fields to the grid of graph edges

computation of optimal routes and metrics

rendering

computation of graph geometry (just once)

identification of ship performance curves from LUT (just once)

computation of graph edge weights (sailing time and CO2 emission rates – called by Tracce)

https://zenodo.org/records/8305527

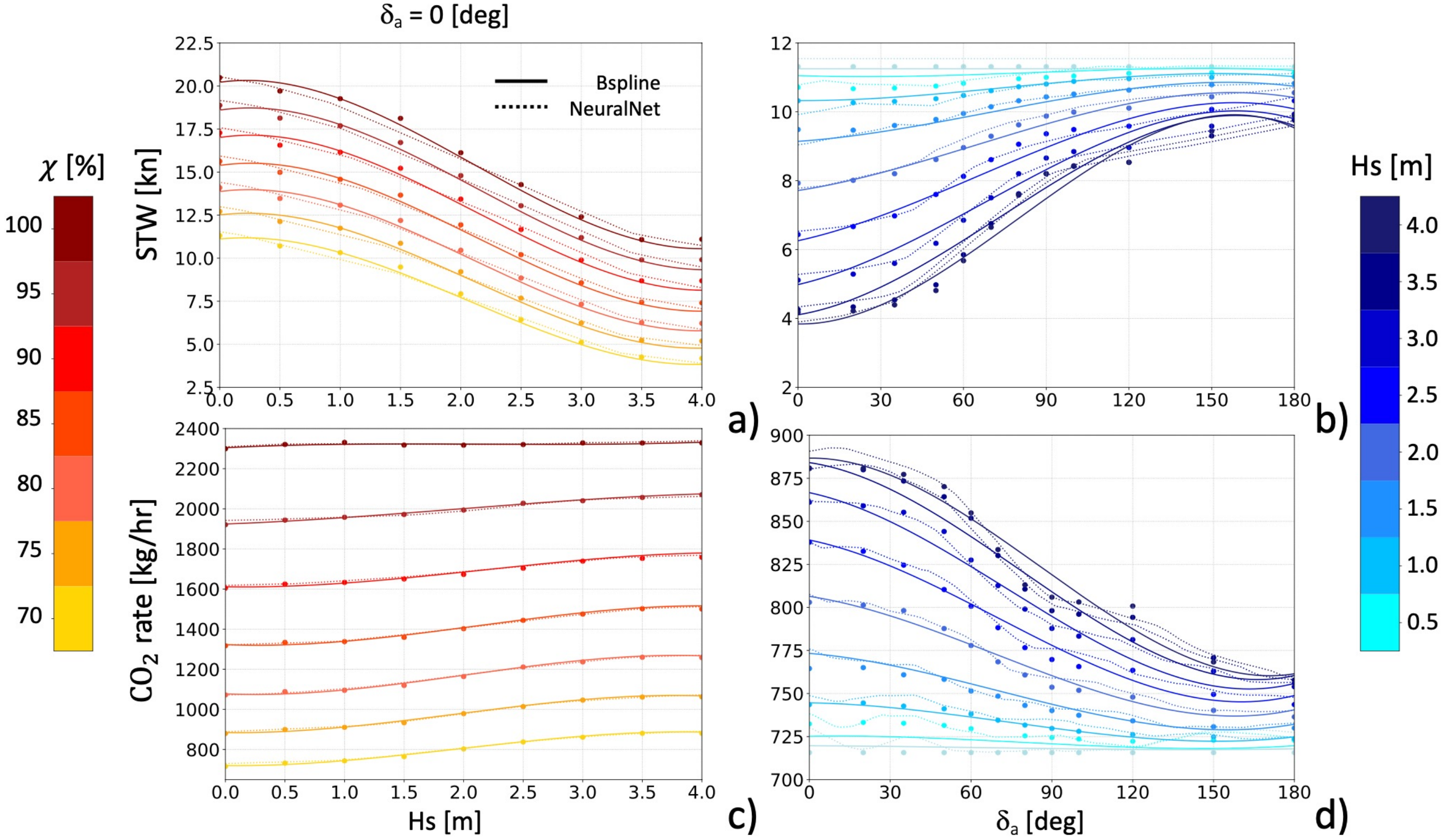
Vessel performance curves: ferry

use of University of Zadar ship command-bridge/ engine-room coupled simulator:

- *wind waves*
- *no leeway*
- *explored dependence of STW on*
  - *engine load*
  - *significant wave height*
  - *relative wave direction*
- *outcome interpolated through a neural network (multi-layer perceptron via the scikit-learn package)*

Interreg
Italy - Croatia
GUTTA
EUROPEAN UNION

Name	Symbol	Value	Units
Length overall	LOA	125	m
Draft middle	T	5.3	m
Deadweight	DWT	4,050	t
Main engine power	P_{main}	4,000	kW
Main engine rated speed	n_{eng}	750	rpm
Service speed	v_S	19	kn



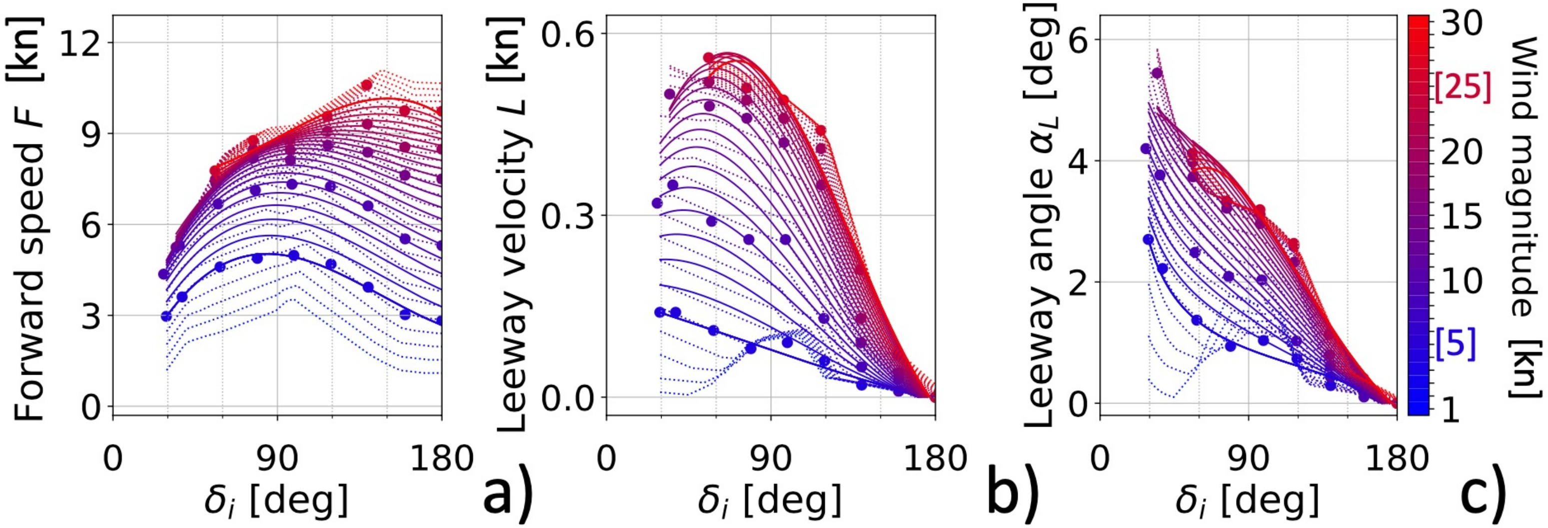
Mannarini et al 2021, https://doi.org/10.3390/jmse9020115

Vessel performance curves: sailboat

use of WinDesign Velocity Prediction Program:

- *both hydrodynamic and aerodynamic effects*
- *wave added resistance via "Delft method" on DSYHS series*
- *same wind-wave relationship of the ferry used*
- *for upwind, main sail and jib assumed; otherwise: main sail and spinnaker*

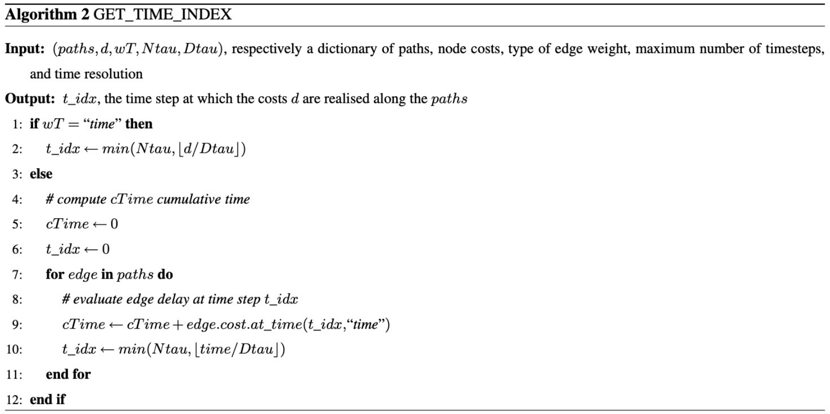
Name	Symbol	Value	Units
Length of hull	L_{hull}	10.68	m
Draft	T	2.2	m
Displacement	∇	5,773	m^3
Rudder wetted surface	-	1.42	m^2
Keel wetted surface	-	3.31	m^2
Main sail area	-	38	m^2
Jib sail area	-	3.97	m^2
Spinnaker area	-	95	m^2



Claughton et al 1999, 2003

Shortest path problem: least-CO$_2$ algorithm

- *Dijkstra's algorithm generalized for dynamic edge weights*
- *same complexity of static algorithm under FIFO hypothesis*
- *built on single_source_Dijkstra function of the networkX library*
- *use of data structures (heaps) to achieve ideal performance*
- *key advancement for least-CO2 paths is retrieving an edge weight at a specific time step*



Algorithm 2 GET_TIME_INDEX

Input: $(paths, d, wT, Ntau, Dtau)$, respectively a dictionary of paths, node costs, type of edge weight, maximum number of timesteps, and time resolution

Output: t_idx, the time step at which the costs d are realised along the $paths$

```
if $wT =$ "time" then
    $t\_idx \leftarrow min(Ntau, \lfloor d/Dtau \rfloor)$
else
    # compute $cTime$ cumulative time
    $cTime \leftarrow 0$
    $t\_idx \leftarrow 0$
    for $edge$ in $paths$ do
        # evaluate edge delay at time step $t\_idx$
        $cTime \leftarrow cTime + edge.cost.at\_time(t\_idx,$"time"$)$
        $t\_idx \leftarrow min(Ntau, \lfloor time/Dtau \rfloor)$
    end for
end if
```

Algorithm 1 _DIJKSTRA_TDEP

Input: $(G, source, target, wT, Ntau, Dtau)$, respectively a `networkX` graph, $source$ and $target$ nodes, type of edge weight, maximum number of timesteps, and time resolution

Output: $(costs, paths)$, Two dictionaries keyed by node id: path costs from the source (e.g. cumulated CO$_2$), and corresponding optimal paths

```
$costs \leftarrow \{\}$
$seen \leftarrow \{source : 0\}$
$paths \leftarrow \{source : [source]\}$
# fringe is a min-priority queue of $(cost, node)$ tuples
$fringe \leftarrow heap()$
$fringe.push(0, source)$
while $fringe \neq \emptyset$ do
    $(d, v) \leftarrow fringe.pop()$
    if $v \in costs$ then
        # Already visited node
        skip
    end if
    $costs[v] \leftarrow d$
    if $v = target$ and $\forall n \in G.neigh(target), n \in seen$ then
        exit
    end if
    $t\_idx \leftarrow get\_time\_index(paths[v], d, wT, Ntau, Dtau)$
    # Iterate on $v$'s forward-star
    for $(u, cost)$ in $G.succ(v)$ do
        # evaluate edge weight of $wT$ type at time step $t\_idx$
        $c \leftarrow cost.at\_time(t\_idx, wT)$
        $vu\_cost \leftarrow costs[v] + c$
        if $u \notin seen$ or $vu\_cost < seen[u]$ then
            $seen[u] \leftarrow vu\_cost$
            $fringe.push(vu\_cost, u)$
            $paths[u] \leftarrow paths[v] + [u]$
        end if
    end for
end while
```

Numerical performance: optimal paths

Three variants of the algorithm:

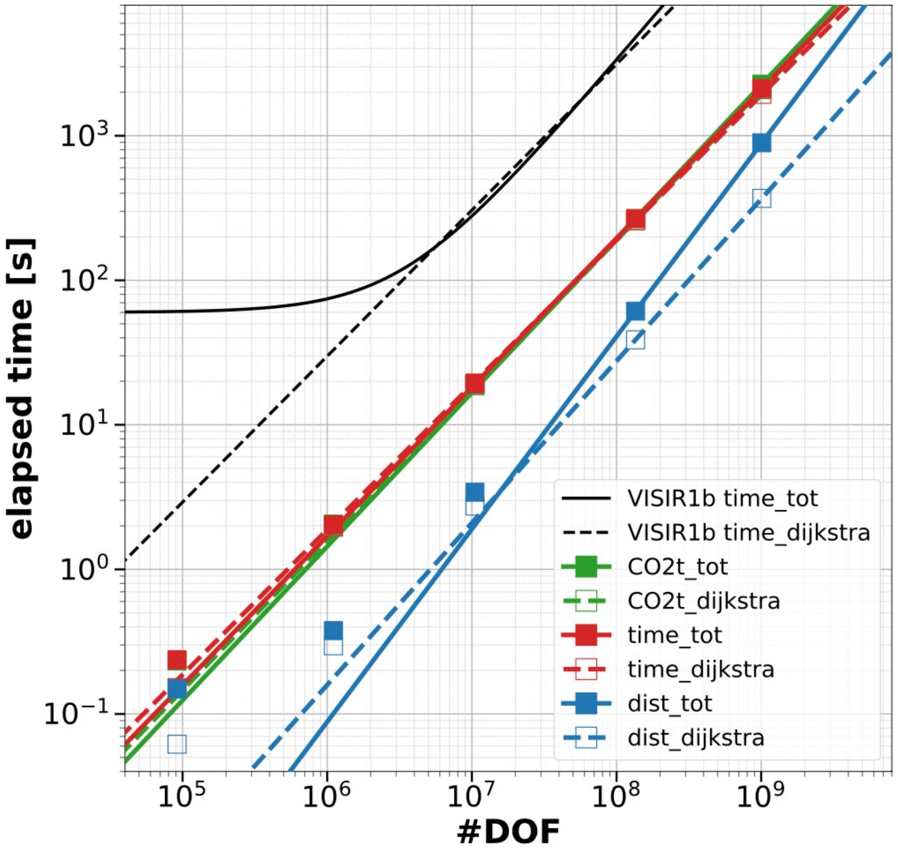
- *least-distance*
- *least-time*
- *least-CO2*

Assessment for:

- *"Dijkstra": optimal sequence of graph nodes*
- *"total" : "Dijkstra" + marine and vessel dynamical information along the path*

Outcome:

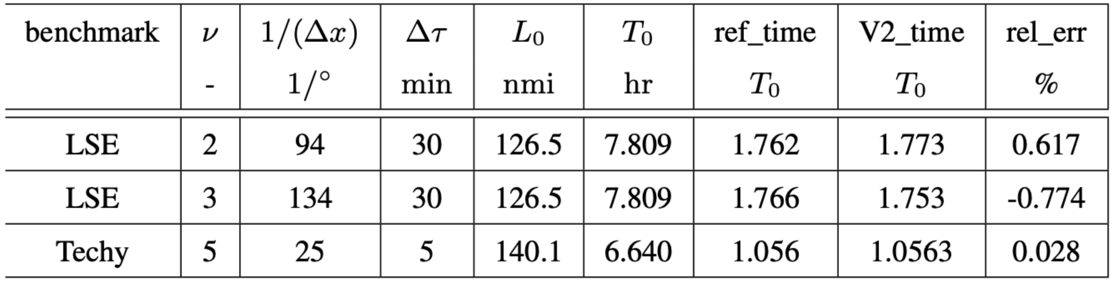
- *linearity in the number of DOF*
- *10x faster than VISIR-1*
- *least-distance routine still to be improved*
- *RAM: 420B per DOF (5x more than VISIR-1, to be improved e.g via single precision)*



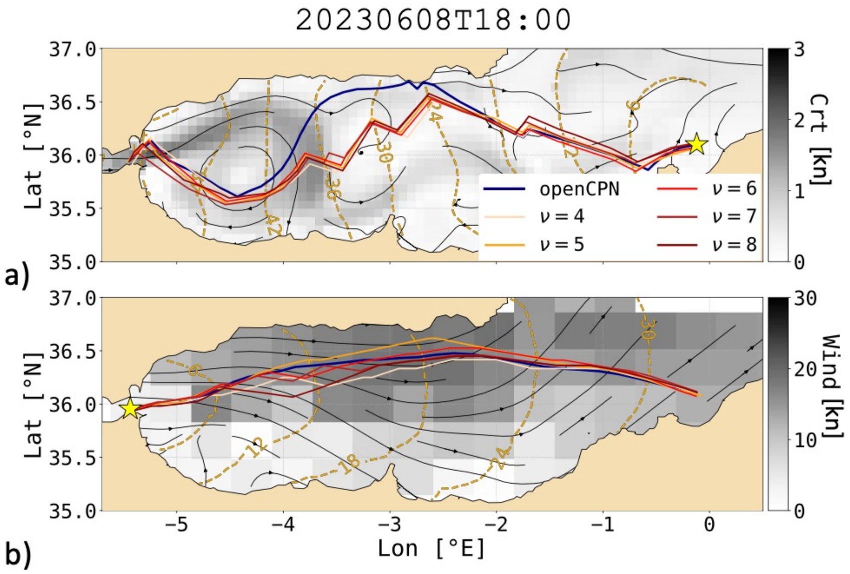
Validation

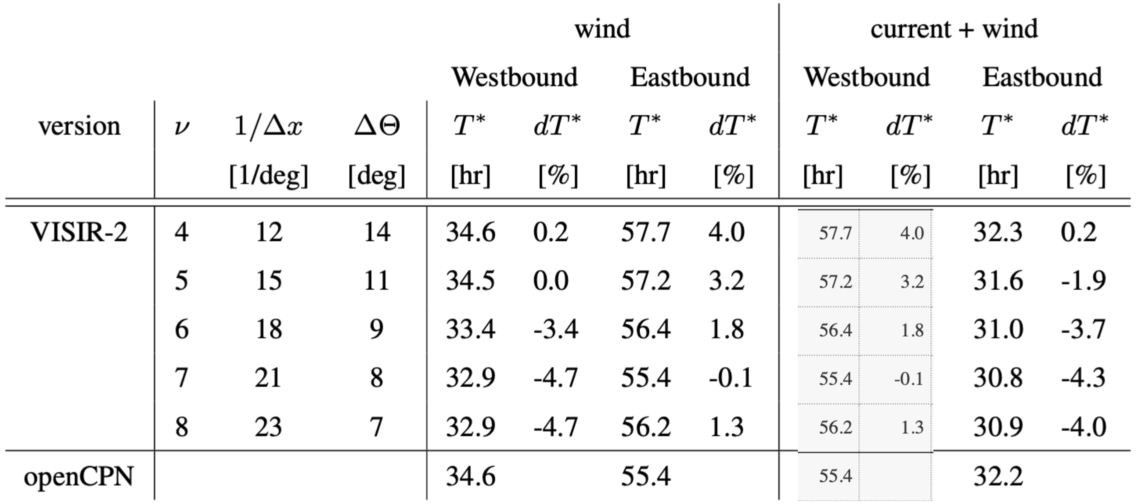
VISIR-2 routes and metrics were compared to

- *MIT model based on partial differential equations (LSE, *)*
- *semi-analytical results (cycloid, Techy)*
- *openCPN (dynamic programming)*



benchmark	ν	$1/(\Delta x)$	$\Delta\tau$	L_0	T_0	ref_time	V2_time	rel_err
	-	$1/°$	min	nmi	hr	T_0	T_0	%
LSE	2	94	30	126.5	7.809	1.762	1.773	0.617
LSE	3	134	30	126.5	7.809	1.766	1.753	-0.774
Techy	5	25	5	140.1	6.640	1.056	1.0563	0.028





version	ν	$1/\Delta x$	$\Delta\Theta$	wind Westbound T^*	wind Westbound dT^*	wind Eastbound T^*	wind Eastbound dT^*	current + wind Westbound T^*	current + wind Westbound dT^*	current + wind Eastbound T^*	current + wind Eastbound dT^*
		[1/deg]	[deg]	[hr]	[%]	[hr]	[%]	[hr]	[%]	[hr]	[%]
VISIR-2	4	12	14	34.6	0.2	57.7	4.0	57.7	4.0	32.3	0.2
	5	15	11	34.5	0.0	57.2	3.2	57.2	3.2	31.6	-1.9
	6	18	9	33.4	-3.4	56.4	1.8	56.4	1.8	31.0	-3.7
	7	21	8	32.9	-4.7	55.4	-0.1	55.4	-0.1	30.8	-4.3
	8	23	7	32.9	-4.7	56.2	1.3	56.2	1.3	30.9	-4.0
openCPN				34.6		55.4		55.4		32.2	

(bug in VISIR-2 manuscript's Tab.6)

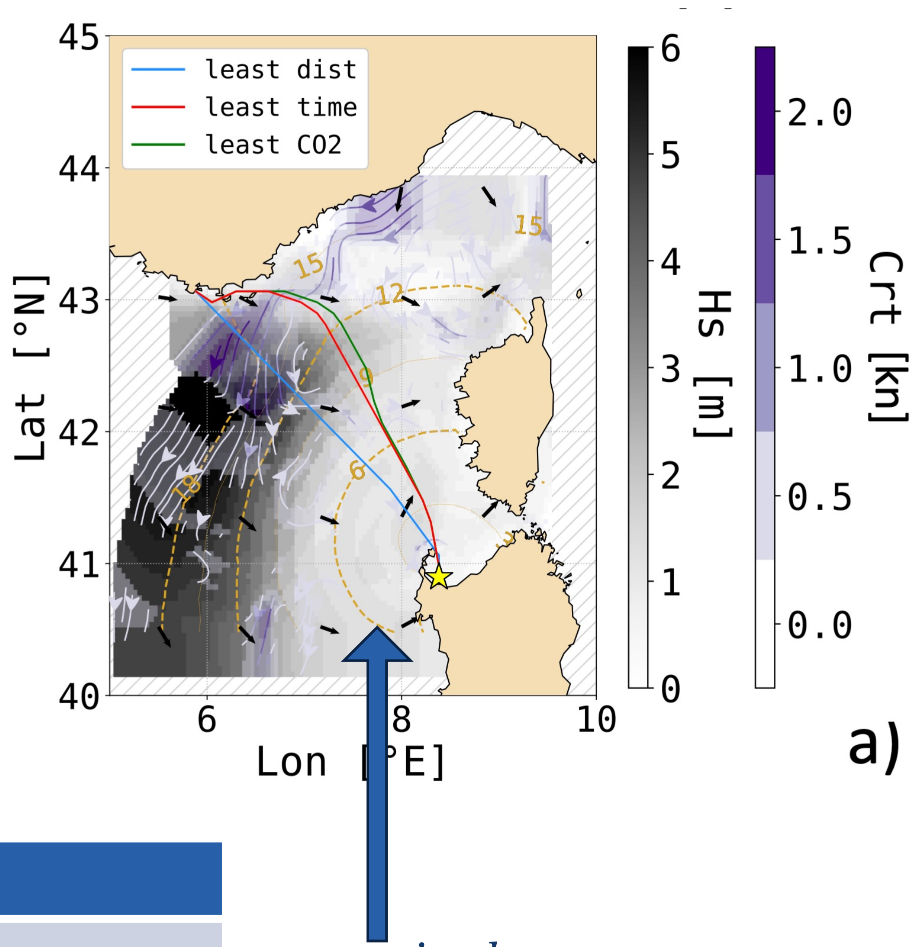
**) Mannarini et al 2019, doi.org/10.1109/TITS.2019.2935614*

Visualization

dynamic environmental fields rendered via

- *concentric shells originating at the departure location*
- *shape of shells defined by isochrones*

saving of 1 dimension (can be used for departure date or engine load)



isochrone

type	***meaning***	***bulging***
isometres	*equal distance*	*at obstructions (shoals, islands, landmass in general)*
isochrones	*equal duration*	*against gradients of 1/STW*
isopones	*equal emissions*	*against gradients of emissions*

Marine forecast data

dynamic environmental fields from data-assimilative models

type	*product*	*Spatial resolution*	*Time resolution*
Waves	*MEDSEA_ANALYSISFORECAST_WAV_006_017*	*(1/24)°* *2.5 miles*	*1 hour*
Currents	*MEDSEA_ANALYSISFORECAST_PHY_006_013*	*(1/24)°* *2.5 miles*	*1 hour*
Wind	*Set I - HRES*	*(1/10)°* *6.0 miles*	*3 hours*

Case study: ferry

One year of routes – video:
https://av.tib.eu/media/62912

Geography

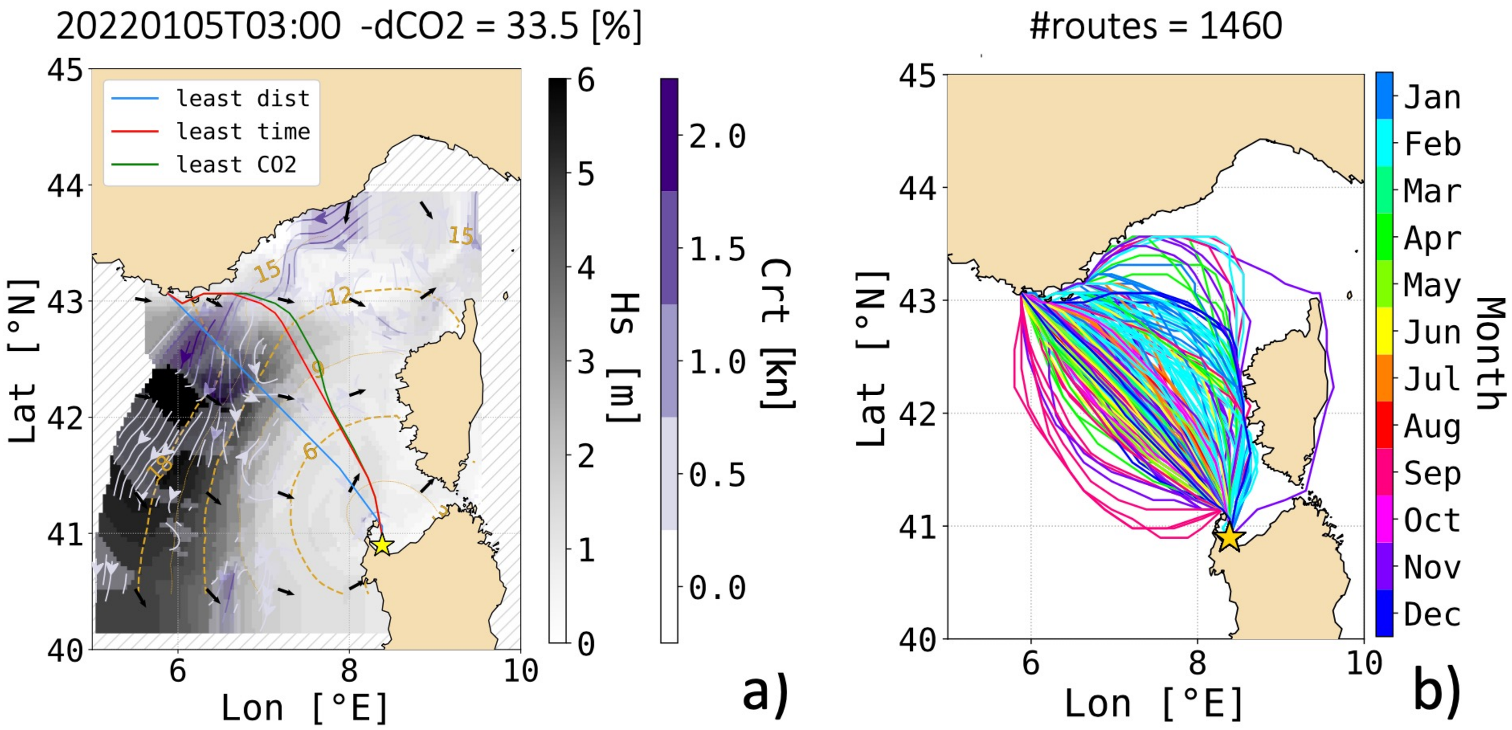
- *Mistral wind*
- *Liguro-Provençal current*

Numerical experiments

- *graph with* $(v, 1/\Delta x) = (4, 12/^{o})$
- *daily departures, 3 engine loads, two orientations, with/without currents (5840 runs)*
- *4 min/run*

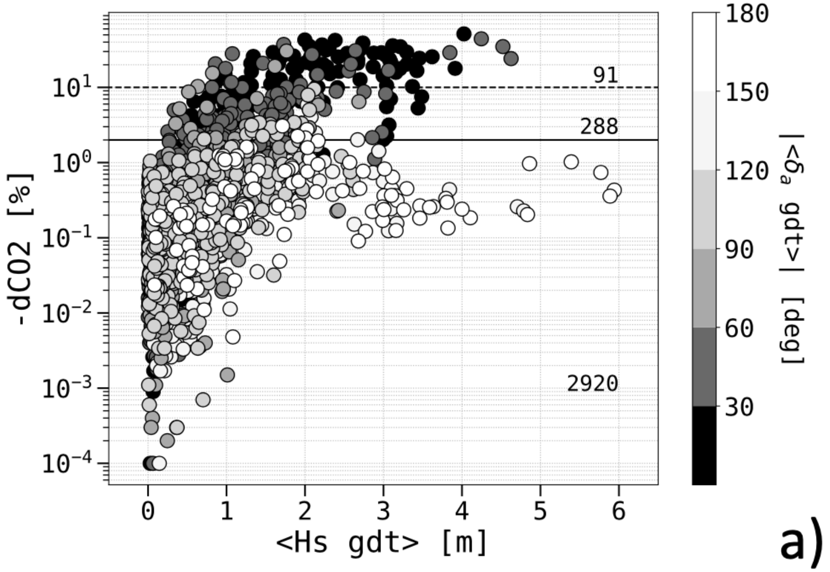
Outcome

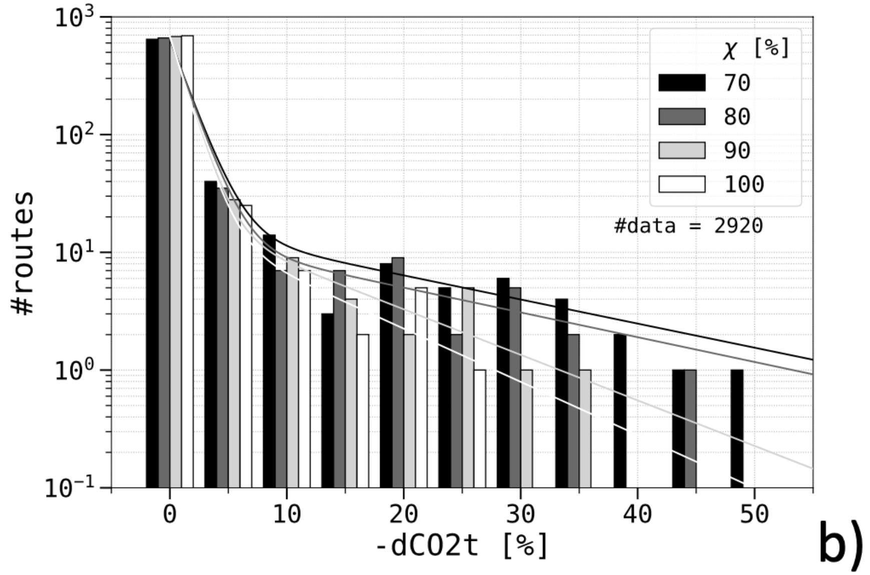
- *large diversions to avoid upwind sailing and exploit currents*
- *two-digit CO2 savings possible*
- *bundle of optimal solutions shifts N-E in winter*

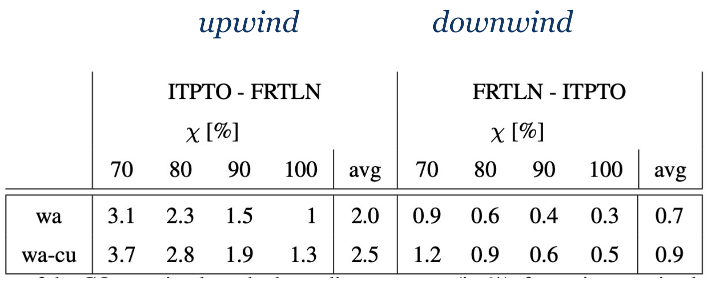


Case study: ferry

Statistics of ***CO2 savings*** *in 2022*







	upwind					*downwind*				
	ITPTO - FRTLN					FRTLN - ITPTO				
	χ [%]					χ [%]				
	70	80	90	100	avg	70	80	90	100	avg
wa	3.1	2.3	1.5	1	2.0	0.9	0.6	0.4	0.3	0.7
wa-cu	3.7	2.8	1.9	1.3	2.5	1.2	0.9	0.6	0.5	0.9

- *largest savings are upwind*
- *currents increase savings, especially downwind*

- *increase in wave height can lead to either substantial or minimal savings*
- *key is angle of attack of waves*
- *> 2% for beam or head seas*
- *>10% once a month, on average*

- *bi-exponential distribution*
- *larger decay length inversely proportional to engine load* χ
- *tail can extend to values ranging between 25 and 50%*

Case study: sailboat

One year of routes – video:
https://av.tib.eu/media/62913

Geography

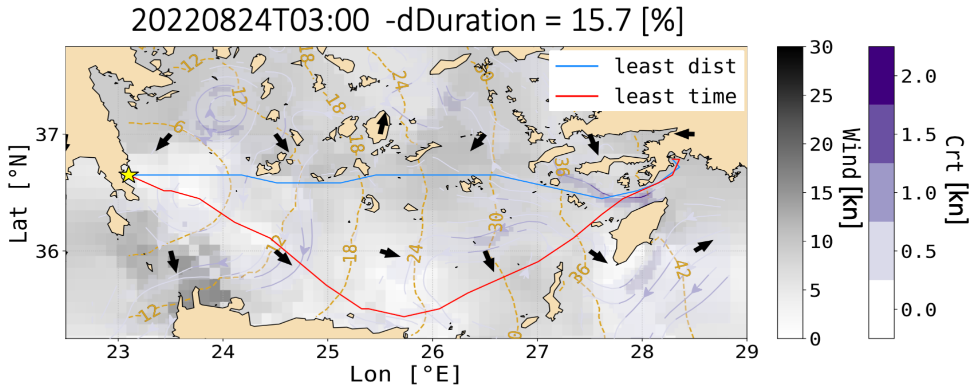
- *Meltemi wind*
- *Asia minor current*
- *archipelagic domain*

Numerical experiments

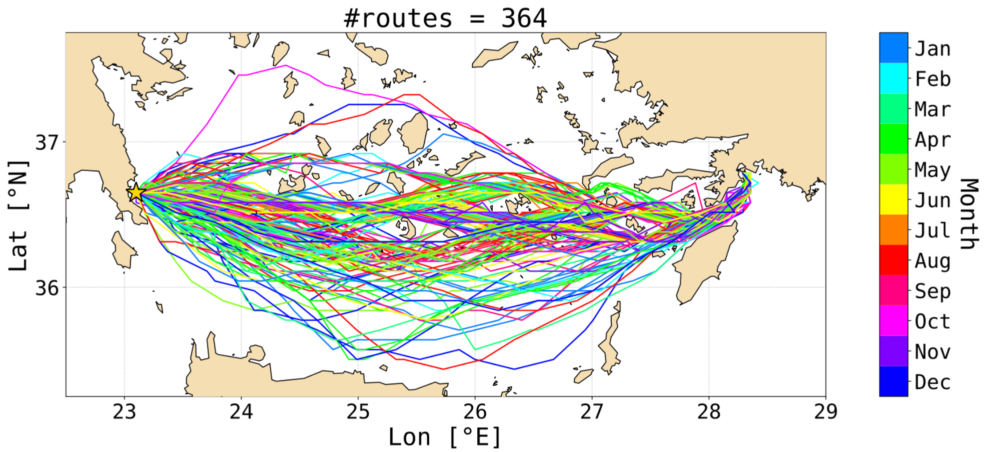
- *graph with* $(v, 1/\Delta x) = (5, 15/^{\circ})$
- *daily departures, two orientations, with/without currents or leeway (2,920 runs)*
- *7 min/run*

Outcome

- *large diversions to avoid upwind sailing*
- *no clear seasonal trend for diversions*



a)



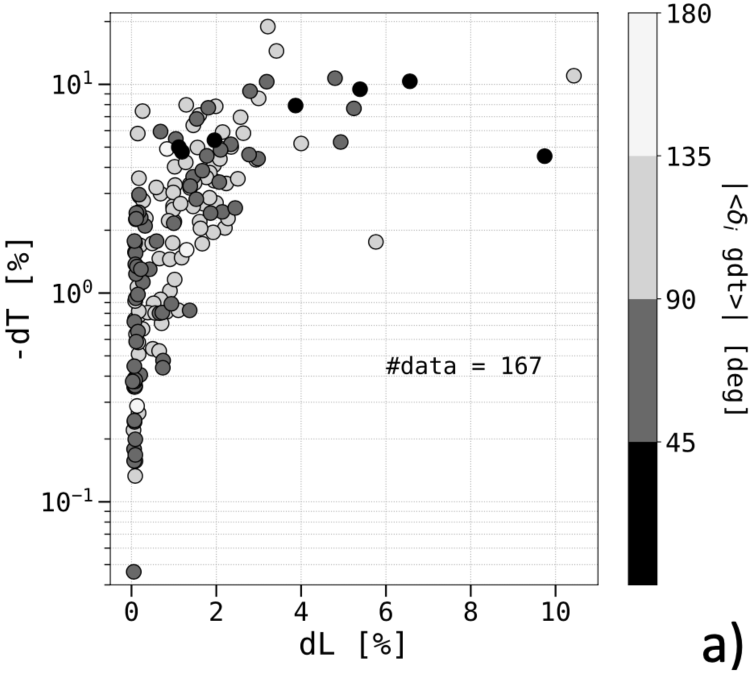
b)

Case study: sailboat

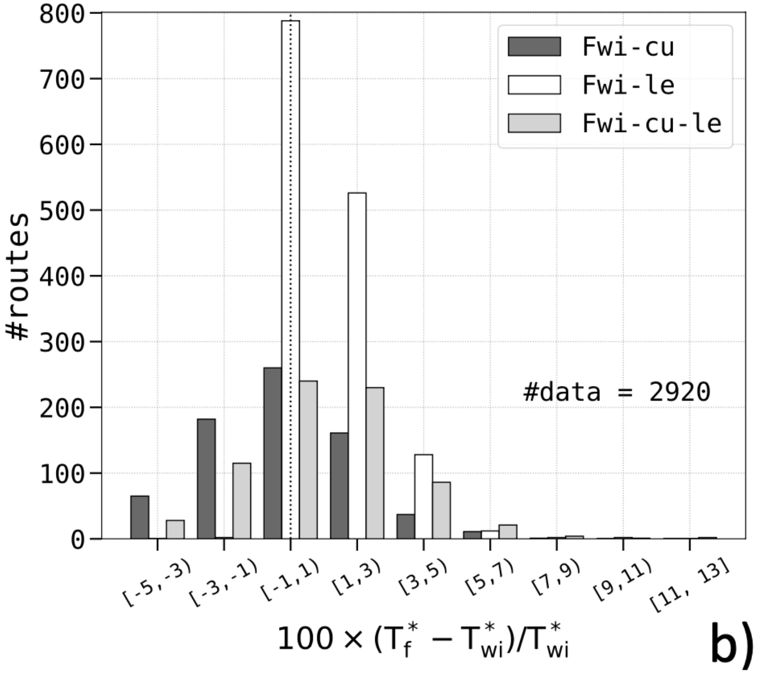
*Statistics of **time savings** in 2022*

	downwind *against current*			*upwind* *with current*		
	GRMON - TRMRM			TRMRM - GRMON		
	$-dT^*$	$N_f^{(g)}$	$N_f^{(o)}$	$-dT^*$	$N_f^{(g)}$	$N_f^{(o)}$
wi	3.0	263	1	3.0	300	1
wi-le	3.0	274	4	3.1	315	4
wi-cu	3.2	262	1	3.6	303	2
wi-cu-le	3.4	273	1	3.2	320	6

- *largest savings from currents when along sailing direction*
- *savings from leeway in downwind routes only thanks avoidance of speed loss along geodetic*



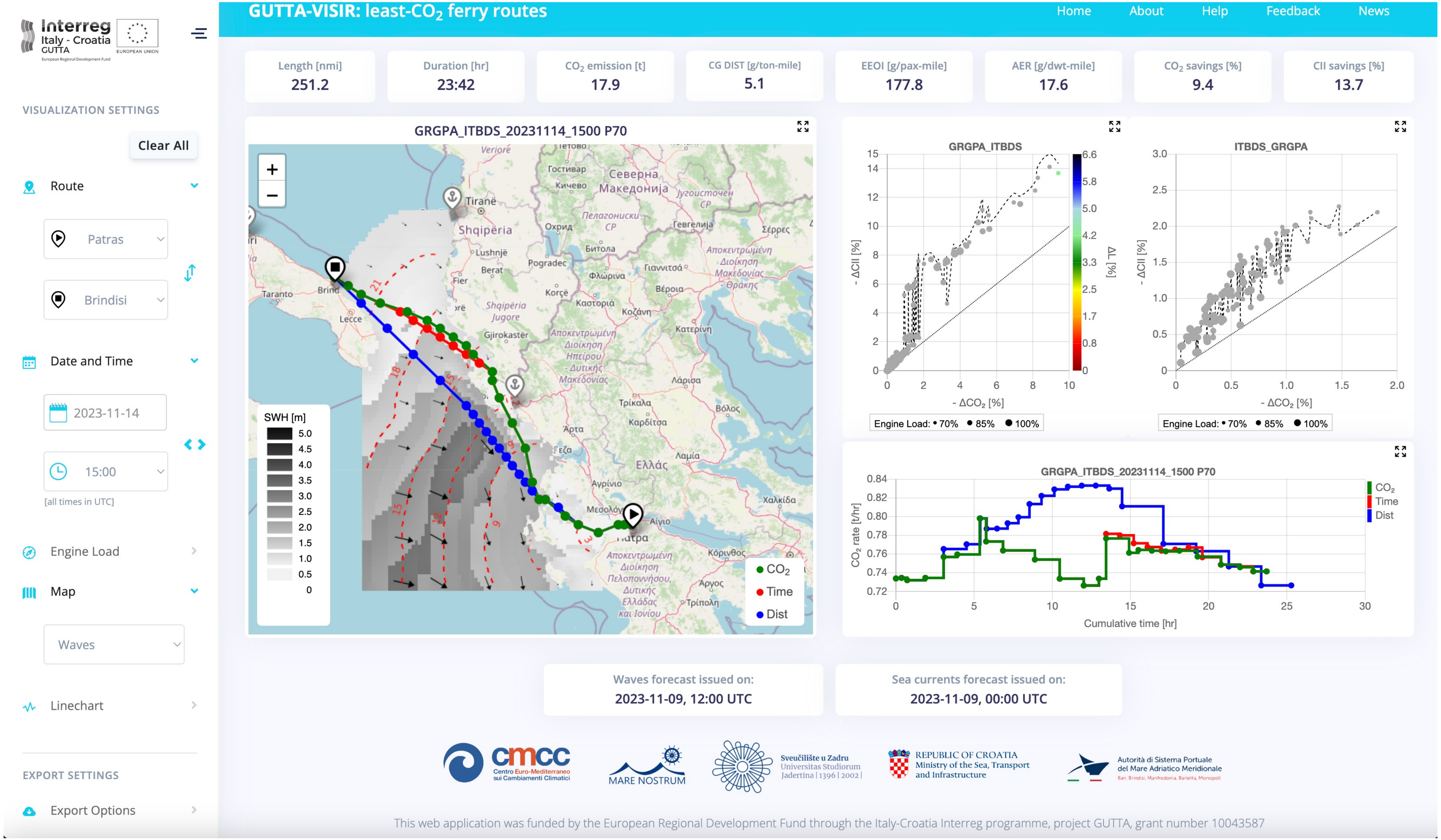
- *time saving increases with spatial diversion*
- *max saving for skipping upwind conditions along geodetic route*



- *currents results in a change in duration (slower/faster) up to about 5%*
- *leeway consistently extends the duration of routes (its cross-course component reduces SOG)*

Operational service: GUTTA-VISIR

https://www.gutta-visir.eu



GUTTA-VISIR: least-CO_2 ferry routes

Length [nmi]	Duration [hr]	CO_2 emission [t]	CG DIST [g/ton-mile]	EEOI [g/pax-mile]	AER [g/dwt-mile]	CO_2 savings [%]	CII savings [%]
251.2	23:42	17.9	5.1	177.8	17.6	9.4	13.7

VISUALIZATION SETTINGS

Route: Patras → Brindisi

Date and Time: 2023-11-14, 15:00 [all times in UTC]

Engine Load

Map: Waves

Linechart

EXPORT SETTINGS

Export Options

Waves forecast issued on: 2023-11-09, 12:00 UTC

Sea currents forecast issued on: 2023-11-09, 00:00 UTC

This web application was funded by the European Regional Development Fund through the Italy-Croatia Interreg programme, project GUTTA, grant number 10043587

Operational service: GUTTA-VISIR (video tutorial)

Bundle of least-CO2 routes

https://www.youtube.com/watch?v=-qORsU-Jh_8&t=4s

Results

- ✓ *VISIR-2: a modular, validated, documented, and portable model for ship weather routing*
- ✓ *for vessels with an angle-dependent performance curve, an improved level of accuracy in the velocity composition with sea currents*
- ✓ *variant of the Dijkstra's algorithm developed (minimise not just the CO2 emissions but any figure of merit depending on dynamic edge weights)*
- ✓ *quasi-linear computational performance up to 1 billion DOF*
- ✓ *10x faster than VISIR-1*
- ✓ *Bi-exponential distribution of CO2 savings found for a ferry*
- ✓ *sailboat routes: duration savings of about 3% , neglecting leeway would underestimate durations*

Possible uses of VISIR-2

❑ *inter-comparison studies*

❑ *creation of baseline numerical experiments*

❑ *weather routing of vessels with Wind-ASsisted Propulsion (WASP)*

❑ *narrowing the uncertainty about the potential of weather routing for CO2 emission reduction*

❑ *exploit generality of its algorithm for minimizing the consumption of costly zero-carbon fuel*

❑ *generate a dataset of optimal routes for the training of AI systems for autonomous vessels*

❑ *educational purposes (ship officials and maritime surveillance authorities, beginner sailors)*

Outlook

Computer Science

- *computational performance improvements for the least-distance procedure*
- *reduce the computer's memory allocation*
- *more use of object-oriented programming principles*

Naval architecture

- *vessel intact stability*
- *voluntary speed reduction*
- *considerations for slamming, green water, lateral acceleration and passenger comfort*

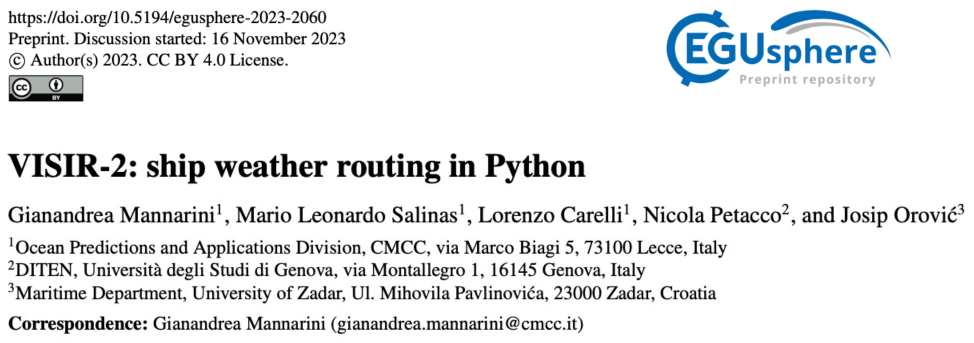
Algorithms

- *multi-objective optimisation techniques*
- *consideration of tacking time and motor-assistance for sailboats*
- *adaptive routing strategies (rerouting)*

VISIR-2 resources

open access – open review manuscript

[*https://egusphere.copernicus.org/preprints/2023/egusphere-2023-2060/*](https://egusphere.copernicus.org/preprints/2023/egusphere-2023-2060/)



https://doi.org/10.5194/egusphere-2023-2060
Preprint. Discussion started: 16 November 2023
© Author(s) 2023. CC BY 4.0 License.

VISIR-2: ship weather routing in Python

Gianandrea Mannarini[1], Mario Leonardo Salinas[1], Lorenzo Carelli[1], Nicola Petacco[2], and Josip Orović[3]

[1]Ocean Predictions and Applications Division, CMCC, via Marco Biagi 5, 73100 Lecce, Italy
[2]DITEN, Università degli Studi di Genova, via Montallegro 1, 16145 Genova, Italy
[3]Maritime Department, University of Zadar, Ul. Mihovila Pavlinovića, 23000 Zadar, Croatia

Correspondence: Gianandrea Mannarini (gianandrea.mannarini@cmcc.it)

---

open language - open source model code

[*https://zenodo.org/records/8305527*](https://zenodo.org/records/8305527)

Published September 7, 2023 | Version v1

[VISIR-2 ship weather routing model] source code

Salinas, Mario Leonardo[1]; Carelli, Lorenzo[1]; Mannarini, Gianandrea[1] cmcc

Centro Euro-Mediterraneo
sui Cambiamenti Climatici

www.cmcc.it