Reading and processing data for use in the LTER-LIFE digital twins: Waddensea Bathymetry and shape files

Karline Soetaert

first version: 01-10-2023; current version: 30 January 2024

This document prepares the bathymetric data that are to be used in the LTER life dtWad package. Shape files and bathymetric data are read and processed. For consistency of the bathymetry data with the other data sets, the depths are converted from lowest astronomical tide (LAT) to normaal amsterdams peil (NAP).

# Waddensea shape and bathymetry data

For many applications, we need to know whether data are located within the Waddensea, or we need to know the water depth at a certain location.

*Shapefiles* contains the boundaries of an area and can be used to select points that lie within or outside the area.

The *Bathymetry* record the depth at specific locations.

# Bathymetry

Bathymetry measures the depth of the sea. We will generate bathymetric data that contain water depths, according to mean sealevel, in a gridded form (a 2D map).

Lowest astronomical tide, *LAT* is the standard chart datum for bathymetric data; it is the lowest possible water level based on lunar and solar gravity.

In contrast to bathymetric data, all heights in the Netherlands are estimated as a function of the “Normaal Amsterdams Peil” in short *NAP* (Amsterdams ordnance datum). NAP is the Dutch Ordinance Level,and an NAP height of 0 m is more or less equal to the mean sealevel of the Northsea.

As the bathymetric data are expressed in lowest astronomical tide, *LAT*, and the water heights are in NAP, we need a conversion between these two data formats. Therefore the bathymetric data will be converted to NAP.

The bathymetry for the waddensea and surroundings was downloaded in *HDF5* format from EMODnet bathymetry: <http://www.emodnet-bathymetry.eu/>.

This dataset contains the mean water depth to Lowest Astronomical Tide (LAT) reference. The downloaded data is in gridded form, represented on a grid of 1/16 \* 1/16 arc minutes (ca. 115 meter grid).

Based on the downloaded dataset, three bathymetries are created:

* *WadDepthHR* has the bathymetry of the Dutch waddensea in high resolution (~115 m).
* *WadDepth* contains the bathymetry of the Dutch waddensea in low resolution (1/6th of the original), i.e. at ~700 m resolution.
* *Marsdiep* contains a section of the bathymetry, at high resolution (~115 m), zoomed on the Marsdiep area.

Only the last two datasets will be made part of the *dtWad* package.

## Reading the bathymetric data

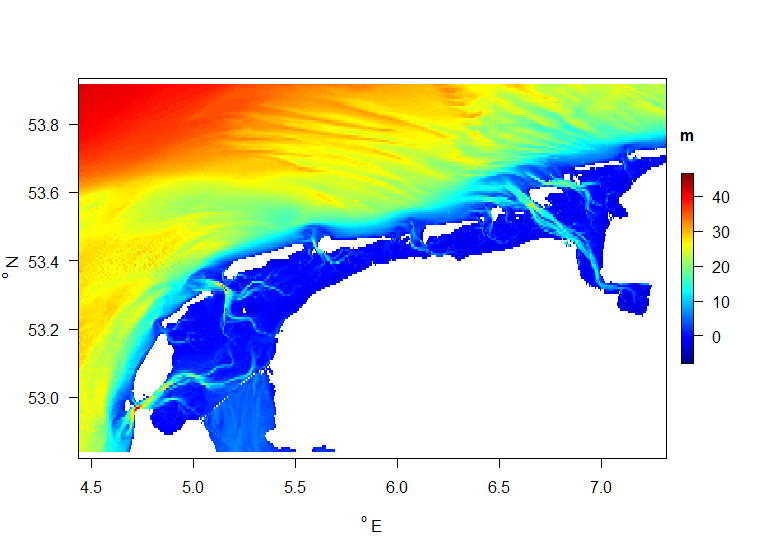
Bathymetric files that are exported from EMODnet in netCDF or HDF format can be read with R-function *readBathymetry* from the *dtLife* R-package.

The result is a list with the bathymetric information; while reading, we add the name of the person that created the file.

# Read the detail of the Marsdiep bathymetry  
Marsdiep <- readBathymetry(  
 dir = "../raw\_data/morphology",   
 file = "Mean depth in multi colour (no land).nc",  
 latlim = c(52.87, 53.1), # restrict latitudes  
 lonlim = c( 4.65, 5.1), # restrict longitudes  
 attr = list(originator="Karline Soetaert")  
 )  
# Waddensea bathymetry at 1/6th of resolution  
WadDepth <- readBathymetry(  
 dir = "../raw\_data/morphology",   
 file = "Mean depth in multi colour (no land).nc",   
 by = 6, # decrease resolution  
 attr = list(originator="Karline Soetaert")  
 )  
  
# Waddensea bathymetry at original high resolution  
WadDepthHR <- readBathymetry(  
 dir = "../raw\_data/morphology",   
 file = "Mean depth in multi colour (no land).nc",  
 attr = list(originator="Karline Soetaert")  
 )

The original bathymetric data extend into the German Waddensea.

This will change once we add the NAP-LAT conversion to it (see below).



Original bathymetry of the Waddensea, low resolution, LAT datum

## Conversion from LAT to NAP

### NLLAT2018

The dataset *NLLAT2018* (from <https://english.defensie.nl/topics/hydrography/documents/applications/2020/06/12/nllat2018>) illustrates how NAP and LAT relate to each other. This data set was made available by the Technical University of Delft.

The *NLLAT2018* dataset is a combination of two datasets, for the open sea and for inland waters. The LAT for inland waters (Westerschelde, Oosterschelde and Wadden Sea) was released in 2006; the data for open waters was released in 2018. The accuracy for LAT is better than 10cm, the accuracy of the geoid is better than 3 cm (at sea).

The units in this dataset are [m] for height. Coordinates are in ETRS89 (= EPSG 25831)

Note the terms of use for this product (from the website above):

* This product is free to download. Use of the product for any purpose is subject to the following conditions:
  + The data provided may not be used for navigation, because the data sets are not linked and are not updated according to the Notices to Mariners.
  + The Hydrographic Service reserves the right to exercise its moral rights (Article 6bis, Berne Convention) in respect of the reproduced materials that contain the data provided.

The dataset is processed as follows:

* We read the entire dataset (text file); the data are in long format (latitude, longitude, NAPminLAT)
* A subset for the Waddensea area (including the adjacent Northsea) is taken.
* The data are then gridded on the Waddensea bathymetry grids and on the Marsdiep bathymetry grid.
* The gridded NAP-LAT conversions are added to the WadDepth, WadDepthHR and Marsdiep bathymetric object
* The processed bathymetric files are saved to binary R-files (rda format), ready to incorporate in the dtWad R-package.

### Reading the NNLLAT2018 data

The NLLAT2018 data are in long format, i.e. it is a long table with 5 columns, that that contain, for multiple points along the Dutch coast:

* latitude (ETRS89)
* longitude (ETRS89)
* the NAP (quasi-geoid) height above ETRS89 (ellipsoid),
* the LAT height above ETRS89 ellipsoid, and
* the NAP height above LAT.

The file is a text file. It is read as a *data.frame*, and the metadata are added as attributes to the object.

These metadata comprise:

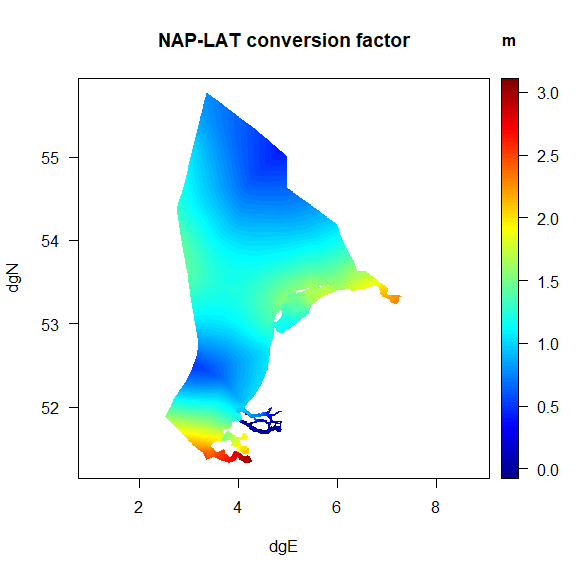
* a description of the columns(variables) in the dataset, with the units
* the datasource (website, and name of the original file)
* the terms of use (as from the website where it was downloaded)
* the aspect ratio for plotting (this depends on the latitude)
* some processing information

The data.frame is then saved as an R binary file.

NAP\_LAT <- read.table(file = "../raw\_data/morphology/NLLAT2018.txt",   
 skip = 5,   
 header = FALSE)  
names(NAP\_LAT) <- c("latitude", "longitude", "NAP", "LAT", "NAPminLAT")  
  
NAP\_LAT <- NAP\_LAT[,c(2, 1, 3, 4, 5)] # rearrange the order  
  
# description of the variables in this dataset  
vars <- data.frame(  
 variable = c("longitude", "latitude",   
 "NAP", "LAT", "NAPminLAT"),  
 description = c("ETRS89 longitude", "ETRS89 latitude",   
 "NAP quasi geoid height above ETRS89 ellipsoid",  
 "Lowest astronomical tide height above ETRS89 ellipsoid",  
 "NAP quasi geoid height above ETRS89 ellipsoid minus Lowest astronomical tide height above ETRS89 ellipsoid"),  
 units = c("dgE", "dgN",   
 "m", "m", "m"))  
  
# add to the attributes  
attributes(NAP\_LAT)$variables <- vars   
attributes(NAP\_LAT)$ETRS <- 89   
  
# data source and terms of use  
attributes(NAP\_LAT)$datasource <- "https://english.defensie.nl/topics/hydrography/documents/applications/2020/06/12/nllat2018)"  
attributes(NAP\_LAT)$file <- "NLLAT2018.txt"  
attributes(NAP\_LAT)$termsOfUse <- "not for navigation; The Hydrographic Service reserves the right to exercise its moral rights (Article 6bis, Berne Convention) in   
respect of the reproduced materials that contain the data provided"  
  
# aspect ratio for plotting  
attributes(NAP\_LAT)$asp <- 1/cos((mean(NAP\_LAT$latitude) \* pi)/180)  
  
# processing information  
attributes(NAP\_LAT)$processing <- paste("Created at", Sys.time(), "by Karline Soetaert")  
class(NAP\_LAT) <- c("dtWad", "data.frame")  
  
save(file="../processed\_data/NAP\_LAT.rda", NAP\_LAT)

The NAP-LAT conversion factors are plotted:

asp <- attributes(NAP\_LAT)$asp # figure aspect ratio  
with(NAP\_LAT,   
 points2D(x = longitude,   
 y = latitude,   
 colvar = NAPminLAT, # color according to NAP-LAT  
 main = "NAP-LAT conversion factor",   
 xlab = "dgE", ylab = "dgN", clab = "m",  
 pch = ".", asp=asp, las=1)  
)



NAP-LAT conversion dataset

### Subsetting the NLLAT2018 data

The NLLAT2018 dataset is rather large, comprising 138816 datapoints, ranging in between 51.325 and 55.7625 degrees north and 2.54 and 7.31 degrees east.

We first make a selection of the data, keeping only the points that fall within the waddensea area.

This new dataset is called *NapLatWad*.

# suitable area spanning the waddensea bathymetry data  
xlim <- range(WadDepth$longitude) + c(-0.1, 0.1)  
ylim <- range(WadDepth$latitude) + c(-0.1, 0.1)  
  
NapLatWad <- subset(NAP\_LAT,   
 subset= longitude >= xlim[1] & longitude <= xlim[2] &   
 latitude >= ylim[1] & latitude <= ylim[2])  
  
# keep only three columns  
NapLatWad <- NapLatWad[,c("longitude", "latitude", "NAPminLAT")]

This has reduced the number of datapoints to 30249.

### Gridding the NLLAT2018 data

The *NapLatWad* data, in long format needs to be added to the three bathymetry datasets that we created and that are represented as 2-dimensional grids (i.e. wide format).

Therefore, the NapLatWad data need to be gridded to match the 2D grids from the three bathymetric data sets.

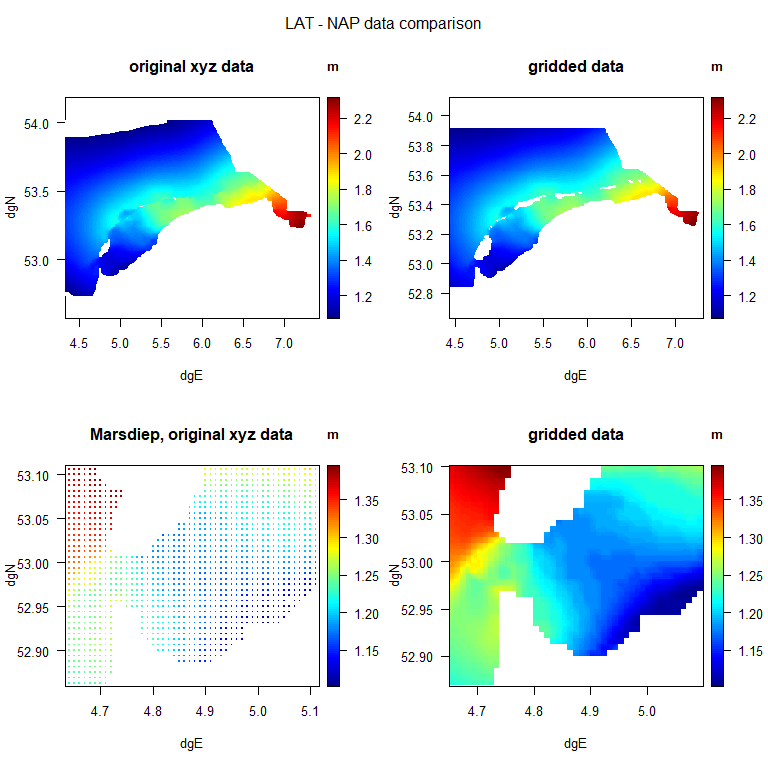
This is done in several steps, using function *map\_xy* from the *dtLife* R-package:

* First the NapLatWad data are mapped from long format to 2D format, using its own grid sizes
* Then the 2D NapLatWad data are mapped on the various grids of bathymetric data

#--------------------------------------------------  
# first map the NapLatWad data to its own 2D grid   
#--------------------------------------------------  
  
# x- and y- grid sizes of the NapLatWad data  
x <- unique(NapLatWad$longitude) # unique x values  
x <- sort(x)  
dx <- min(diff(x)) # distances between x  
  
y <- unique(NapLatWad$latitude)  
y <- sort(y)  
dy <- min(diff(y))  
  
# output grid for the 2D NaplatWad data  
output.x <- seq(min(x), max(x), by=dx)   
output.y <- seq(min(y), max(y), by=dy)  
  
# map data from long format to 2D gridded format   
NapLatWad2D <- map\_xy(  
 input.xyv = NapLatWad,   
 output.x = output.x,   
 output.y = output.y)  
  
#--------------------------------------------------  
# map the 2D data to the bathymetric grids  
#--------------------------------------------------  
  
# from NapLatWad2D grid to the waddensea bathymetry LR grid  
NL\_bat <- map\_xy(   
 input.x = NapLatWad2D$longitude,   
 input.y = NapLatWad2D$latitude,   
 input.2D = NapLatWad2D$NAPminLAT,   
   
 output.x = WadDepth$longitude,   
 output.y = WadDepth$latitude  
 )  
  
# from NapLatWad2D grid to the waddensea bathymetry HR grid  
NL\_batHR <- map\_xy(  
 input.x = NapLatWad2D$longitude,   
 input.y = NapLatWad2D$latitude,   
 input.2D = NapLatWad2D$NAPminLAT,   
   
 output.x = WadDepthHR$longitude,   
 output.y = WadDepthHR$latitude  
 )  
  
# from NapLatWad2D grid to the Marsdiep bathymetry grid  
NL\_batMarsdiep <- map\_xy(  
 input.x = NapLatWad2D$longitude,   
 input.y = NapLatWad2D$latitude,   
 input.2D = NapLatWad2D$NAPminLAT,   
   
 output.x = Marsdiep$longitude,   
 output.y = Marsdiep$latitude  
 )

The original (xyz) data are compared to the gridded data, to show correspondence.

par (mfrow=c(2,2), oma=c(0,0,2,0), las=1)  
  
asp <- 1/cos((mean(NL\_bat$y) \* pi)/180) # figure aspect ratio  
  
# plot for the entire waddensea  
xlim <- range(NL\_bat$x)  
ylim <- range(NL\_bat$y)  
zlim <- range(NL\_bat$z, na.rm=TRUE)  
with (NapLatWad,   
 points2D(x = longitude,   
 y = latitude,   
 colvar = NAPminLAT, # color variable  
 xlim = xlim, ylim = ylim, clim = zlim, # plot ranges  
 clab = "m", xlab = "dgE", ylab = "dgN",   
 main = "original xyz data",  
 asp = asp, # aspect ratio  
 pch = ".", cex = 4)  
 )  
with (NL\_bat,   
 image2D(x = x,   
 y = y,   
 z = z,   
 xlab = "dgE", ylab = "dgN", clab = "m",   
 main = "gridded data",   
 asp = asp)  
)  
  
# plot for the Marsdiep area  
xlim <- range(NL\_batMarsdiep$x)  
ylim <- range(NL\_batMarsdiep$y)  
zlim <- range(NL\_batMarsdiep$z, na.rm=TRUE)  
  
with (NapLatWad,   
 points2D(longitude, latitude,   
 colvar = NAPminLAT, # color variable  
 xlim = xlim, ylim = ylim, clim = zlim, # plot ranges  
 clab = "m", xlab = "dgE", ylab = "dgN",   
 main = "Marsdiep, original xyz data",  
 asp = asp, # aspect ratio  
 pch = ".", cex = 2))  
  
with (NL\_batMarsdiep,  
 image2D(x = x,   
 y = y,   
 z = z,   
 xlab = "dgE", ylab = "dgN", clab = "m",   
 main = "gridded data",   
 asp = asp)  
)  
   
mtext(outer=TRUE, side=3, line=0, "LAT - NAP data comparison")



Correspondence of original and gridded LAT-NAP data

### Converting bathymetric depths from LAT to NAP

The gridded LAT-NAP values are now added to the bathymetric depths, to convert depths from LAT to NAP. A note is added to the variables description to signify this transformation.

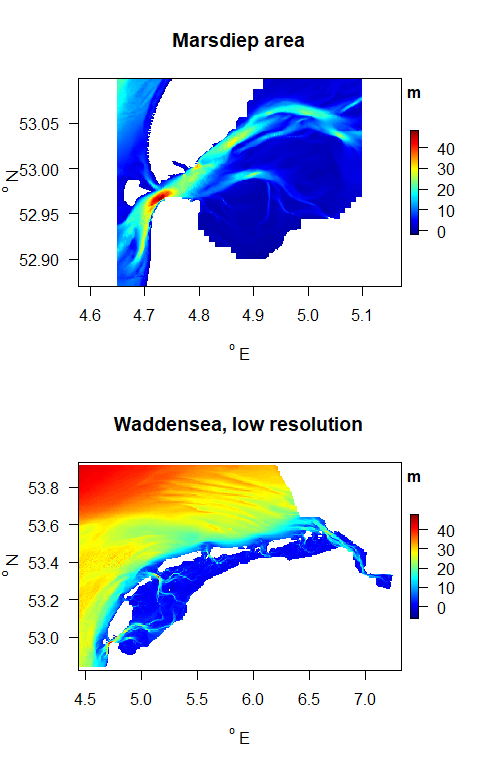
The results are written as binary R-files.

newnote <- c("added LAT-NAP conversion from NLLAT2018",   
 "final data: depth versus NAP (Normaal Amsterdams Peil)")  
   
Marsdiep$depth <- Marsdiep$depth + NL\_batMarsdiep$z  
Marsdiep$variables$note <- c(Marsdiep$variables$note,   
 newnote)  
  
WadDepth$depth <- WadDepth$depth + NL\_bat$z  
WadDepth$variables$note <- c(WadDepth$variables$note,   
 newnote)  
  
WadDepthHR$depth <- WadDepthHR$depth + NL\_batHR$z  
WadDepthHR$variables$note <- c(WadDepthHR$variables$note,   
 newnote)  
  
save(file="../processed\_data/Marsdiep.rda", Marsdiep)  
save(file="../processed\_data/WadDepth.rda", WadDepth)  
save(file="../processed\_data/WadDepthHR.rda", WadDepthHR)

## Plotting the final bathymetry

The R-package *dtLife* also contains a simple function to plot these bathymetric maps: *plotBathymetry*. This function is based on function *image2D* from the *plot3D* package.

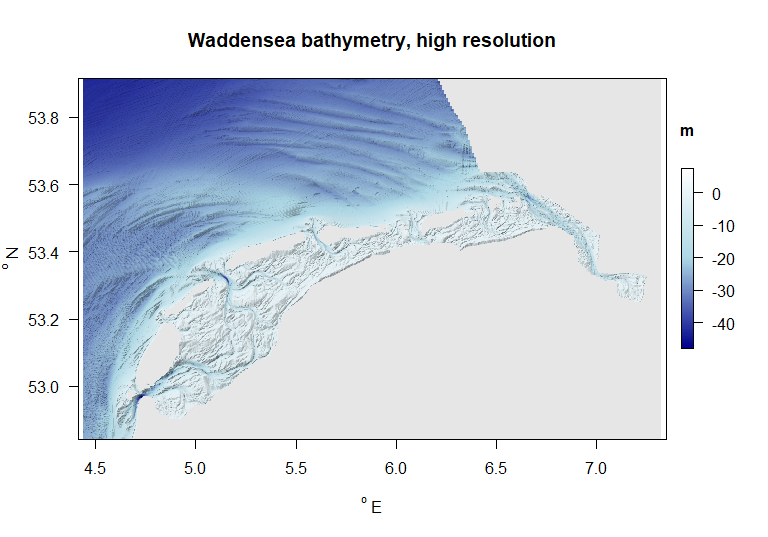
par(mfrow=c(2,1))  
plotBathymetry(Marsdiep, main="Marsdiep area")  
plotBathymetry(WadDepth, main="Waddensea, low resolution")



Final bathymetry, in NAP datum

In the last example, the high resolution map is plotted with a white-blue color scheme, textured with shaded relief. Shading is an often used technique that mimics how the sun would drape the landscape. By casting shades, it render the map a three-dimensional appearance.

plotBathymetry(WadDepthHR,   
 NAcol = grey(0.9),  
 col = ramp.col(col=c("darkblue", "lightblue", "white")),  
 shade = 0.1, negativeDepth = TRUE,  
 main = "Waddensea bathymetry, high resolution")



Final high-resolution bathymetry, in NAP datum

# Shape file

The Waddensea shapefile contains the boundaries of the area. This will be used to select points that lie within the Waddensea.

Shapefiles can be found in <https://marineregions.org>.

The Waddensea shape file was downloaded from <https://marineregions.org/gazetteer.php?p=details&id=26877>.

The shape file source is the “World Heritage Marine Programme”; its ranges are between 4.7157 and 9.0169 E and 52.8877 and 55.5973 N.

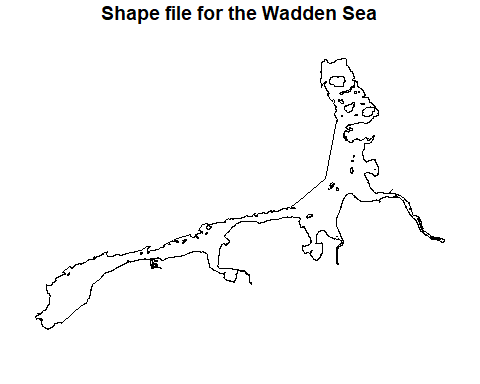
## reading the shape file

Shapefiles can be read with function *st\_read* from the *sf* package.

dir <- "../raw\_data/shapefile/seavox"  
WadShape <- st\_read(paste(dir, "seavox\_v19.shp", sep="/"))[1]

## Reading layer `seavox\_v19' from data source   
## `C:\Users\karlines\OneDrive - NIOZ\Documents\git\LTER\dtR\data4dt\raw\_data\shapefile\seavox\seavox\_v19.shp'   
## using driver `ESRI Shapefile'  
## Simple feature collection with 1 feature and 20 fields  
## Geometry type: POLYGON  
## Dimension: XY  
## Bounding box: xmin: 4.724179 ymin: 52.89209 xmax: 9.835879 ymax: 54.88455  
## Geodetic CRS: WGS 84

plot(WadShape,   
 main="Shape file for the Wadden Sea")



Shape <- WadShape  
save(file="../processed\_data/WadShape.rda", WadShape)  
save(file="../processed\_data/Shape.rda", Shape)

## uses of the shape file

Shapefiles can be used to select data that lie within the shape.

The R-package *dtLife* contains two function to mark the cells that do not fall within a shape file with *NA*.

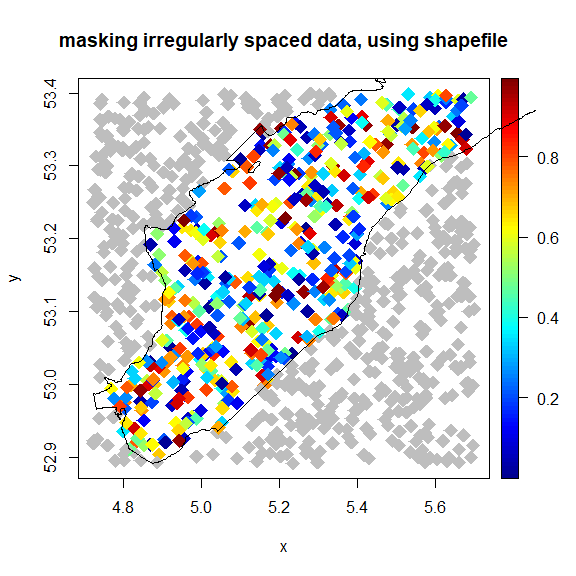
* function *mask\_xy* masks data in long format
* function *mask\_2D* does the same for data in gridded format, i.e. bathymetric data:

The use of *mask\_xy* is demonstrated by creating a grid with irrregularly spaced coordinates and then masking them according to the waddensea shapefile.

## create irregular spaced coordinates  
xy <- data.frame (x = runif(min=4.724179, max= 5.7, n=1000),   
 y = runif(min=52.89209, max=53.4, n=1000),  
 z = runif(min=0, max=1, n=100))  
  
# use mask\_xy to   
xy <- mask\_xy(coordinates=xy, shape=WadShape)

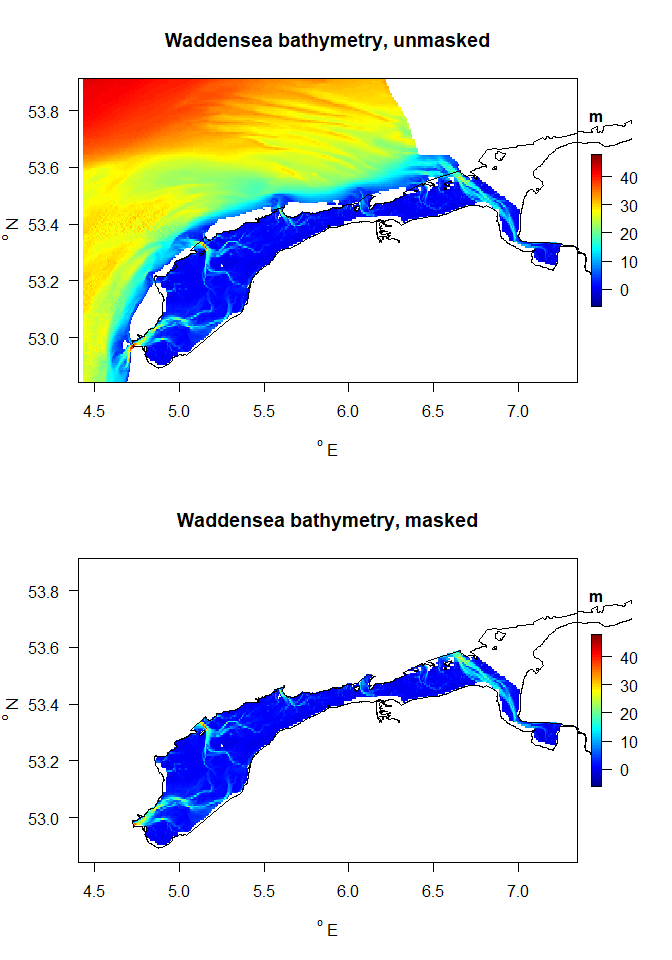
## st\_as\_s2(): dropping Z and/or M coordinate

# make z-values outside of the shape = NA  
xy$z[is.na(xy$mask)] <- NA   
  
# plot it, and add the shape  
points2D(x = xy$x,   
 y = xy$y,   
 colvar= xy$z,   
 main = "masking irregularly spaced data, using shapefile",  
 pch=18, cex=2, NAcol="grey")  
plot(WadShape[1], add=TRUE)



The use of *mask\_2D* is demonstrated by masking the bathymetry data, using the waddensea shapefile.

# regular spaced coordinates: the bathymetry data  
Batmask <- mask\_bat(WadDepth, shape=WadShape)  
Batmask$depth <- WadDepth$depth \* Batmask$mask  
  
# unmasked plot  
par(mfrow=c(2,1), las=1)  
plotBathymetry(WadDepth,   
 main = "Waddensea bathymetry, unmasked")  
plot(WadShape[1], add=TRUE)  
  
# masked plot  
plotBathymetry(Batmask,   
 main = "Waddensea bathymetry, masked")  
plot(WadShape, add=TRUE)



# References

The following R-sources were used for this work:

R-core:

* R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

R-package dtWad,

* Soetaert K (2024). dtWad: Waddensea Digital Twin: general utilities. R package version 0.0.1.

R-package sf

* Pebesma, E., & Bivand, R. (2023). Spatial Data Science: With Applications in R. Chapman and Hall/CRC. <https://doi.org/10.1201/9780429459016>

R-package plot3D

* Soetaert K (2021). plot3D: Plotting Multi-Dimensional Data. R package version 1.4, <https://CRAN.R-project.org/package=plot3D>.