Test of package **OceanView** , visualisation of oceanographic data.

Karline Soetaert
NIOZ-Yerseke
The Netherlands

Abstract

Plotting the output of a 3-D hydrodynamic model applied to the North Sea, as produced by the GETM software.

The text in `boxes` = R-code; the text below boxes in this `format` (if any) is output, produced by R.

Keywords: marine science, hydrodynamic models, 3-D data, 4-D data, quiver, image2D, R .

1. Preamble

Load the required packages: **RNetCDF** (Michna 2011) for reading netcdf files, **OceanView** (Soetaert in prep) for visualising output.

```
require(RNetCDF)
require(OceanView)
```

Open the net cdf file(s) and print its content:

```
d.nc <- open.nc( "ns_06nm_sept_1997_3d.nc")
print.nc(d.nc)
```

```
dimensions:
        xc = 111 ;
        yc = 87 ;
        level = 26 ;
        time = UNLIMITED ; // (30 currently)
variables:
        int grid_type ;
        int vert_cord ;
        int ioff ;
                ioff:long_name = "index offset (i)" ;
        int joff ;
                joff:long_name = "index offset (j)" ;
        double dx ;
                dx:units = "m" ;
```

```
        dx:long_name = "grid spacing (x)" ;
double dy ;
        dy:units = "m" ;
        dy:long_name = "grid spacing (y)" ;
double xc(xc) ;
        xc:units = "m" ;
double yc(yc) ;
        yc:units = "m" ;
double lonc(xc, yc) ;
        lonc:units = "degrees_east" ;
        lonc:long_name = "longitude" ;
        lonc:valid_range = -180180 ;
        lonc:_FillValue = -999 ;
        lonc:missing_value = -999 ;
double latc(xc, yc) ;
        latc:units = "degrees_north" ;
        latc:long_name = "latitude" ;
        latc:valid_range = -9090 ;
        latc:_FillValue = -999 ;
        latc:missing_value = -999 ;
double convc(xc, yc) ;
        convc:units = "degrees" ;
        convc:long_name = "grid rotation" ;
        convc:valid_range = -180180 ;
        convc:_FillValue = -999 ;
        convc:missing_value = -999 ;
double latu(xc, yc) ;
        latu:units = "degrees" ;
        latu:long_name = "latu" ;
        latu:valid_range = -9090 ;
        latu:_FillValue = -999 ;
        latu:missing_value = -999 ;
double latv(xc, yc) ;
        latv:units = "degrees" ;
        latv:long_name = "latv" ;
        latv:valid_range = -9090 ;
        latv:_FillValue = -999 ;
        latv:missing_value = -999 ;
double level(level) ;
        level:units = "level" ;
double bathymetry(xc, yc) ;
        bathymetry:units = "m" ;
        bathymetry:long_name = "bathymetry" ;
        bathymetry:valid_range = -54000 ;
        bathymetry:_FillValue = -10 ;
        bathymetry:missing_value = -10 ;
float time(time) ;
        time:units = "seconds since 1997-09-01 00:00:00" ;
        time:long_name = "time" ;
float elev(xc, yc, time) ;
        elev:units = "m" ;
        elev:long_name = "elevation" ;
        elev:valid_range = -1515 ;
```

```
        elev:_FillValue = -9999 ;
        elev:missing_value = -9999 ;
float u(xc, yc, time) ;
        u:units = "m/s" ;
        u:long_name = "int. zonal vel." ;
        u:valid_range = -33 ;
        u:_FillValue = -9999 ;
        u:missing_value = -9999 ;
float v(xc, yc, time) ;
        v:units = "m/s" ;
        v:long_name = "int. meridional vel." ;
        v:valid_range = -33 ;
        v:_FillValue = -9999 ;
        v:missing_value = -9999 ;
float h(xc, yc, level, time) ;
        h:units = "m" ;
        h:long_name = "layer thickness" ;
        h:_FillValue = -9999 ;
        h:missing_value = -9999 ;
float hcc(xc, yc, level) ;
        hcc:units = "" ;
        hcc:long_name = "hcc" ;
        hcc:valid_range = 01 ;
        hcc:_FillValue = -1 ;
        hcc:missing_value = -1 ;
float uu(xc, yc, level, time) ;
        uu:units = "m/s" ;
        uu:long_name = "zonal vel." ;
        uu:valid_range = -33 ;
        uu:_FillValue = -9999 ;
        uu:missing_value = -9999 ;
float vv(xc, yc, level, time) ;
        vv:units = "m/s" ;
        vv:long_name = "meridional vel." ;
        vv:valid_range = -33 ;
        vv:_FillValue = -9999 ;
        vv:missing_value = -9999 ;
float w(xc, yc, level, time) ;
        w:units = "m/s" ;
        w:long_name = "vertical vel." ;
        w:valid_range = -33 ;
        w:_FillValue = -9999 ;
        w:missing_value = -9999 ;
float salt(xc, yc, level, time) ;
        salt:units = "PSU" ;
        salt:long_name = "salinity" ;
        salt:valid_range = 040 ;
        salt:_FillValue = -9999 ;
        salt:missing_value = -9999 ;
float temp(xc, yc, level, time) ;
        temp:units = "degC" ;
        temp:long_name = "temperature" ;
        temp:valid_range = 040 ;
```

```
            temp:_FillValue = -9999 ;
            temp:missing_value = -9999 ;

// global attributes:
            :title = "North Sea - 6nm" ;
            :history = "Generated by getm, ver. 1.8.0" ;
```

1.1. Reading the data

```
lat_c <- var.get.nc(d.nc, "latc")
lon_c <- var.get.nc(d.nc, "lonc")
time  <- var.get.nc(d.nc, "time")
level <- var.get.nc(d.nc, "level")
convc <- var.get.nc(d.nc, "convc")
xc    <- var.get.nc(d.nc, "xc") / 1000  # in km
yc    <- var.get.nc(d.nc, "yc") / 1000
dx    <- var.get.nc(d.nc, "dx") / 1000
dy    <- var.get.nc(d.nc, "dy") / 1000
hcc   <- var.get.nc(d.nc, "hcc")
h     <- var.get.nc(d.nc, "h")
bathy <- var.get.nc(d.nc, "bathymetry")
# get 2-D data
u     <- var.get.nc(d.nc, "u")
v     <- var.get.nc(d.nc, "v")
elev  <- var.get.nc(d.nc, "elev")
temp  <- var.get.nc(d.nc, "temp")
ww    <- var.get.nc(d.nc, "w")
uu    <- var.get.nc(d.nc, "uu")
vv    <- var.get.nc(d.nc, "vv")
```

Show some values

```
dx
```

```
[1] 11.112
```

```
dy
```

```
[1] 11.112
```

The ranges of the velocities

```
range(uu, na.rm = TRUE)
```

```
[1] -1.217723  1.751082
```

```
range(vv, na.rm = TRUE)
```

```
[1] -1.303300  2.244228
```

```
range(ww, na.rm = TRUE)
```

```
[1] -0.01644286  0.02704342
```

1.2. Box depths

There does not seem to be an output variable that has the depth, in the middle of each compartment.

Here it is created from the thickness of each box.

First we take the temporal mean of the box thickness:

```
hh <- apply(h, MARGIN = 1:3, FUN = mean)
```

Then we take the cumulative sum of the box thicknesses. This gives the depth at the box interfaces:

```
depth <- hh
depth[ , ,1] <-  0
for (i in 26:2)
  depth[ , , i-1] <- depth[ , , i-1] + depth[ , , i]
```

Then we take the mean at each interface, to get the depth at the centre of a box.

```
for (i in 2:26)
  depth[ , , i] <- 0.5*(depth[ , , i-1] + depth[ , , i])
```

2. Plot bathymetry and grid

A suitable color scheme, not too dark.

```
par(mfrow = c(2, 2))
col <- jet2.col(100)
#
# Plot every fourth grid point
plot(x = lon_c[seq(1, length(lon_c), by = 4)],
     y = lat_c[seq(1, length(lat_c), by = 4)],
     xlab = "longitude", ylab = "latitude",
     pch = ".", main = "grid points")
legend("bottom", legend = paste (c("x: ", "y: "), dim(lon_c)))
#
# Plot the bathymetry on the grid (distorted)
image2D(bathy, x = xc, y = yc, NAcol = "black",
      col = col, xlab = "km", ylab = "km",
      main = "Bathymetry on grid", clab = c("","","m"))
#
# Map the bathymetry on regular coordinates
image2D(bathy, x = lon_c, y = lat_c, NAcol = "black",
      col = col, main = "Bathymetry on coordinates",
      xlab = "longitude", ylab = "latitude", clab = c("","","m"))
#
# add transect line
lines(lon_c[30,], lat_c[30,]) # from where the sigma grid is plotted
#
# plot the sigma grid on transect line
matplot(x = yc, depth[30,,], type = "l", col = "black",
        main = "sigma grid", ylab = "depth, m", xlab = "y",
        lty = 1, ylim = c(140, 0))
lines(yc, bathy[30, ], lwd = 2, col = "red")
```

2.1. Another bathymetric view

```
par(mfrow = c(1, 1))
# make bathymetry that does not have NAs (for z-values)
D <- bathy ;  D[is.na(D)] <- 0
persp3D(x = xc, y = yc, z = -D, colvar = -bathy,
  col = "grey", shade = 0.5, scale = FALSE, phi = 80,
  NAcol = grey(0.2), theta = 0, resfac = 2, inttype = 2,
  box = FALSE)
```

try:

```
plotrgl()
plotdev(xlim = c(-100, 500), ylim = c(5667, 6300), zlim = c(-10,0), phi = 50)
```
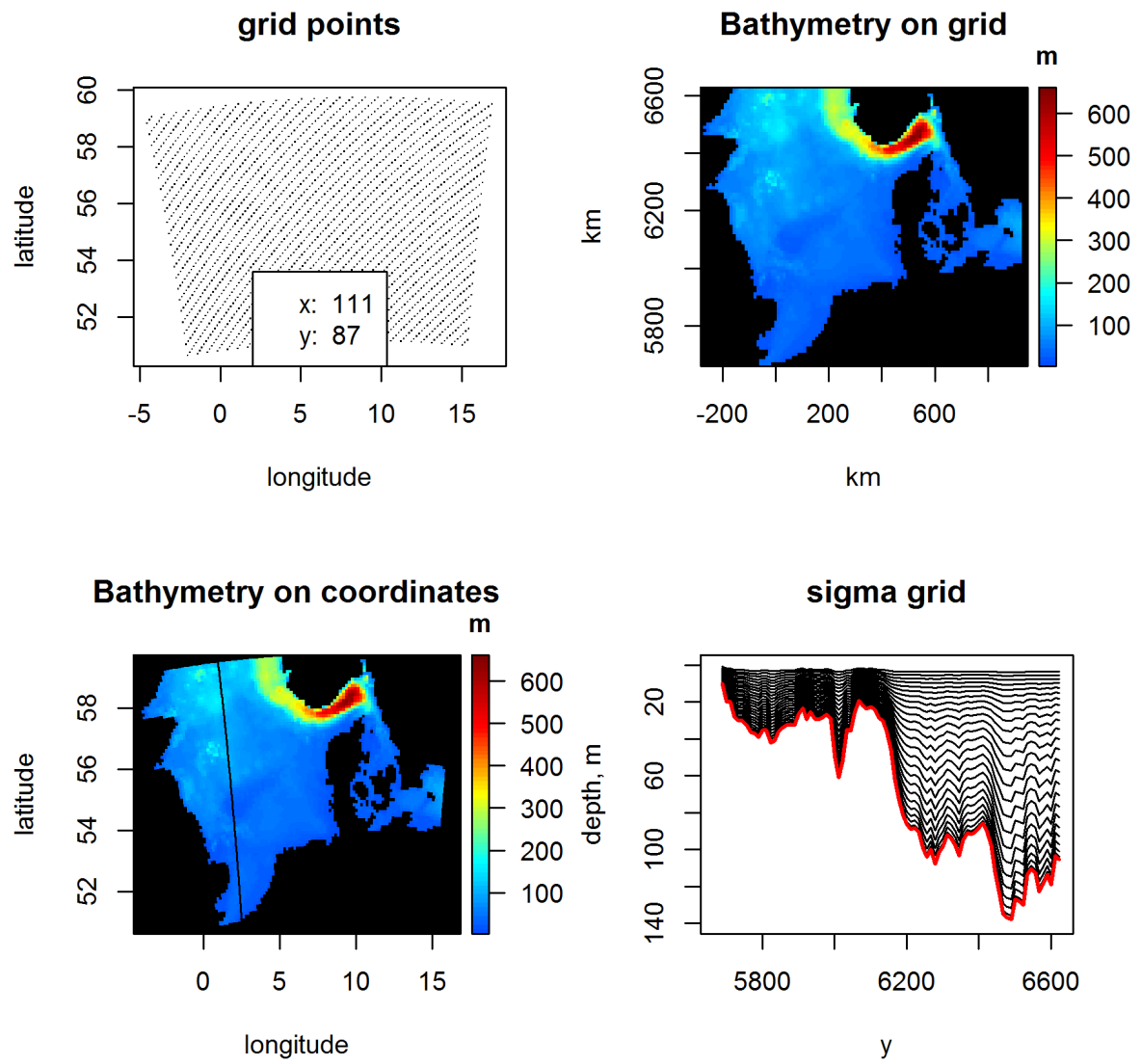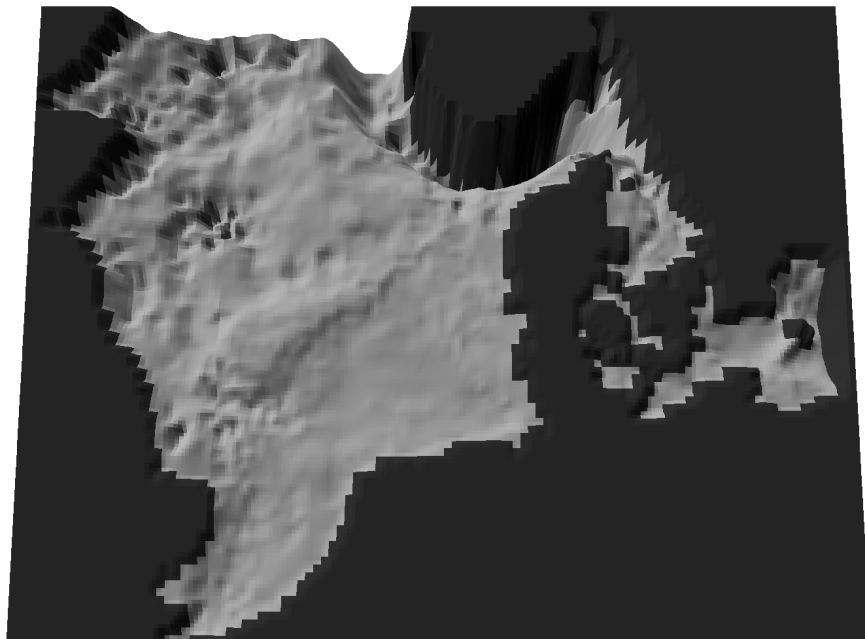
Figure 1: Bathymetry and grid
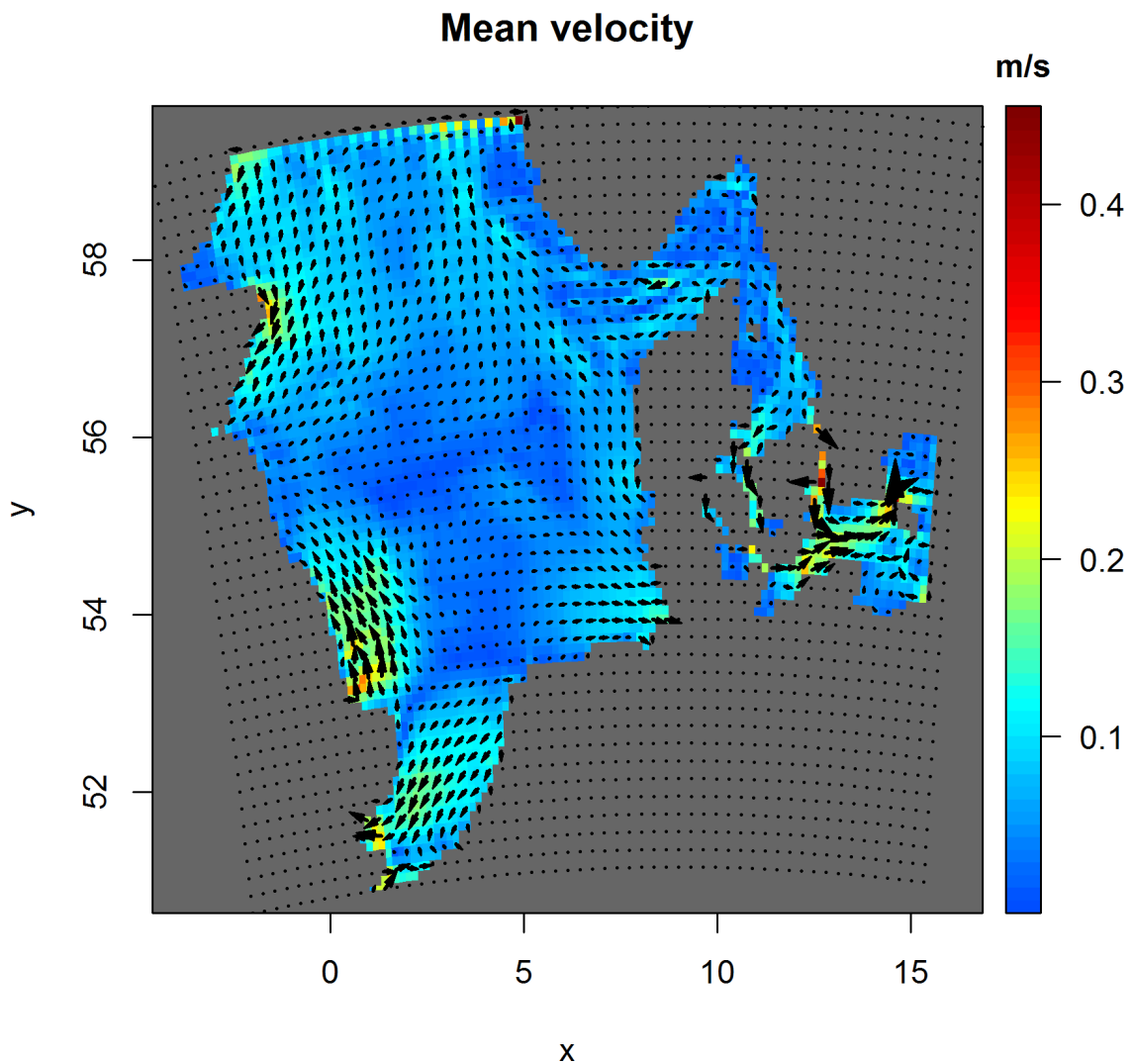
Figure 2: Bathymetry in perspective

3. integrated velocities

Calculate average velocity over the 30 days

```
U <- apply (u, FUN = mean, MARGIN = 1:2)
V <- apply (v, FUN = mean, MARGIN = 1:2)
meanV <- sqrt(U^2 + V^2)
```

plot average velocities

```
par(mfrow = c(1, 1))
image2D(z = meanV, x = lon_c, y = lat_c,
      col = col, NAcol = grey(0.4),
      main = "Mean velocity", clab = c("","","m/s"))
quiver(U, V, x = lon_c, y = lat_c, add = TRUE,
      by = 2, scale = 2, lwd = 2, arr.max = 0.4)
```

Mean velocity

Try this, and use left and right mousekey:

```
imagergl(z = meanV, x = lon_c, y = lat_c,
     col = col, NAcol = grey(0.4),
     main = "Mean velocity")
quiverrgl(U, V, x = lon_c, y = lat_c, add = TRUE,
     by = 2, scale = 2, lwd = 2, arr.max = 0.4)
# select a rectangular region
cutrgl()
```

Velocity (log scale) draped on the bathymetry.

```
persp3D(x = xc, y = yc, z = -D, colvar = meanV,
   lighting = TRUE, scale = FALSE, phi = 80, box = FALSE,
```

```
    log = "c", clab= "m/s", main = "Mean velocity",
  NAcol = grey(0.2), theta = 0, resfac = 2, inttype = 2)
```
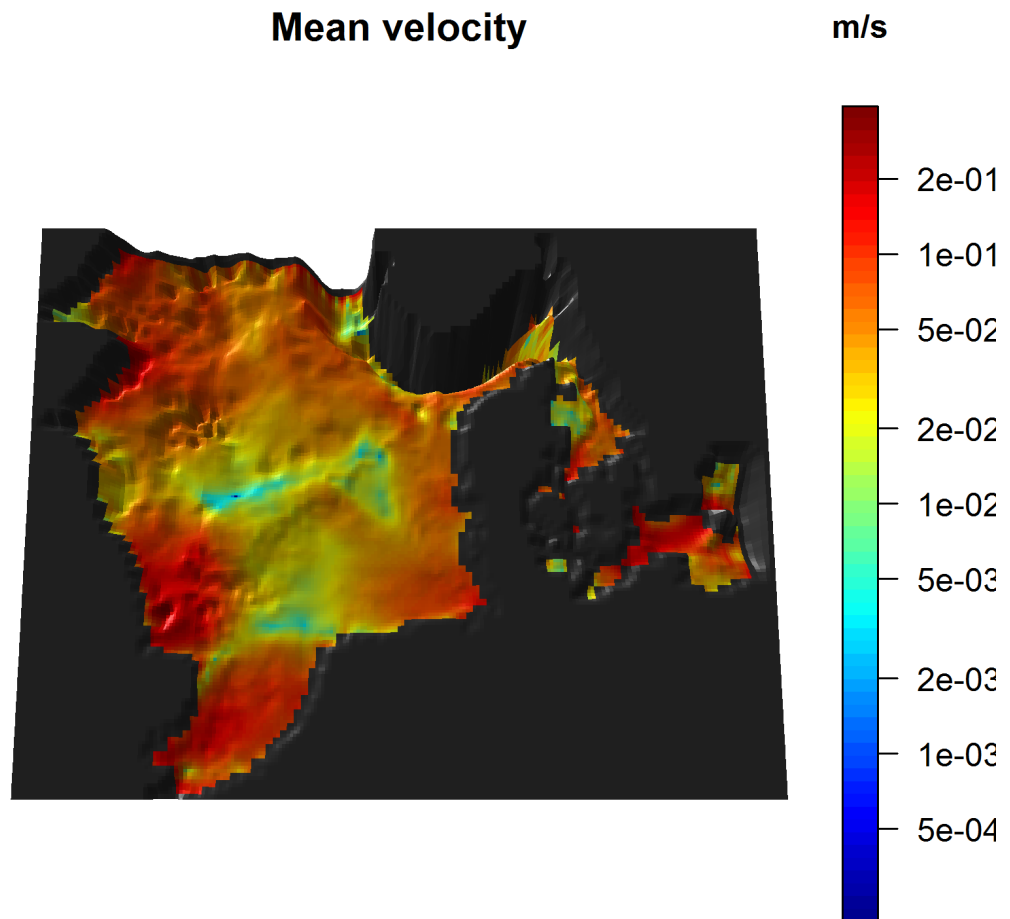

Try:

```
plotrgl()
```

4. Quivers at selected time points

```
tselect <- time[seq(from = 1, to = 30, by = 8)]
par(oma = c(0, 0, 2, 0), las = 1)
quiver(u, v, x = xc, y = yc,
       subset = (time %in% tselect),
       scale = 2, by = 3, mask = bathy,
       NAcol = "darkgrey", main = paste("day ", tselect/86400))
```
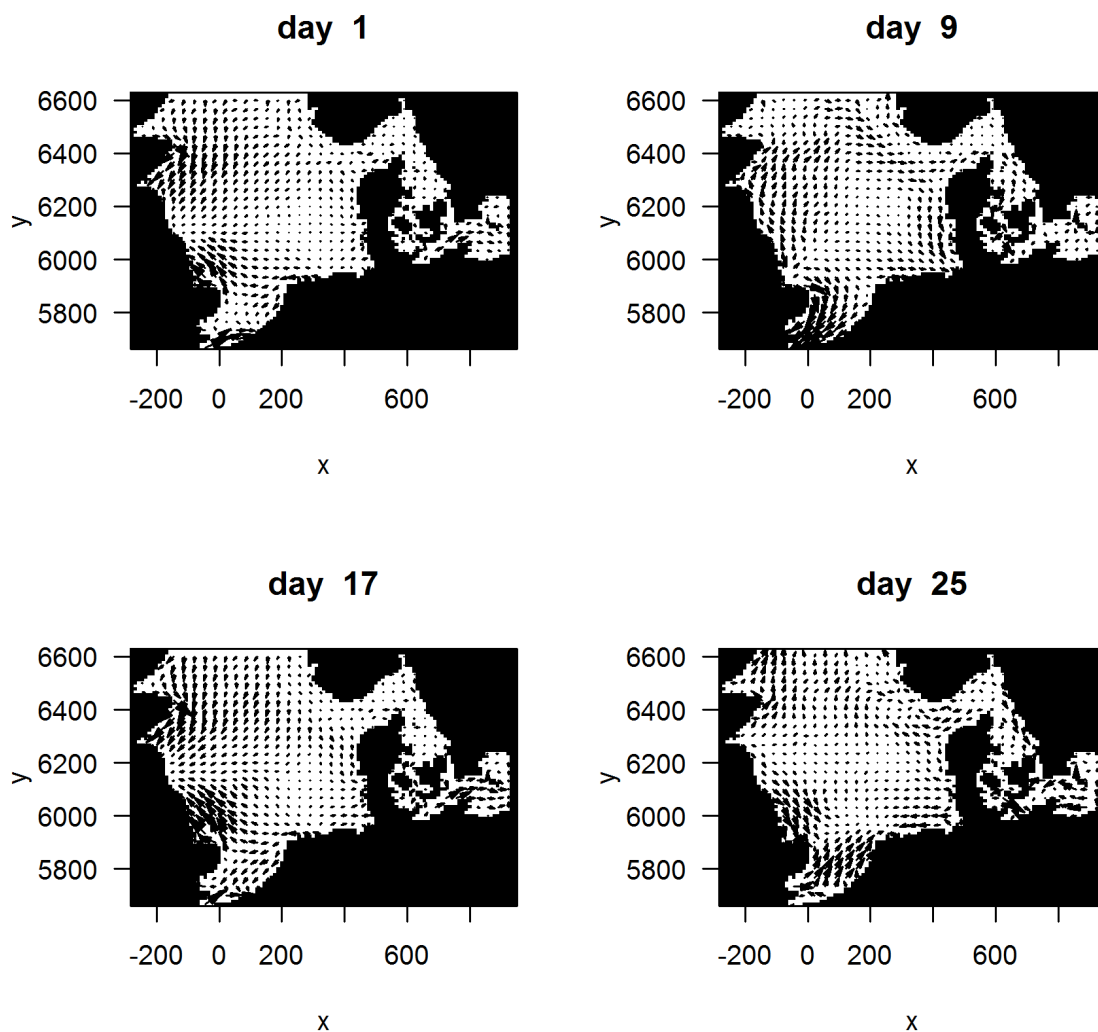
5. surface elevation, for each second time step

```
zlim <- range(elev, na.rm = TRUE)
tselect <- time[seq(from = 1, to = 30, by = 2)]
```

```
par(oma = c(0, 0, 2, 0))
pm <- par(mar = c(2, 2, 2, 0))
image2D(elev, subset = (time %in% tselect), mfrow = c(4,4), NAcol = "black",
       axes = FALSE, xlab = "", ylab = "", contour = TRUE,
       main = paste("day ", tselect/3600/24),
       zlim = zlim, colkey = FALSE)
colkey(clim = zlim, clab = "m")
mtext (outer = TRUE, side = 3, "SURFACE ELEVATION", line = 0)
par(mar = pm)
```
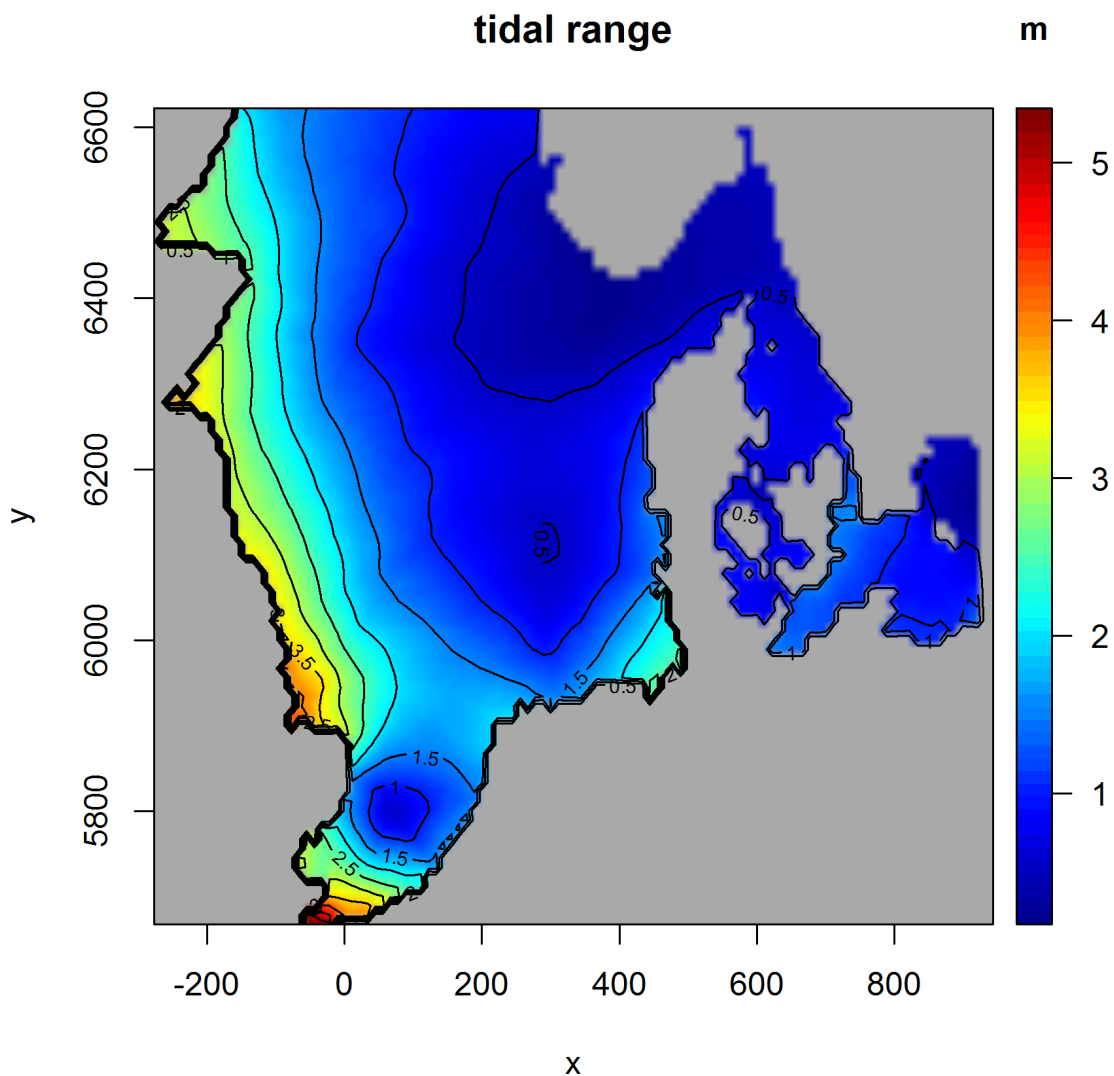

SURFACE ELEVATION
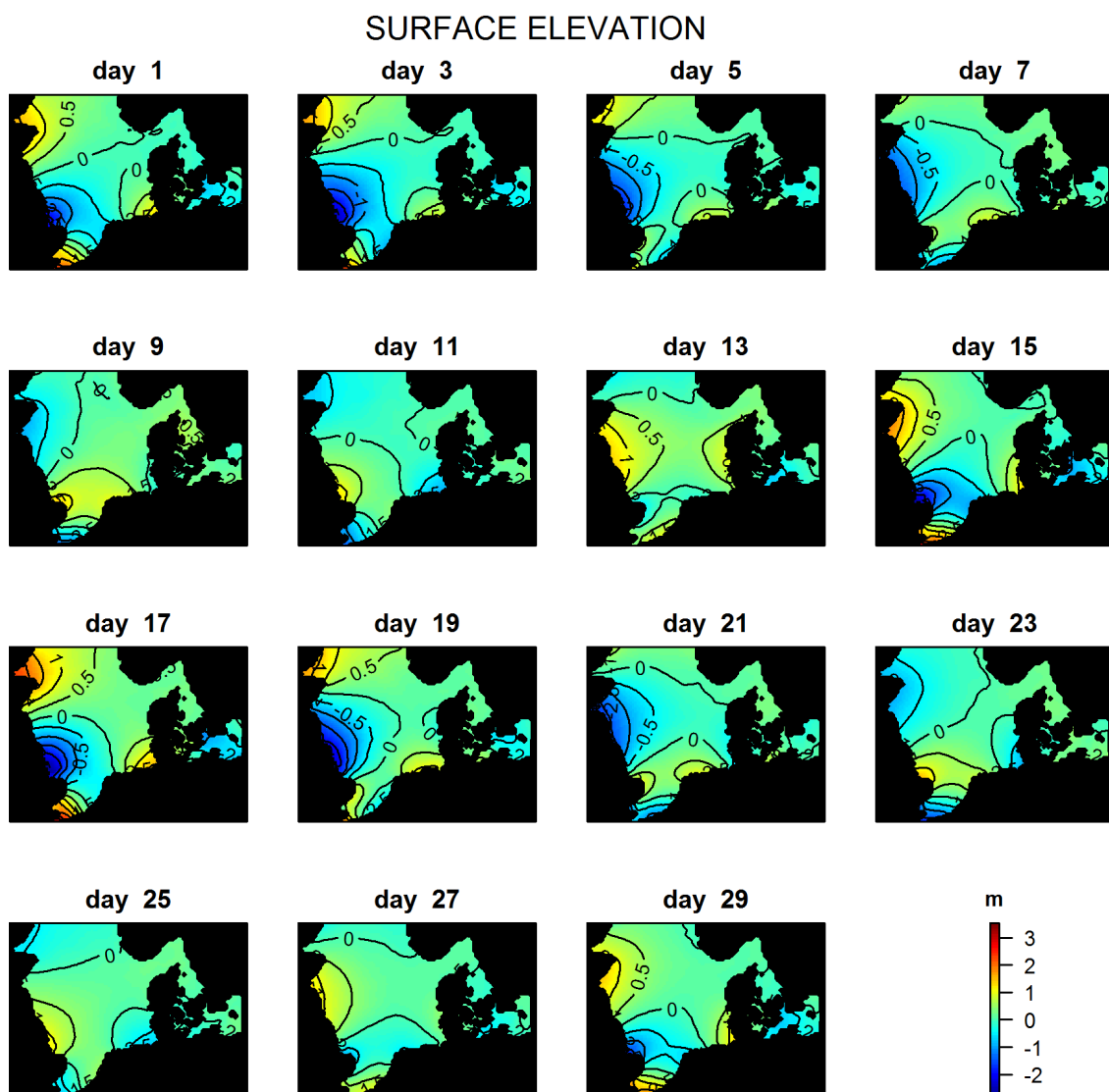
Tidal range

```
tidal.range <- apply(elev, MARGIN = 1:2,
                     FUN = function(x) diff(range(x)))
```

plot it, using `rasterImage` (smooth) and adding contours:

```
par(mfrow = c(1, 1))
image2D(x = xc, y = yc, z = tidal.range, contour = TRUE, rasterImage = TRUE,
   NAcol = "darkgrey", clab = "m", main = "tidal range")
```

6. surface temperature, for each second time step

```
surfTemp <- temp[ , , 26, ]
zlim <- range(surfTemp, na.rm = TRUE)
```

```
par(oma = c(0, 0, 2, 0))
pm <- par(mar = c(2, 2, 2, 0))
image2D(surfTemp, subset = (time %in% tselect), mfrow = c(4,4),
      NAcol = "darkgrey", axes = FALSE, xlab = "", ylab = "",
      main = paste("day ", tselect/3600/24), zlim = zlim, clab = "dg")
mtext (outer = TRUE, side = 3, "SURFACE TEMPERATURE")
par(mar = pm)
```


SURFACE TEMPERATURE
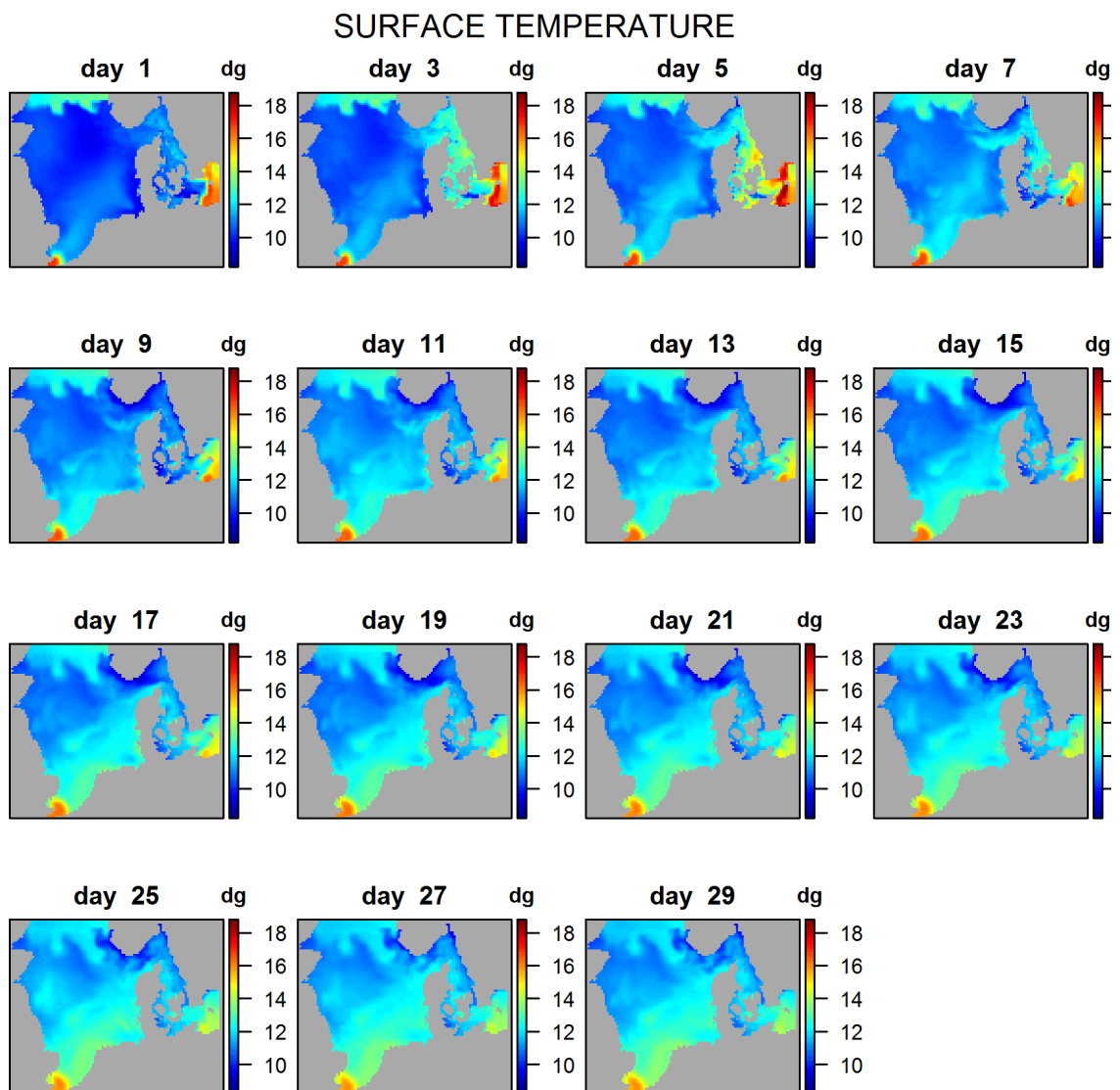
7. mean temperature along a NS-depth and a SW-depth transect

For two transects, first mean temperature is estimated (`apply`). The temperature matrix is then converted from sigma coordinates to depth values (`mapsigma`), increasing the resolution (`resfac`). Then the `image` is generated.

```
par(oma = c(0, 0, 2, 0))
par(mfrow = c(2, 1))
TranTemp <- apply(temp[30 , , , ], MARGIN = 1:2, FUN = mean)
MS <- mapsigma(TranTemp, sigma = depth[30, , ], x = yc,
  resfac = 3) # maps from sigma to z
image2D(MS$var, y = MS$depth, x = MS$x, ylim = rev(range(MS$depth)),
      NAcol = "black", ylab = "depth, m", xlab = "ypos, km",
      main = "NS transect at x = 30", clab = c("","dgC"))
TranTemp <- apply(temp[  , 68, , ], MARGIN = 1:2, FUN = mean)
MS       <- mapsigma(TranTemp, sigma = depth[ ,68, ], x = xc, resfac = 3)
image2D(MS$var, y = MS$depth, x = MS$x, ylim = rev(range(MS$depth)),
      NAcol = "black", ylab = "depth, m", xlab = "xpos, km",
      main = "EW transect at y = 68", clab = c("","dgC"))
mtext (outer = TRUE, side = 3, "Temperature transects")
```

Temperature transects
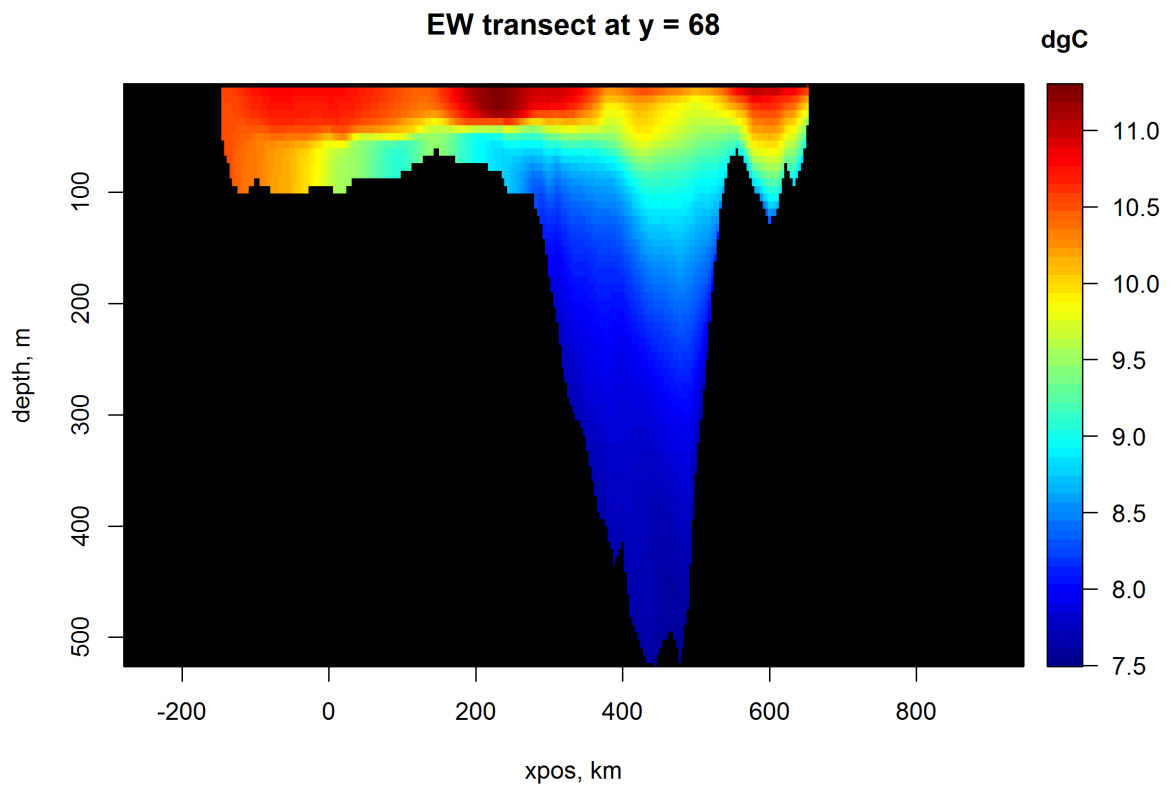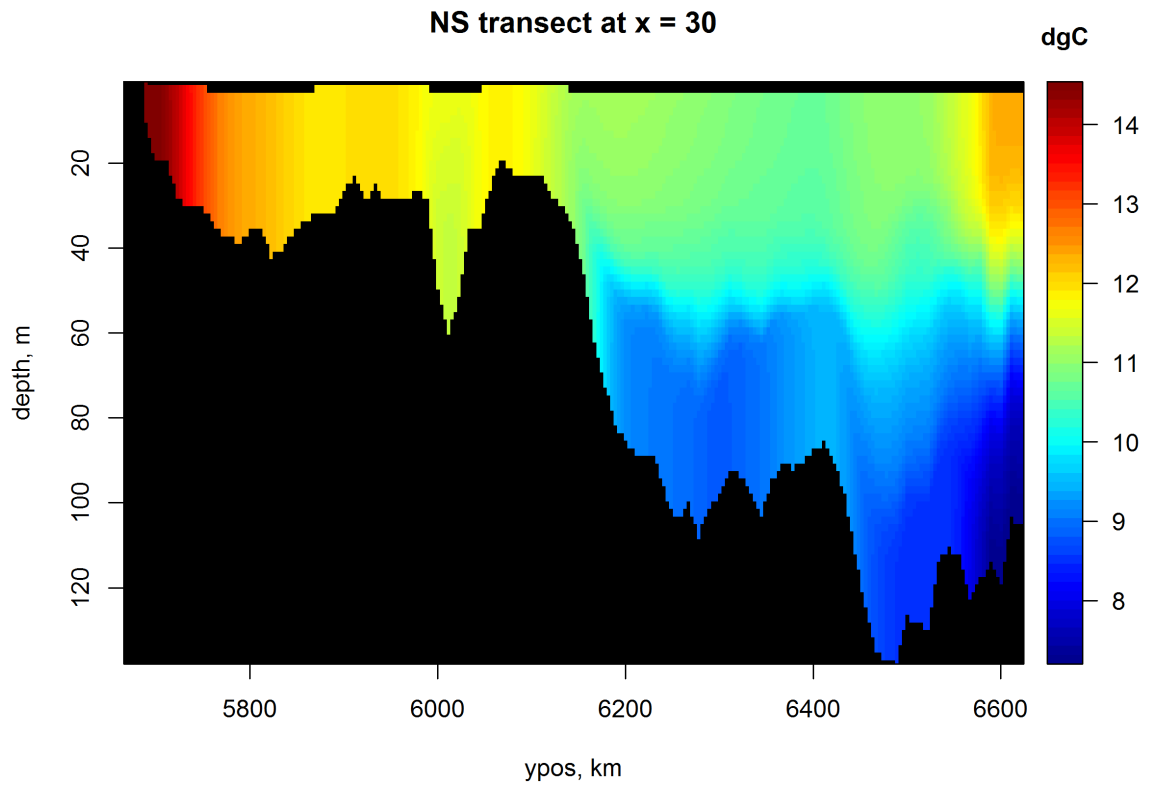
8. time-averaged vertical velocities at several depths

We create the temporally averaged vertical velocities, the selected depth levels and remove
the N and E points, as they are unstable

```
wmean <- apply(ww, MARGIN = 1:3, FUN = mean)
subset.level <- c(2, 5, 15, 25)
x.remove <- 105:111 ; y.remove <- 82:87
```

```
par(oma = c(0, 0, 2, 0))
image2D(wmean[-x.remove, -y.remove, ], NAcol = "black",
        main = paste("depth level", subset.level), zlim = c(-1e-3, 4e-4),
        subset = (level %in% subset.level), clab = c("w","m/s"))
mtext (outer = TRUE, side = 3, "Vertical velocity")
```
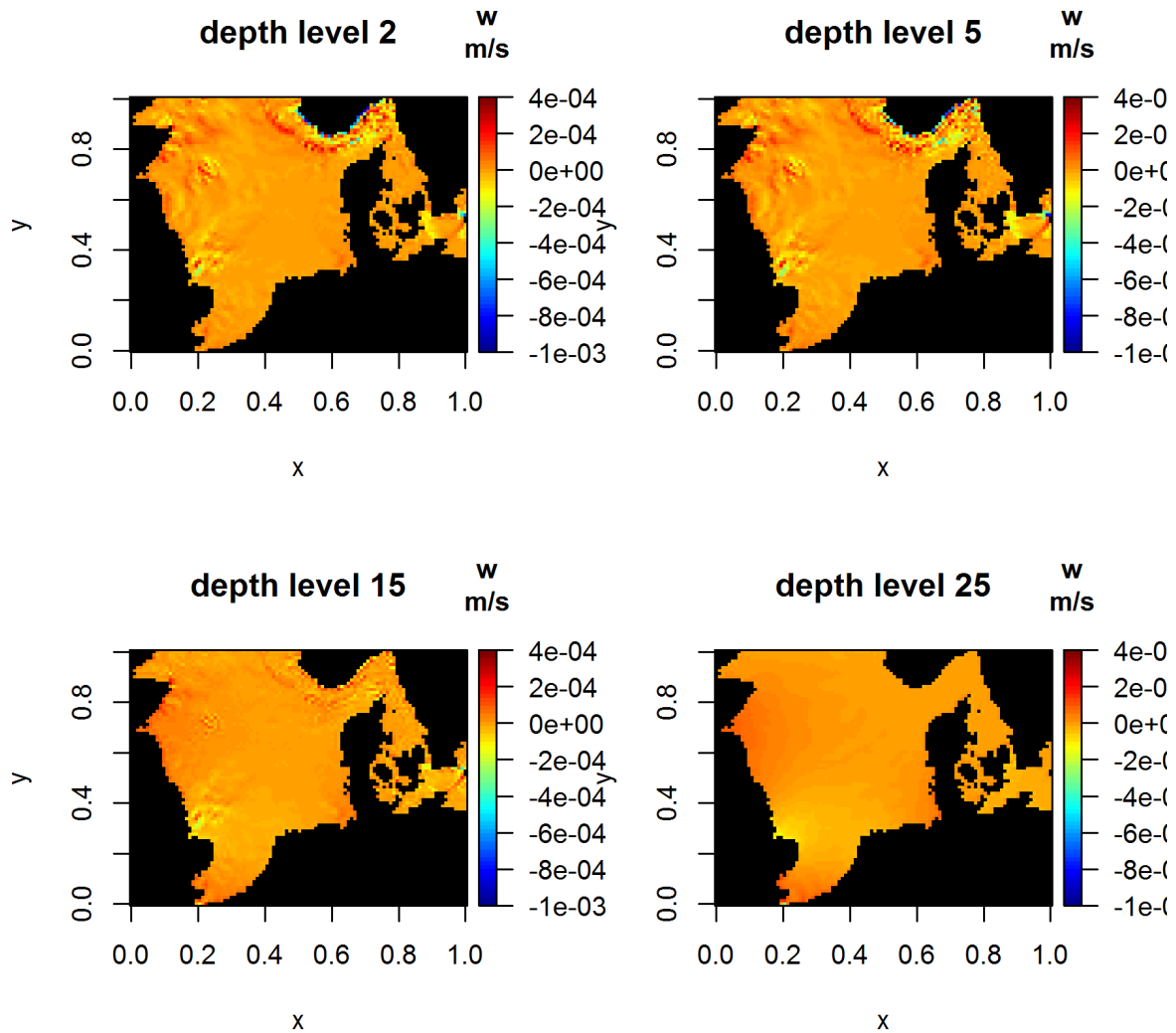
# 9. time-averaged horizontal velocity vectors at several depths

temporally averaged horizontal velocities

```
umean <- apply(uu, MARGIN = 1:3, FUN = mean)
vmean <- apply(vv, MARGIN = 1:3, FUN = mean)
```

The N and E points are removed

```
par(oma = c(0, 0, 2, 0))
quiver(umean[-x.remove, -y.remove, ], vmean[-x.remove, -y.remove, ],
       NAcol = "grey", mask = bathy[-x.remove, -y.remove],
       main = paste("depth level", subset.level), by = 3,
        subset = (level %in% subset.level))
mtext (outer = TRUE, side = 3, "Time-average flows")
```

Time-average flows

depth level 2

depth level 5

depth level 15

depth level 25
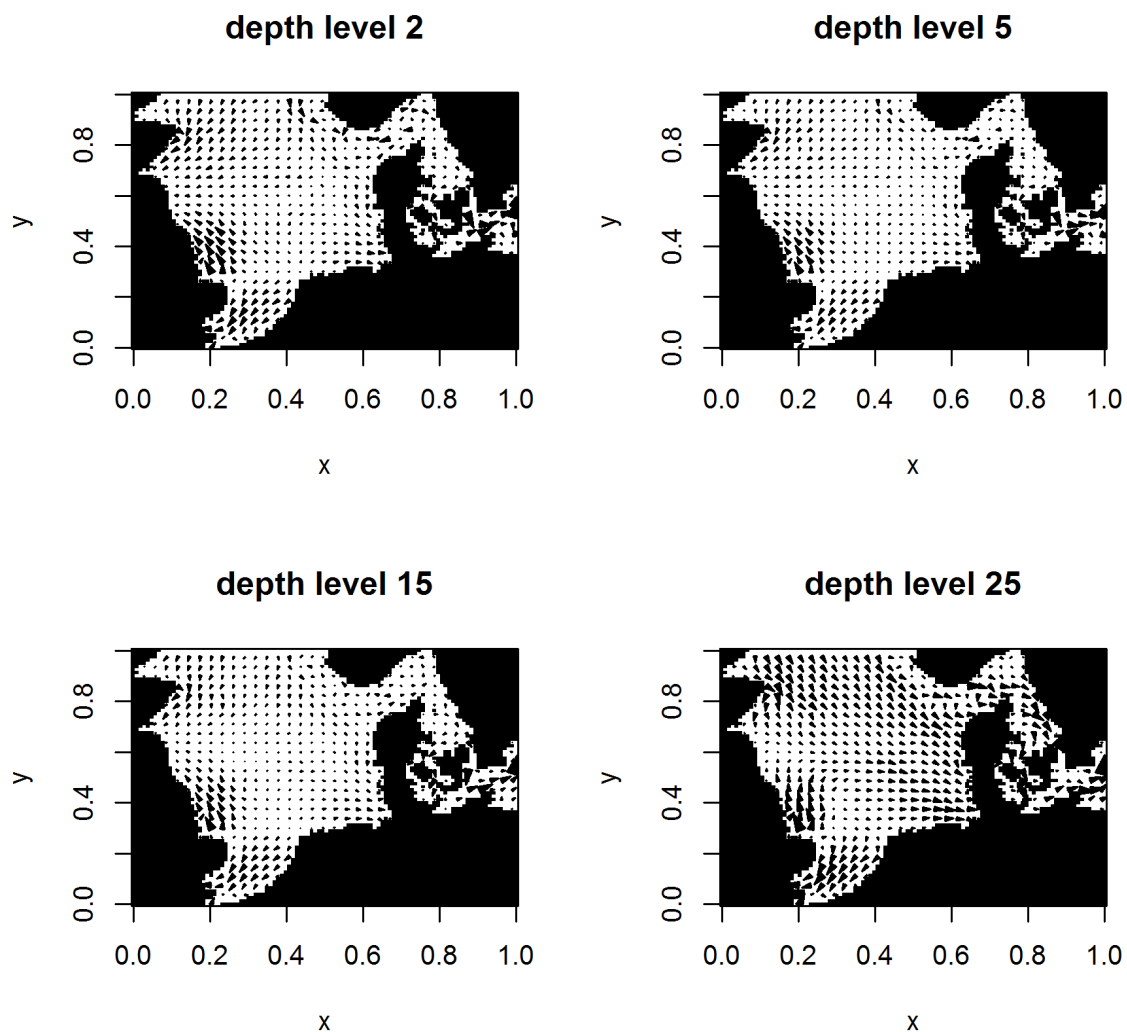
10. NE and SW transects of time-averaged flows

The vertical velocities are 1000 times smaller than the horizontal ones; to produce the quiver plots, they are multiplied with 1000

```
par(mfrow = c(2, 1))
# NS transect
Tran.u <- mapsigma(umean[30 , -y.remove, ], sigma = depth[30, -y.remove,])
Tran.w <- mapsigma(wmean[30 , -y.remove, ], sigma = depth[30, -y.remove,])
TotVelo <- sqrt(Tran.u$var^2 + Tran.w$var^2 )
DD <- Tran.w$depth
# ranges of u and v
range(Tran.u$var, na.rm = TRUE)
```

```
[1] -0.09098173  0.12890817
```

```
range(Tran.w$var, na.rm = TRUE)
```

```
[1] -7.923670e-05  7.631142e-05
```

```
#
image2D(TotVelo, y = DD, x = yc[ -y.remove], col = col,
      ylim = rev(range(DD)), NAcol = "black", resfac = 2,
      ylab = "depth, m", xlab = "ypos, km",
      main = "NS transect at x = 30")
quiver(Tran.u$var, Tran.w$var*1000, arr.min = 0.1, arr.max = 0.1,
      y = DD, x = yc[ -y.remove],
      by = c(2,1), scale = 2, add = TRUE)
#
# EW transect
Tran.v <- mapsigma(vmean[-x.remove , 68, ], sigma = depth[-x.remove, 68,])
Tran.w <- mapsigma(wmean[-x.remove , 68, ], sigma = depth[-x.remove, 68,])
TotVelo <- sqrt(Tran.v$var^2 + Tran.w$var^2 + 1e-8)
DD <- Tran.w$depth
image2D(TotVelo, y = DD, x = xc[ -x.remove], col = col,
      ylim = rev(range(DD)), NAcol = "black", resfac = 2,
      ylab = "depth, m", xlab = "xpos, km",
      main = "EW transect at y = 68")
quiver(Tran.v$var, Tran.w$var*1000, arr.min = 0.1, arr.max = 0.1,
      y = DD, x = xc[ -x.remove],
      by = c(2, 1), scale = 2, add = TRUE)
mtext (outer = TRUE, side = 3, "Flow transects")
```

NS transect at x = 30
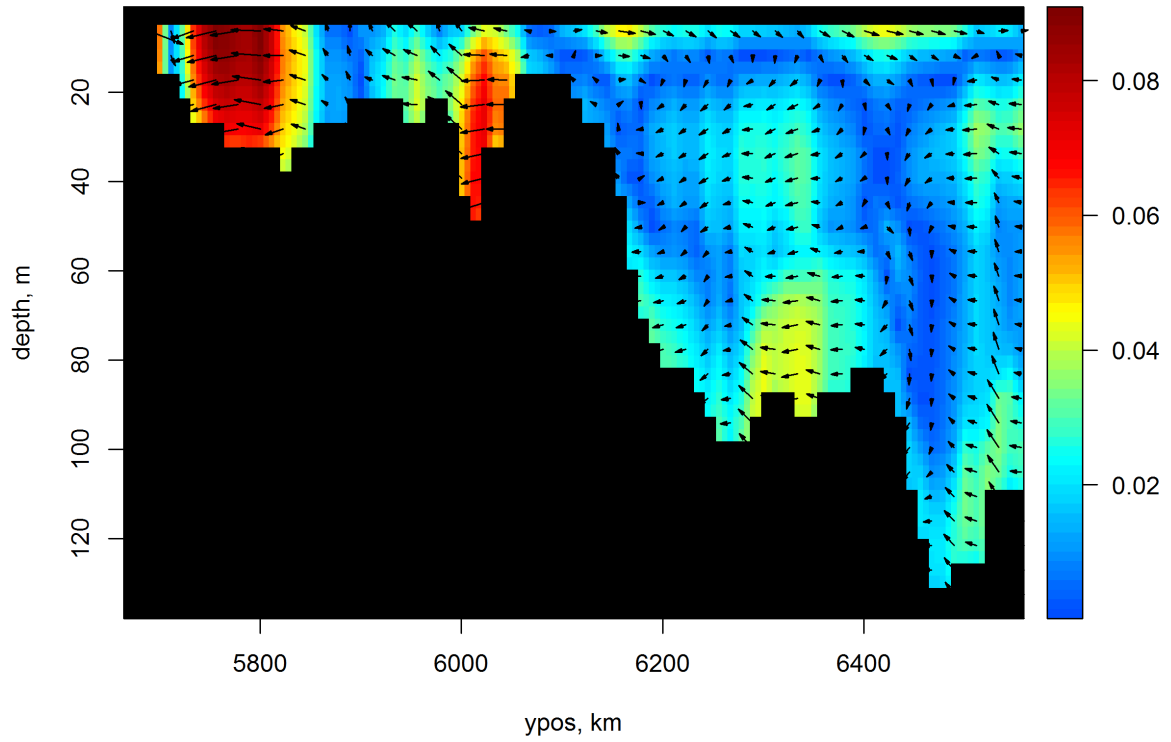
EW transect at y = 68
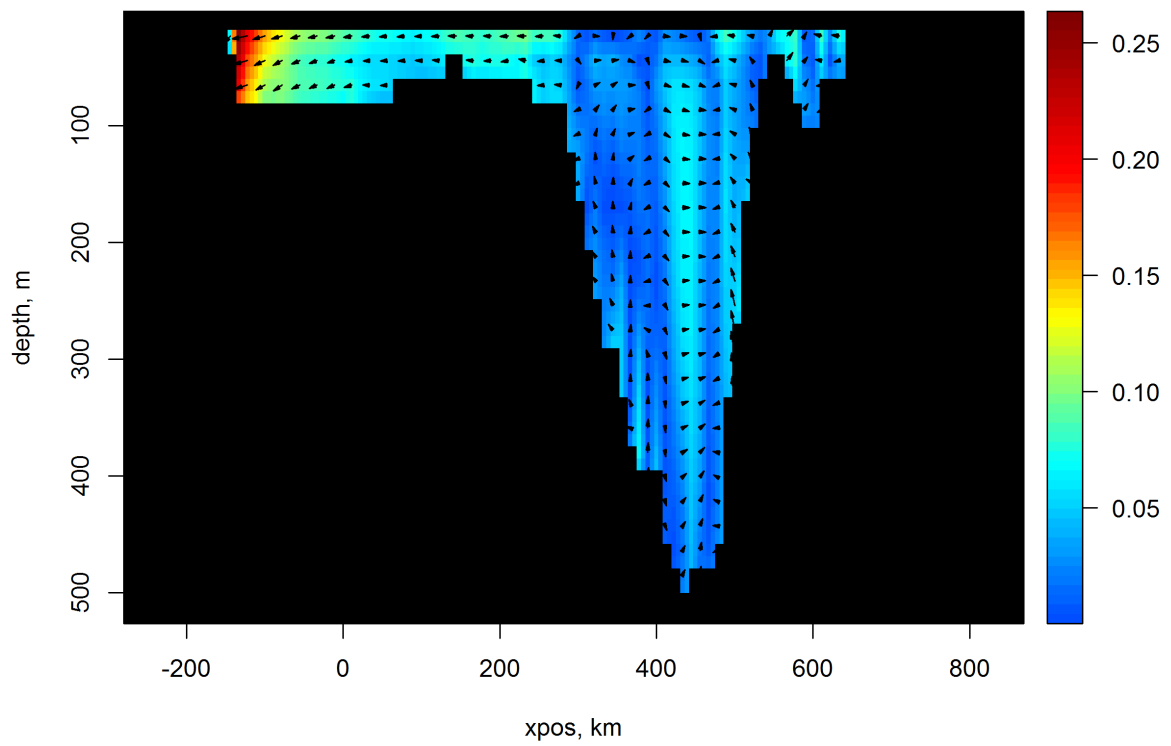
11. Irregular transects, e.g. ship track

```
# assume a ship track:
track <- cbind(seq(400, 600, length.out = 50),
               seq(6400, 6500, length.out = 50))
# Draw the bathymetry on km grid
par(mfrow = c(2, 2))
image2D(bathy, x = xc, y = yc, NAcol = "black",
      xlim = c(200, 700), ylim = c(6100, 6600),
      col = col, xlab = "km", ylab = "km",
      clab = c("","","m"), main = "cruise track")
# add track
points(track, pch = ".", cex=3)
# Depth on track
trackBat <- extract(var = bathy, x = xc, y = yc, xyto = track)
plot(x = trackBat$xy[,1], y =-trackBat$var, type = "l", xlab = "xc",
  ylab = "m", main = "bathymetry on track")
# temperature
Avgtemp <- apply(temp, MARGIN = 1:3, FUN = mean, na.rm = TRUE)
tempSigma <- transectsigma(var = Avgtemp, sigma = depth,
  x = as.vector(xc), y = as.vector(yc), to = track, resfac = 5)
image2D(tempSigma$var, x = tempSigma$x, y = tempSigma$depth,
  ylim = rev(range(tempSigma$depth)), NAcol = "black",
  main = "temperature on track")
# vertical velocity
wSigma <- transectsigma(var = wmean, sigma = depth,
  x = as.vector(xc), y = as.vector(yc), to = track, resfac = 5)
image2D(wSigma$var, x = wSigma$x, y = wSigma$depth,
  ylim = rev(range(tempSigma$depth)), NAcol = "black",
  main = "vertical velocity on track")
```
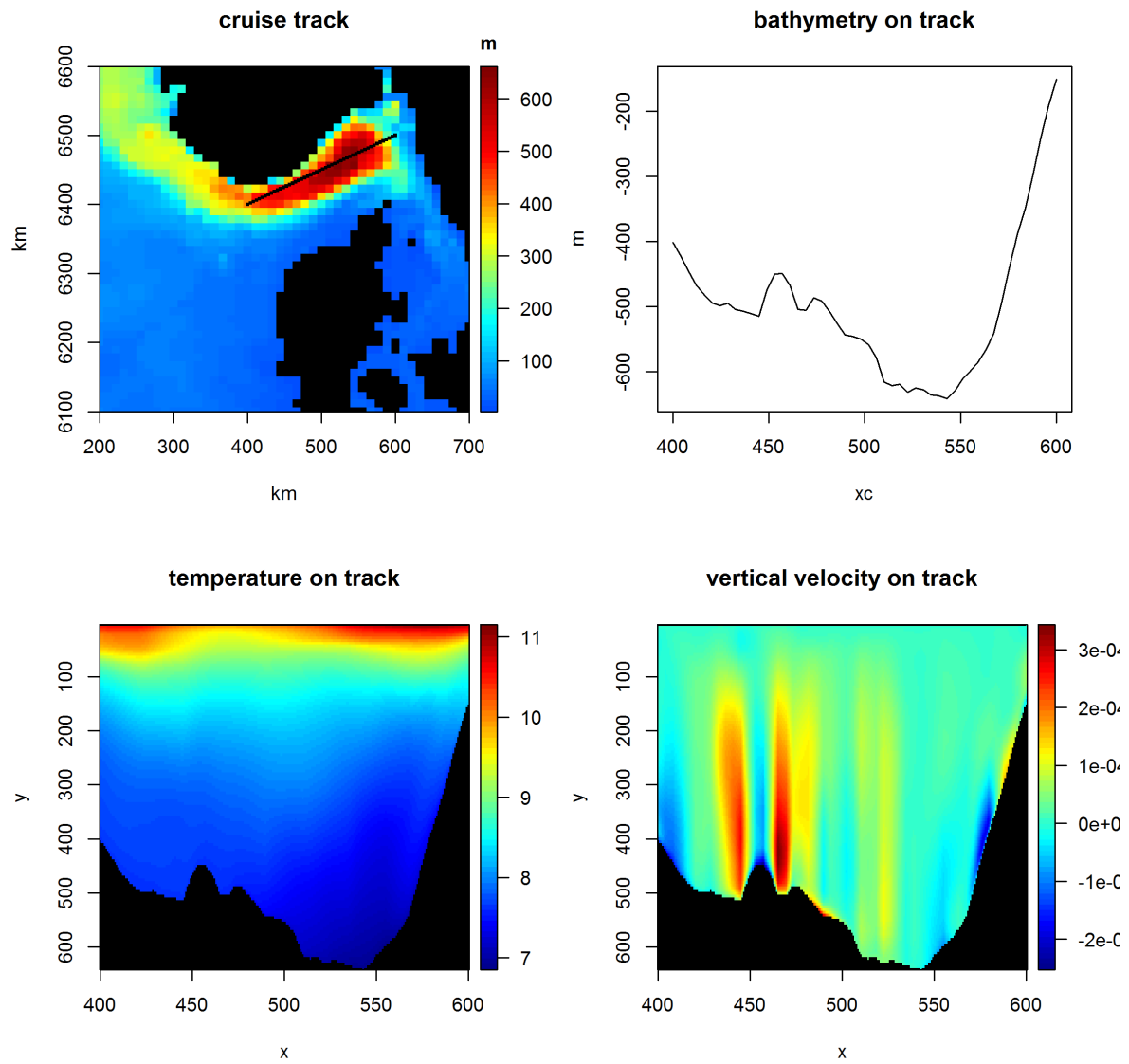
12. Finally

This vignette was made with Sweave (Leisch 2002).

References

Leisch F (2002). "Sweave: Dynamic Generation of Statistical Reports Using Literate Data Analysis." In W Härdle, B Rönz (eds.), "Compstat 2002 - Proceedings in Computational Statistics," pp. 575–580. Physica Verlag, Heidelberg. ISBN 3-7908-1517-9, URL http://www.stat.uni-muenchen.de/~leisch/Sweave.

Michna P (2011). *RNetCDF: R Interface to NetCDF Datasets.* R package version 1.5.2-2, URL http://CRAN.R-project.org/package=RNetCDF.

Soetaert K (in prep). *OceanView: Visualisation of Ocean Data and Model Output.* R package version 1.0.

Affiliation:

Karline Soetaert
Royal Netherlands Institute of Sea Research (NIOZ)
4401 NT Yerseke, Netherlands
E-mail: karline.soetaert@nioz.nl
URL: http://http://www.nioz.nl/