Supplemental Information: Ekman Mass Transport and Pumping Calculations

This report shows the Ekman Mass Transport (EMT) upwelling calculations and the Ekman Pumping calculations. The calculations are compared to values provided by the Environmental Research Division (ERD) at the Southwest Fisheries Science Center. Although we do not use the ERD upwelling indices in the paper, they are shown here to illustrate why we are not using them and to confirm that our EMT calculations are correct.

The second part of the report compares four different wind-products. The ERD products use pressure to estimate winds and this is known to be inaccurate near the equator. Instead, researchers studying winds off the coast of India use scatterometer instruments (ASCAT and QSCAT). In our paper, we use winds from the ERA5 Reanalysis product, which uses ASCAT and QSCAT data. This allows us to use one product for the entire time-range of our catch data. The wind and derived Ekman Mass Transport and Pumping values are compared using ASCAT data directly versus the ERA5 product.

Hersbach, H, Bell, B, Berrisford, P, et al. The ERA5 global reanalysis. Q J R Meteorol Soc. 2020; 146: 1999–2049. https://doi.org/10.1002/qj.3803

Load R libraries

```
library(ggplot2)
library(tidyr)
library(dplyr)
library(raster)
library(rasterVis)
```

Get the data

Get pressure, wind, and Ekman Transport from ERDDAP servers. This gets the data for the current month and has all the needed parameters. Current FNMOC grid resolution is 1 degree while older resolution was 2.5 degree and later interpolated to 1 degree.

Get data for latitude degrees 7 to 15N and longitude degrees 70 to 78E from https://coastwatch.pfeg.noaa.gov/erddap/griddap/erdlasFnTransMon. The `getdata()` function is at the end of this report. The downloaded data file `erdlasFnTransMon-7-15-70-78-2020-11-01-2020-11-02.csv` is also at the end of the report.

```
dates <- c("2020-11-01T00:00:00Z", "2020-11-02T06:00:00Z")
dat <- getdata("erdlasFnTransMon", date=dates)
```

```
## data read from erdlasFnTransMon-7-15-70-78-2020-11-01-2020-11-02.csv
## data erdlasFnTransMon date 2020-11-01T00:00:00Z-2020-11-02T06:00:00Z, latitude 7-15, longitude 70-78
```

```r
cat(colnames(dat))
```

```
## time latitude longitude P_msl u v uv_mag taux tauy curl ektrx ektry
```

Calculations of wind from FNMOC pressure grid

Schwing, F. B., O'Farrell, M., Steger, J., and Baltz, K., 1996: Coastal Upwelling Indices, West Coast of North America 1946 - 1995, NOAA Technical Memorandum NMFS-SWFSC-231

Schwing et al 1996 gives the calculations for computing the wind vector from the pressures reported by FNMOC. First the pressure differential in the y (north-south) and x (east-west) directions is computed. y is the latitude center of the box. x is the longitude center of the box. P is in Pascals. Note the the pressure reported by FNMOC must be multiplied by 100 to get Pascals. h is the resolution of the grid in radians ($= \pi \times 1$ degree / 180 degree). Equation 1 in Schwing et al 1996:

$$\frac{\Delta P}{\Delta x} = (P_{x+h,y} - P_{x-h,y})/2h \qquad \frac{\Delta P}{\Delta y} = (P_{x,y+h} - P_{x,y-h})/2h \tag{1}$$

The east ($\vec{u}_g$) and north ($\vec{v}_g$) components of the geostrophic wind are (Equation 2 in Schwing et al 1996):

$$\vec{u}_g = -\frac{1}{f\, p_a\, R} \frac{\Delta P}{\Delta y} \tag{2}$$

$$\vec{v}_g = \frac{1}{f\, p_a\, R\, cos(\pi y/180)} \frac{\Delta P}{\Delta x} \tag{3}$$

In the equation, $p_a = 1.22\, kg/m^3$ and R is the radius of the earth in meters $= 6371000\, m$. f is the Coriolis parameters and is $2\Omega sin(\pi y/180)$, where $\Omega - 7.272205 \times 10^{-5}$ and y is the latitude of the center of the grid cell in degrees (and $\pi y/180$ is the latitude in radians). Positive $\vec{u}$ is wind blowing west to east. Positive $\vec{v}$ is wind blowing south to north (in the northern hemisphere). Notice the the east-west geostrophic wind is associated with the north-south pressure differential while the north-south wind is associated with the east-west differential.

On the last line of page 7 in Schwing et al 1996, they state "To approximate frictional effects, the geostrophic wind at the sea surface is estimated by rotating the geostrophic wind 15 deg to the left and reducing its magnitude by 30%". The actual angle that the wind is rotated in the numbers provided by ERD however appears to be 30 degrees. The following equation rotates and reduces the magnitude. α is the angle in radians ($\pi \times 30/180$).

$$\vec{u} = 0.7(cos(\alpha)\, u_g - sin(\alpha)\, v_g) \tag{4}$$

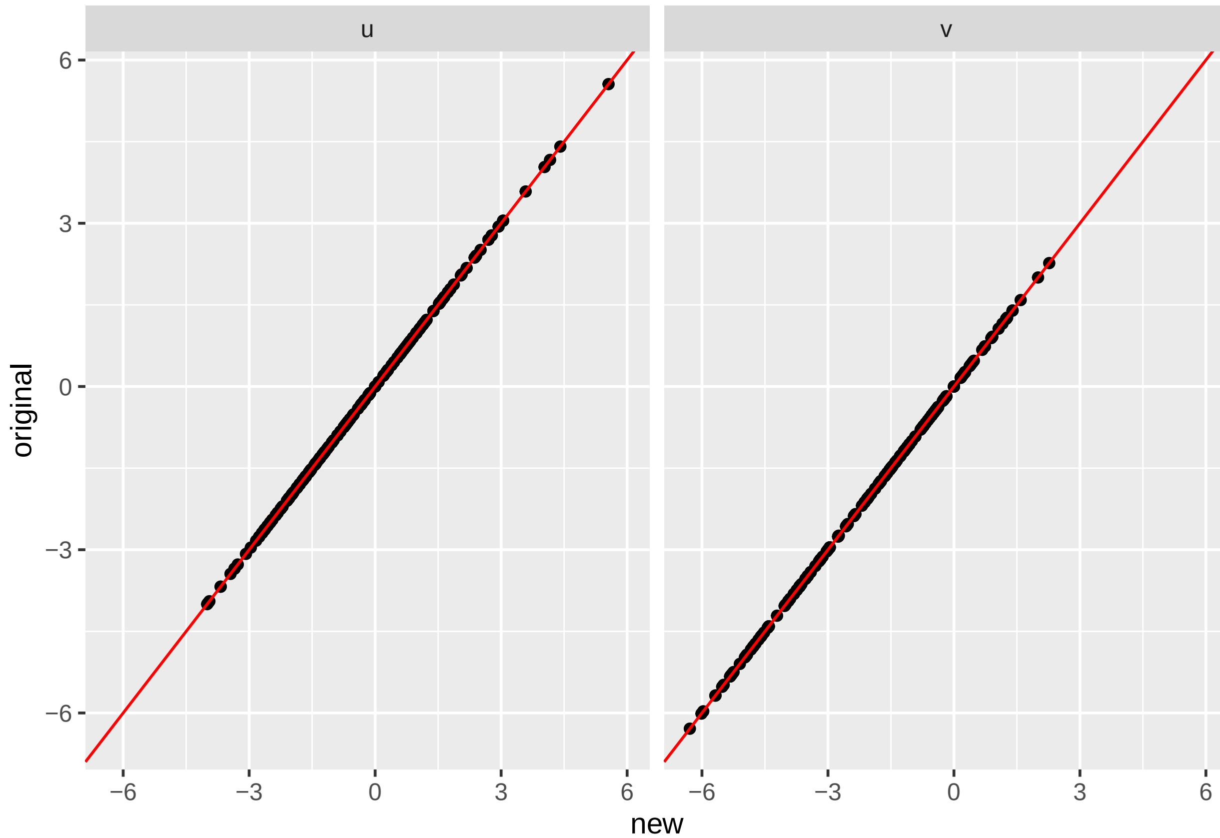
$$\vec{v} = 0.7(sin(\alpha)\, u_g + cos(\alpha)\, v_g) \tag{5}$$

The `getwindfromP()` function at the end of this report shows the R code for these calculations.

```r
wind1 <- getwindfromP(dat)
```

Compare to wind numbers reported by ERD

Show that this is the same as what is reported by SWFSC-ERD.



Calculations of Ekman Mass Transport and Pumping

The following equations are in

Schwing, F. B., O'Farrell, M., Steger, J., and Baltz, K., 1996: Coastal Upwelling Indices, West Coast of North America 1946 - 1995, NOAA Technical Memorandum NMFS-SWFSC-231

The same equations and the non-linear C_d discussed below are also given in

Shafeeque et al. 2019: Effect of precipitation on chlorophyll-a in an upwelling dominated region along the West coast of India. Journal of Coastal Research, 86, 218-224. https://doi.org/10.2112/SI86-032.1

Wind stress is computed as

$$\vec{\tau} = p_a \, C_d \, |\vec{w}| \, \vec{w} \tag{6}$$

$$\tau_x = p_a \, C_d \, |\vec{w}| \, \vec{u} \qquad \tau_y = p_a \, C_d \, w \, \vec{v} \tag{7}$$

where C_d is the coefficient of drag and $|\vec{w}|$ is the wind speed (in m/s) and $\vec{w}$ is the wind vector. C_d is the empirical drag coefficient. It is a non-linear drag coefficient based on Large and Pond (1981) and modified for low wind speeds as in Trenberth et al. (1990), per discussion on the ERD upwelling page. This is

$$C_d = \begin{cases} 2.18 \times 10^{-}3, & \text{if } |\vec{w}| < 1 \\ (0.62 + \frac{1.56}{|\vec{w}|}) \times 10^{-}3 & \text{if } 1 < |\vec{w}| < 3 \\ 1.14 \times 10^{-}3 & \text{if } 3 \geq |\vec{w}| < 10 \\ (0.49 + 0.065|\vec{w}|) \times 10^{-}3 & \text{if } 10 \geq |\vec{w}| \end{cases} \tag{8}$$

Ekman Mass Transport (EMT kg/m s) is perpendicular (rotated 90 degrees clockwise) to wind stress. The EMT in the x and y directions is (Equation 4 in Schwing et al 1996):

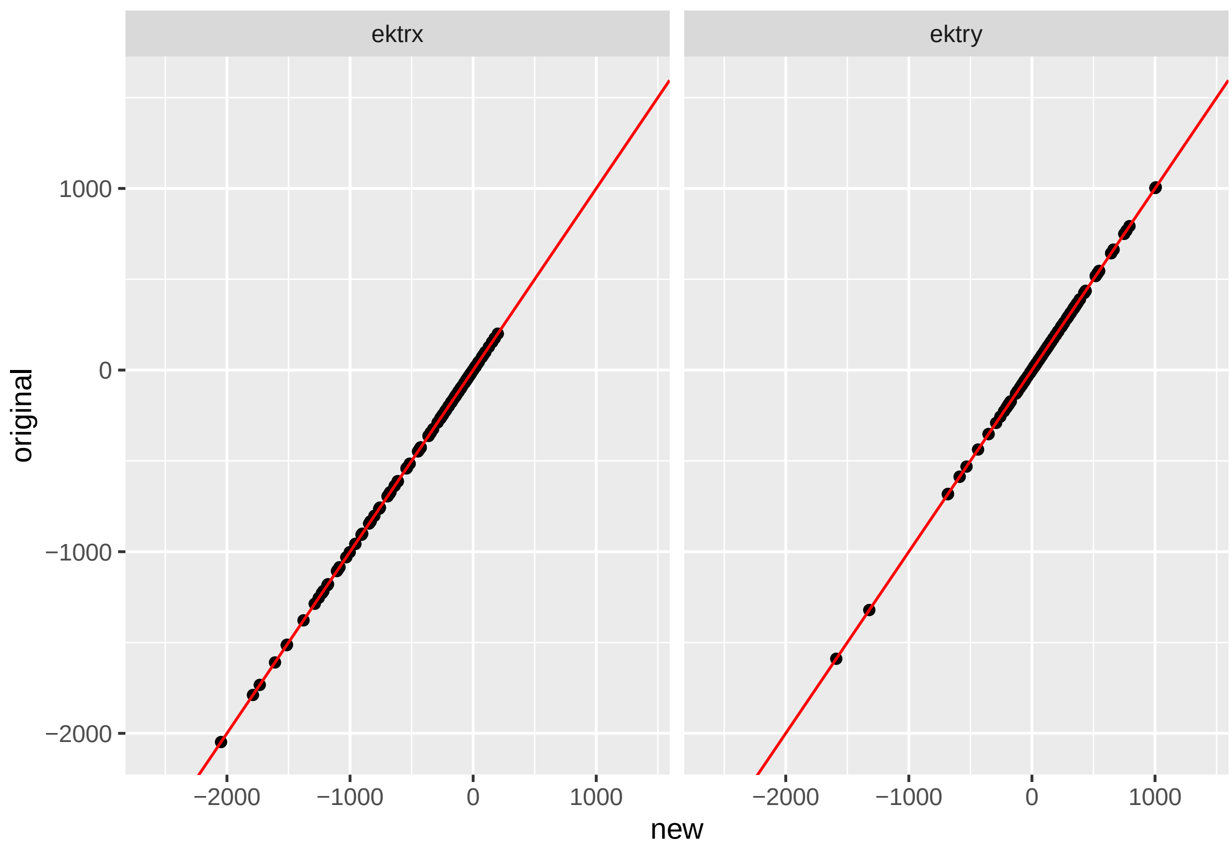
$$EMT_y = -\tau_x/f \qquad EMT_x = \tau_y/f \tag{9}$$

The `getEMT()` and `Cd()` functions shows how to compute these. `coast_angle` is added here for the upwelling index discussed below.

Compare EMT computed from wind to that reported by ERD

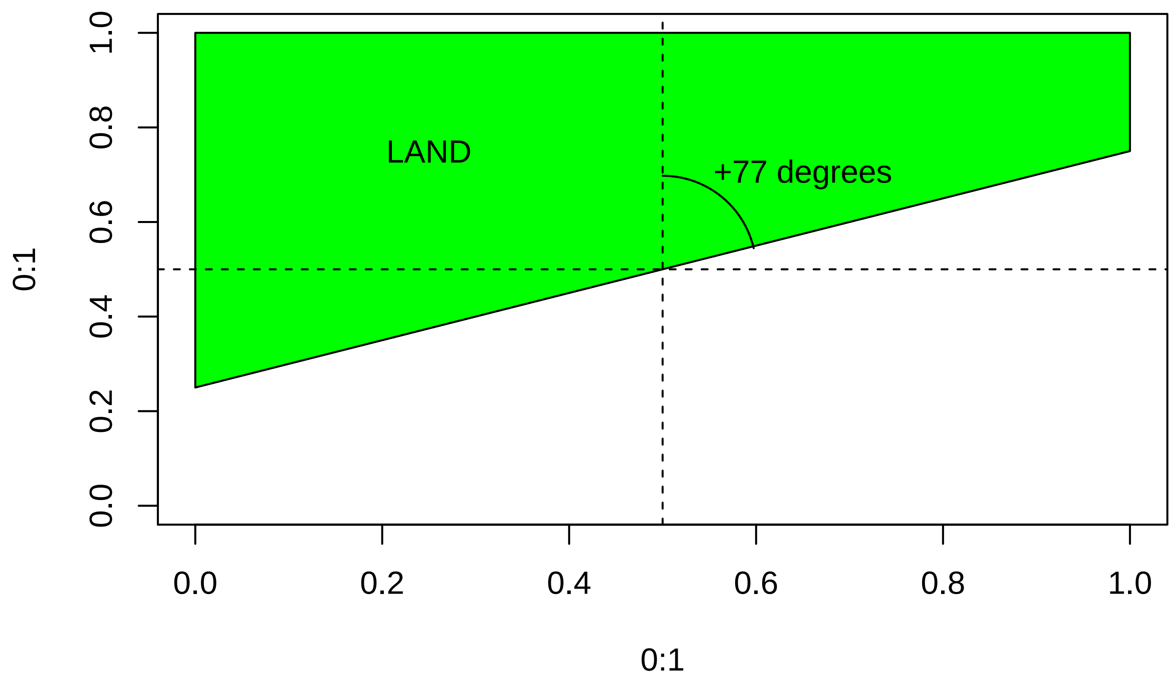
The values are the same, confirming that the equations are correct.

```
wind1 <- getEMT(wind1, coast_angle=158)
```



Upwelling index

ERD's Bakun upwelling index is derived from the EMT that is perpendicular and away from the coast. Their upwelling index is computed by finding the EMT vector that is perpendicular to the coast and defining a positive vector as away from the coast and a negative vector as toward the coast. ERD then divides this value by 10. The coast angle is defined as the degrees rotation clockwise away from north-south with land to the west (see figure). The coast angle for the SW coast of India is 158 degrees.



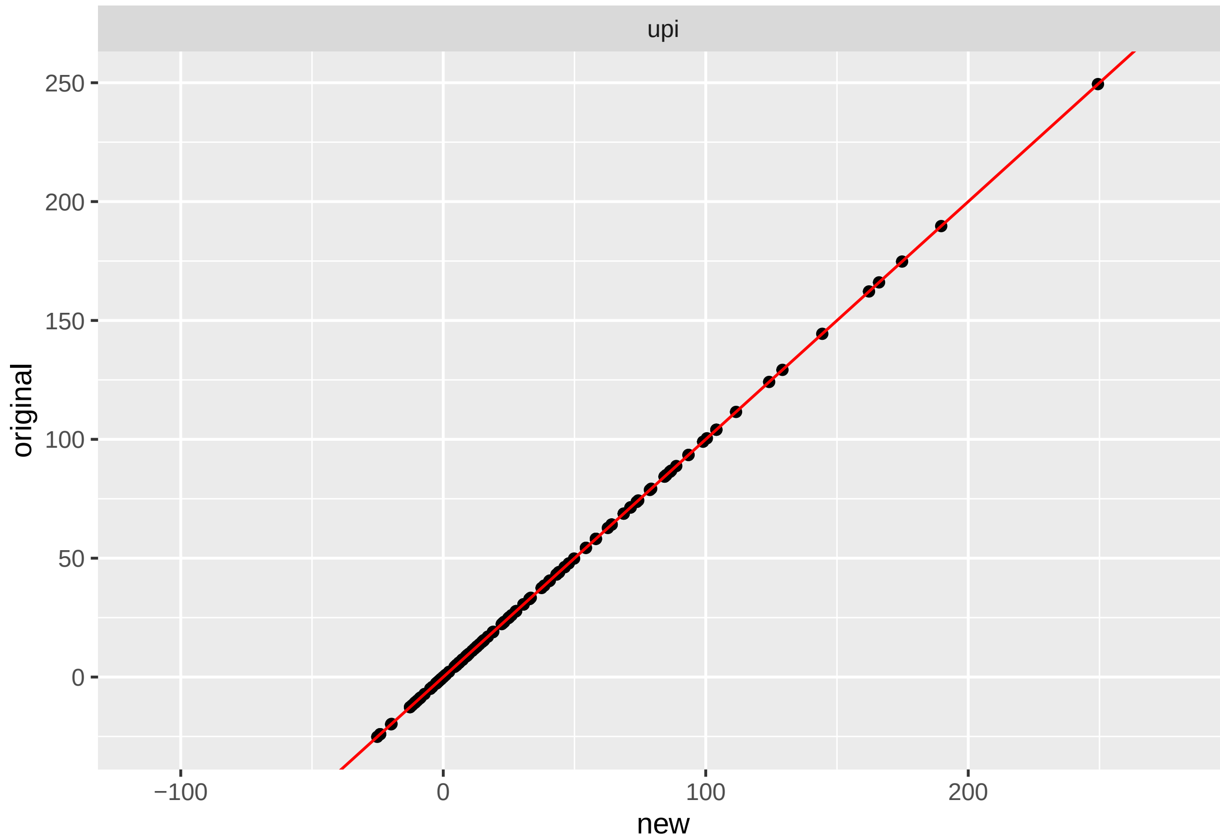
The equation to rotate the EMT vector is the following with the coast angle θ in degrees.

$$\alpha = (360 - \theta)\pi/180 \tag{10}$$

$$upi = \frac{EMT_x cos(\alpha) + EMT_y sin(\alpha)}{10} \tag{11}$$

The `upwell()` function shows how to compute this from the coast angle and the EMT in the x and y directions. Here the upwelling index using the data from the ERD EMT values is added to the data frame.

```
wind1$upi_orig <- upwell(wind1$ektrx_orig, wind1$ektry_orig, 158)
```



Ekman Pumping

$$
\begin{aligned}
\frac{\Delta EMT_x}{\Delta x} &= ((EMT_x)_{x+h,y} - (EMT_x)_{x-h,y})/2h_{m,lon} \\
\frac{\Delta EMT_y}{\Delta y} &= ((EMT_y)_{x,y+h} - (EMT_y)_{x,y-h})/2h_{m,lat}
\end{aligned} \tag{12}
$$

h_m is the resolution of the grid in meters (in the lon east-west and lat north-south directions). The resolution in meters varies by latitude. The following function returns the resolution in meters in the north-south and east-west directions as a function of latitude.

```
h.meter <- function(lat){
    m_per_deg_lat <- 111132.954 - 559.822 * cos( 2 * pi * lat/180 ) +
                     1.175 * cos( 4 * pi * lat / 180)
   m_per_deg_lon <- 111132.954 * cos ( pi*lat/180 )
   return(list(lat=m_per_deg_lat, lon=m_per_deg_lon))
  }
```

Ekman Pumping (m/s) is the sum of these divided by the density of sea water.

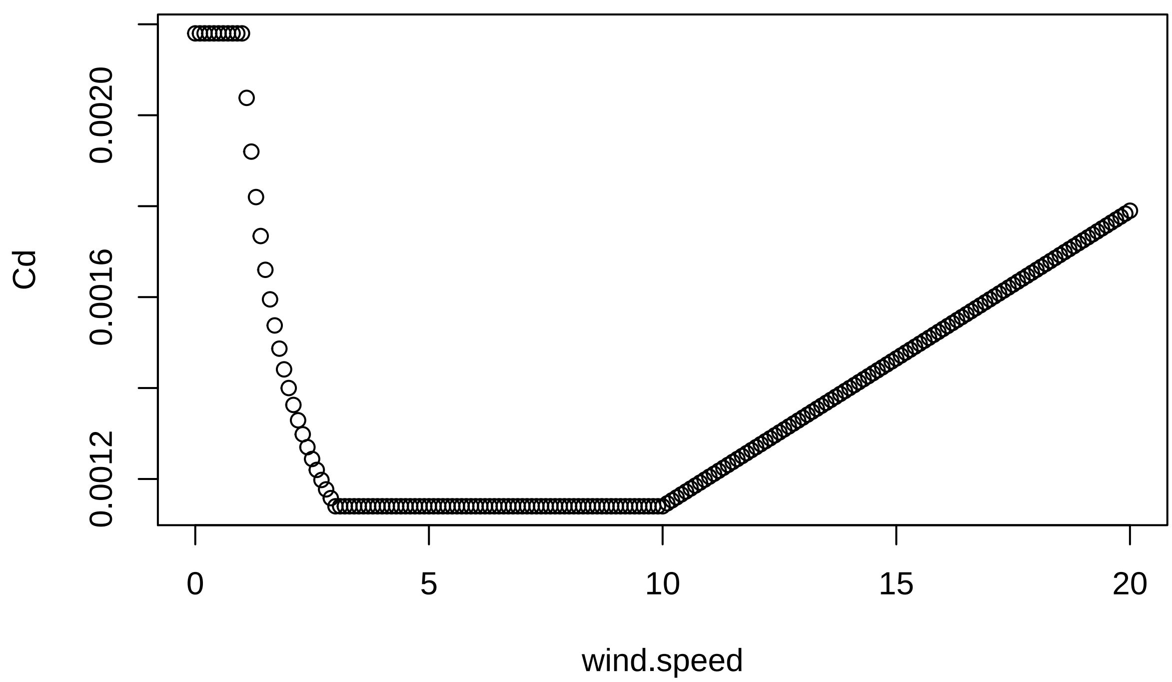
$$
W_e = \frac{1}{p_{sw}}(\frac{\Delta EMT_x}{\Delta x} + \frac{\Delta EMT_y}{\Delta y}) \tag{13}
$$

where p_{sw} is the density of sea water (assumed to be $1023.6 kgm^{-3}$).

Comparison of wind stress from ASCAT winds versus ERD product

In the coastal upwelling products (EMT based) provided by NOAA ERD, a non-linear coefficient of drag is used:

```
plot(seq(0,20,0.1), Cd(seq(0,20,0.1)), xlab="wind.speed", ylab="Cd")
```



Comparing the τ computed from the equations above and that provided by ERD in their ASCAT wind stress product, it is clear that this non-linear C_d is not used. The wind stress products from ERD were not used for this reason.

Get data for a few days in 2010. The wind stress and Ekman upwelling (Pumping) computed by ERD is downloaded from https://coastwatch.pfeg.noaa.gov/erddap/info/erdQAstress1day/index.html and the winds are from https://coastwatch.pfeg.noaa.gov/erddap/info/erdQAwind1day/index.html.

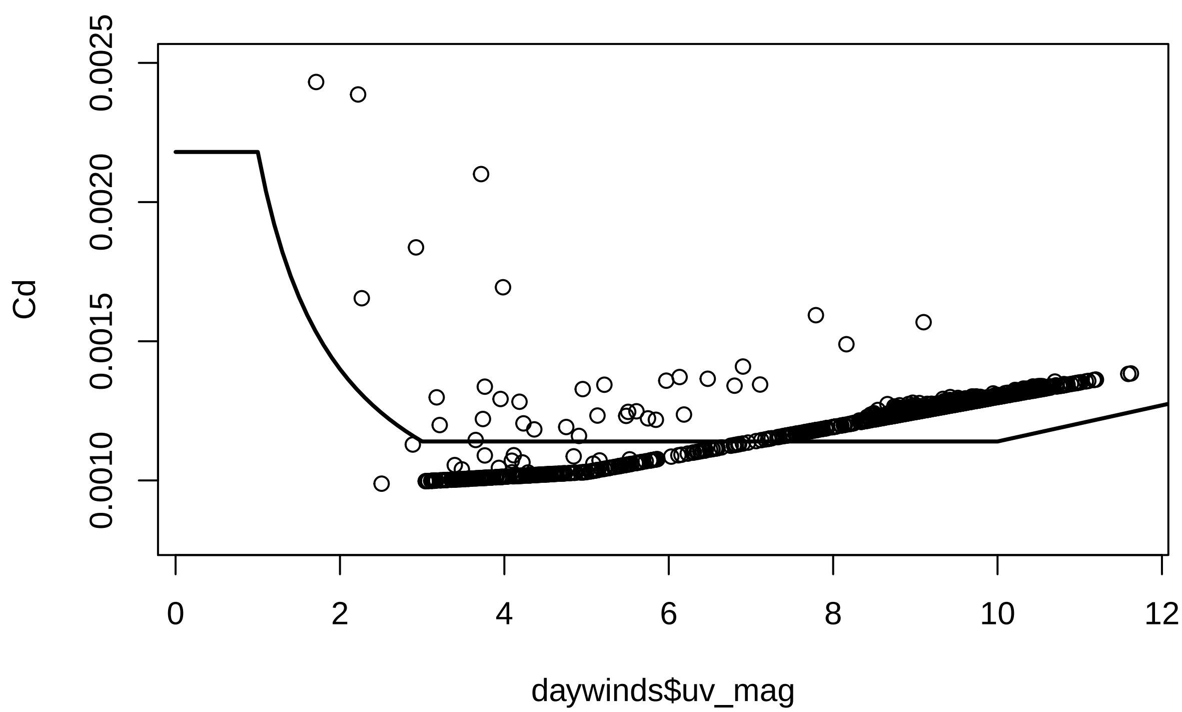
```
#10m winds
lats <- c(7, 15); lons <- c(70,78)
daywinds <- data.frame()
for(dates in c("2010-07-01", "2010-09-01", "2010-12-01")){
tmp <- getdata("erdQAwind1day", date=dates, lat=lats, lon=lons, altitude=10)
windmon5 <- getdata("erdQAstress1day", date=dates, lat=lats, lon=lons, altitude=0)
windmon5$u <- tmp$x_wind
windmon5$v <- tmp$y_wind
windmon5$uv_mag <- sqrt(tmp$y_wind^2 + tmp$x_wind^2)
windmon5$date <- format(windmon5$time, "%Y-%m-01")
windmon5$month <- format(as.Date(windmon5$date), "%m")
```

```
res <- attr(windmon5, "resolution")
windmon5 <- getEMT(windmon5, coast_angle=158)
daywinds <- rbind(daywinds, windmon5)
}
```

```
## data read from erdQAwind1day-7-15-70-78-2010-07-01-2010-07-01.csv
## data erdQAwind1day date 2010-07-01-2010-07-01, latitude 7-15, longitude 70-78
## data read from erdQAstress1day-7-15-70-78-2010-07-01-2010-07-01.csv
## data erdQAstress1day date 2010-07-01-2010-07-01, latitude 7-15, longitude 70-78
## data read from erdQAwind1day-7-15-70-78-2010-09-01-2010-09-01.csv
## data erdQAwind1day date 2010-09-01-2010-09-01, latitude 7-15, longitude 70-78
## data read from erdQAstress1day-7-15-70-78-2010-09-01-2010-09-01.csv
## data erdQAstress1day date 2010-09-01-2010-09-01, latitude 7-15, longitude 70-78
## data read from erdQAwind1day-7-15-70-78-2010-12-01-2010-12-01.csv
## data erdQAwind1day date 2010-12-01-2010-12-01, latitude 7-15, longitude 70-78
## data read from erdQAstress1day-7-15-70-78-2010-12-01-2010-12-01.csv
## data erdQAstress1day date 2010-12-01-2010-12-01, latitude 7-15, longitude 70-78
```

Dividing τ_x computed by ERD by $p_a = 1.22$, wind speed, and zonal wind speed should return C_d. Plotting this we see that the C_d function used is very different. The dots would fall on the solid black line if there were the same.

```
daywinds$a=daywinds$taux_orig/(1.22*daywinds$uv_mag*daywinds$u)
plot(daywinds$uv_mag, daywinds$a, ylim=c(0.0008, 0.0025), ylab="Cd")
lines(seq(0,20,0.1), Cd(seq(0,20,0.1)), lwd=2)
```



Comparison of monthly wind products

Get the FNMOC monthly winds. This is on a 1 degree grid. The data are from https://coastwatch.pfeg.noaa.gov/erddap/info/erdlasFnWPr/index.html.

```
dates <- c("2010-01-01", "2019-12-16")
windmon1 <- getdata("erdlasFnWPr", date=dates, lat=lats, lon=lons)
```

```
## data read from erdlasFnWPr-7-15-70-78-2010-01-01-2019-12-16.csv
## data erdlasFnWPr date 2010-01-01-2019-12-16, latitude 7-15, longitude 70-78
```

```
windmon1$date <- format(windmon1$time, "%Y-%m-01")
colnames(windmon1) <- stringr::str_replace(colnames(windmon1), "_mean", "")
windmon1 <- getEMT(windmon1, coast_angle=158)
```

Now get ASCAT monthly 10m winds https://coastwatch.pfeg.noaa.gov/erddap/info/erdQAwindmday/index.html. . This is on a 0.25 degree grid.

```
#10m winds
windmon2 <- getdata("erdQAwindmday", date=dates, lat=lats, lon=lons, altitude=10)
```

```
## data read from erdQAwindmday-7-15-70-78-2010-01-01-2019-12-16.csv
## data erdQAwindmday date 2010-01-01-2019-12-16, latitude 7-15, longitude 70-78
```

```
windmon2$u <- windmon2$x_wind
windmon2$v <- windmon2$y_wind
windmon2$date <- format(windmon2$time, "%Y-%m-01")
windmon2 <- getEMT(windmon2, coast_angle=158)
```

Now get the Reanalysis Data ERA5 monthly 10m winds from http://apdrc.soest.hawaii.edu/erddap/info/hawaii_soest_66d3_10d8_0f3c/index.html The altitude specification is in millibar. Air pressure at 10m is ca 1000 millibar, thus altitude is set to 1000.

```
windmon4 <- getdata("hawaii_soest_66d3_10d8_0f3c", date=dates, lat=lats, lon=lons,
                    altitude=1000, eserver="http://apdrc.soest.hawaii.edu/erddap",
                    alt.name="LEV")
```

```
## data read from hawaii_soest_66d3_10d8_0f3c-7-15-70-78-2010-01-01-2019-12-16.csv
## data hawaii_soest_66d3_10d8_0f3c date 2010-01-01-2019-12-16, latitude 7-15, longitude 70-78
```

```
windmon4$date <- format(windmon4$time, "%Y-%m-01")
windmon4 <- getEMT(windmon4, coast_angle=158)
```

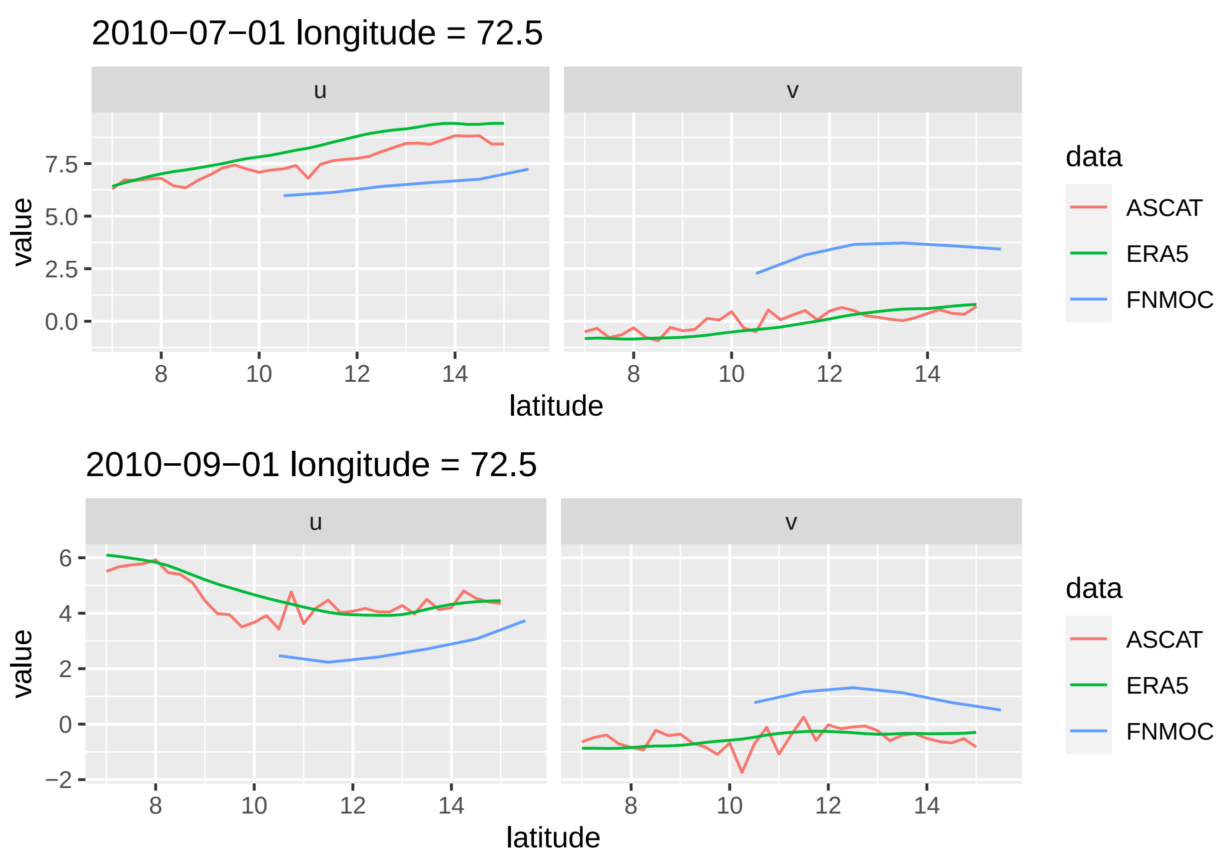
Create the data frame.

```
cols <- c("date", "latitude", "longitude", "u", "v", "EMTperp", "We")
df <- rbind(data.frame(windmon1[,cols], data="FNMOC"),
            data.frame(windmon2[,cols], data="ASCAT"),
            data.frame(windmon4[,cols], data="ERA5"))
dfl <- df %>%
  pivot_longer(!date & !latitude & !longitude & !data,
                     names_to = "name",
                     values_to = "value")
```

Winds on specific dates

FNMOC winds are quite different. ERA5 are much smoother than ASCAT winds.

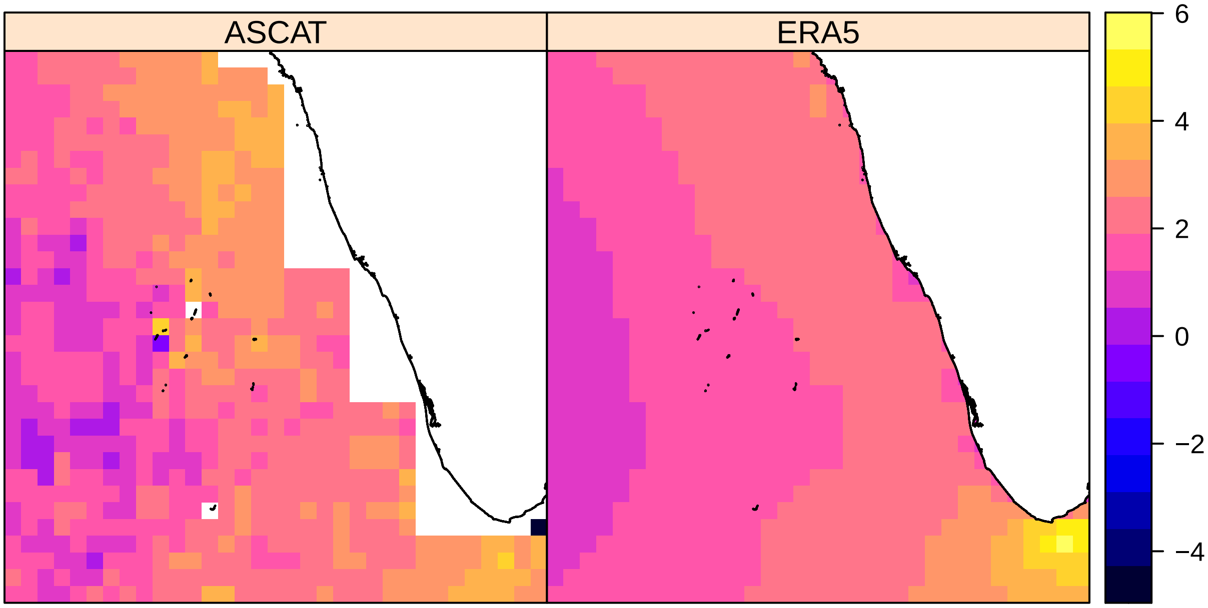
```
thedate <- "2010-07-01"
plotlons <- c(72.5)
pars <- c("u", "v")
p1 <- ggplot(subset(dfl, longitude==plotlons[1] & date==thedate & name%in%pars),
             aes(x=latitude, y=value, color=data)) + geom_line() +
  facet_wrap(~name) + ggtitle(paste(thedate, "longitude =", plotlons[1]))
thedate <- "2010-09-01"
p2 <- ggplot(subset(dfl, longitude==plotlons[1] & date==thedate & name%in%pars),
             aes(x=latitude, y=value, color=data)) + geom_line() +
  facet_wrap(~name) + ggtitle(paste(thedate, "longitude =", plotlons[1]))
gridExtra::grid.arrange(p1,p2)
```



We can see the ASCAT versus ERA5 difference especially for certain months such as April 2010. Here `indiashp2` is a shapefile for the coast of India.

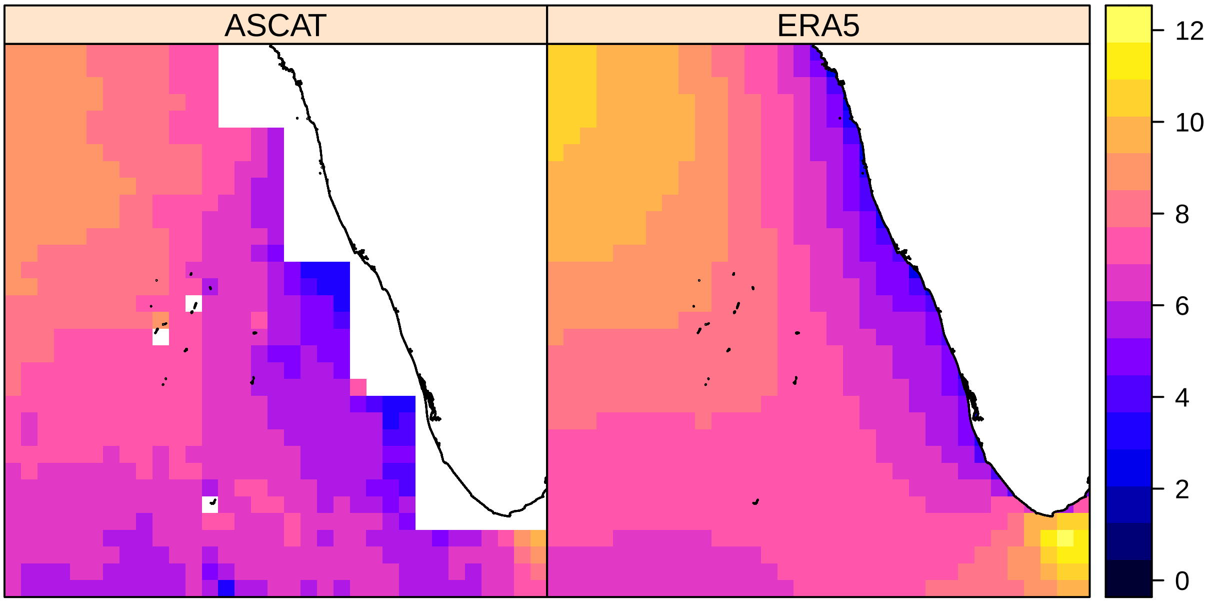
```
thedate <- "2010-04-01"
rdf <- subset(windmon2, date==thedate)
rdf$x <- rdf$longitude
rdf$y <- rdf$latitude
rdf <- rdf[, c("x", "y", "u")]
rst1 <- rasterFromXYZ(rdf); names(rst1) <- "ASCAT"
rdf <- subset(windmon4, date==thedate)
```

```
rdf$x <- rdf$longitude
rdf$y <- rdf$latitude
rdf <- rdf[, c("x", "y", "u")]
rst2 <- rasterFromXYZ(rdf); names(rst2) <- "ERA5"
rst <- stack(rst1, rst2)
proj4string(rst) <- "+proj=longlat +datum=WGS84"
at=seq(-3000,0,50)
spplot(rst) + layer(sp.polygons(indiashp2, fill="white"))
```



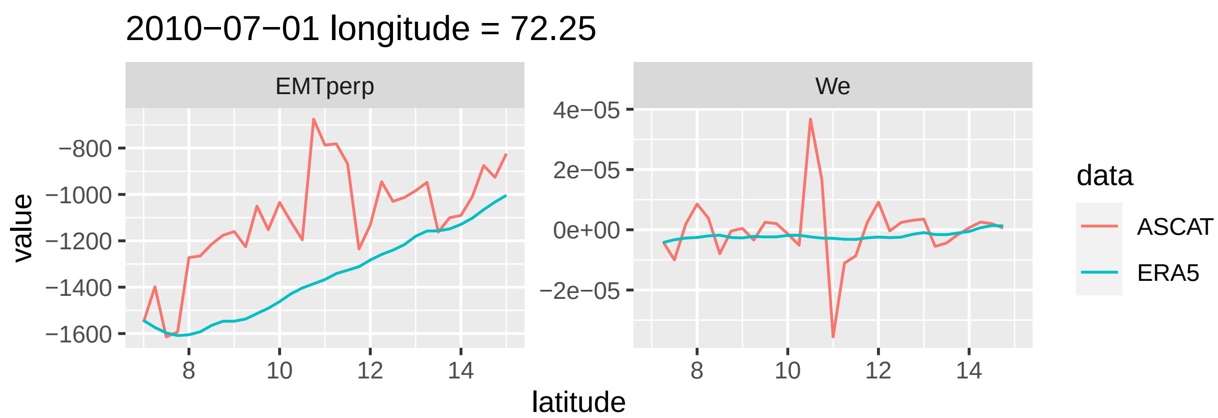
But less so for others, such as August 2010.

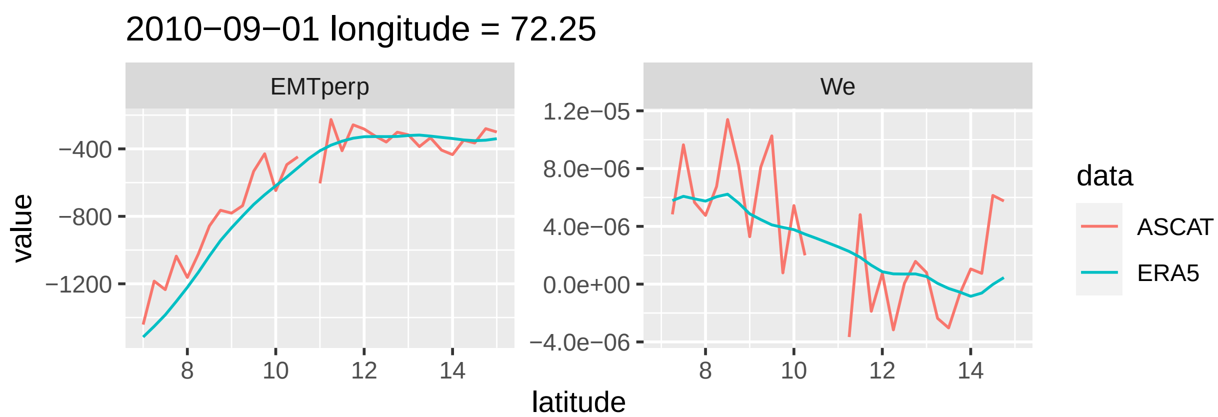
```
thedate <- "2010-08-01"
rdf <- subset(windmon2, date==thedate)
rdf$x <- rdf$longitude
rdf$y <- rdf$latitude
rdf <- rdf[, c("x", "y", "u")]
rst1 <- rasterFromXYZ(rdf); names(rst1) <- "ASCAT"
rdf <- subset(windmon4, date==thedate)
rdf$x <- rdf$longitude
rdf$y <- rdf$latitude
rdf <- rdf[, c("x", "y", "u")]
rst2 <- rasterFromXYZ(rdf); names(rst2) <- "ERA5"
rst <- stack(rst1, rst2)
proj4string(rst) <- "+proj=longlat +datum=WGS84"
at=seq(-3000,0,50)
spplot(rst) + layer(sp.polygons(indiashp2, fill="white"))
```



Ekman Mass Transport and Pumping

```
thedate <- "2010-07-01"
plotlons <- 72.25
pars <- c("EMTperp", "We")
p1 <- ggplot(subset(dfl, longitude==plotlons[1] & date==thedate & name%in%pars), aes(x=latitude, y=valu
  facet_wrap(~name, scales = "free_y") +
  ggtitle(paste(thedate, "longitude =", plotlons[1]))
thedate <- "2010-09-01"
p2 <- ggplot(subset(dfl, longitude==plotlons[1] & date==thedate & name%in%pars), aes(x=latitude, y=valu
  facet_wrap(~name, scales = "free_y") +
  ggtitle(paste(thedate, "longitude =", plotlons[1]))
gridExtra::grid.arrange(p1,p2)
```





Compare the coastal EMT (used in paper)

Define the coast latitude-longitude line.

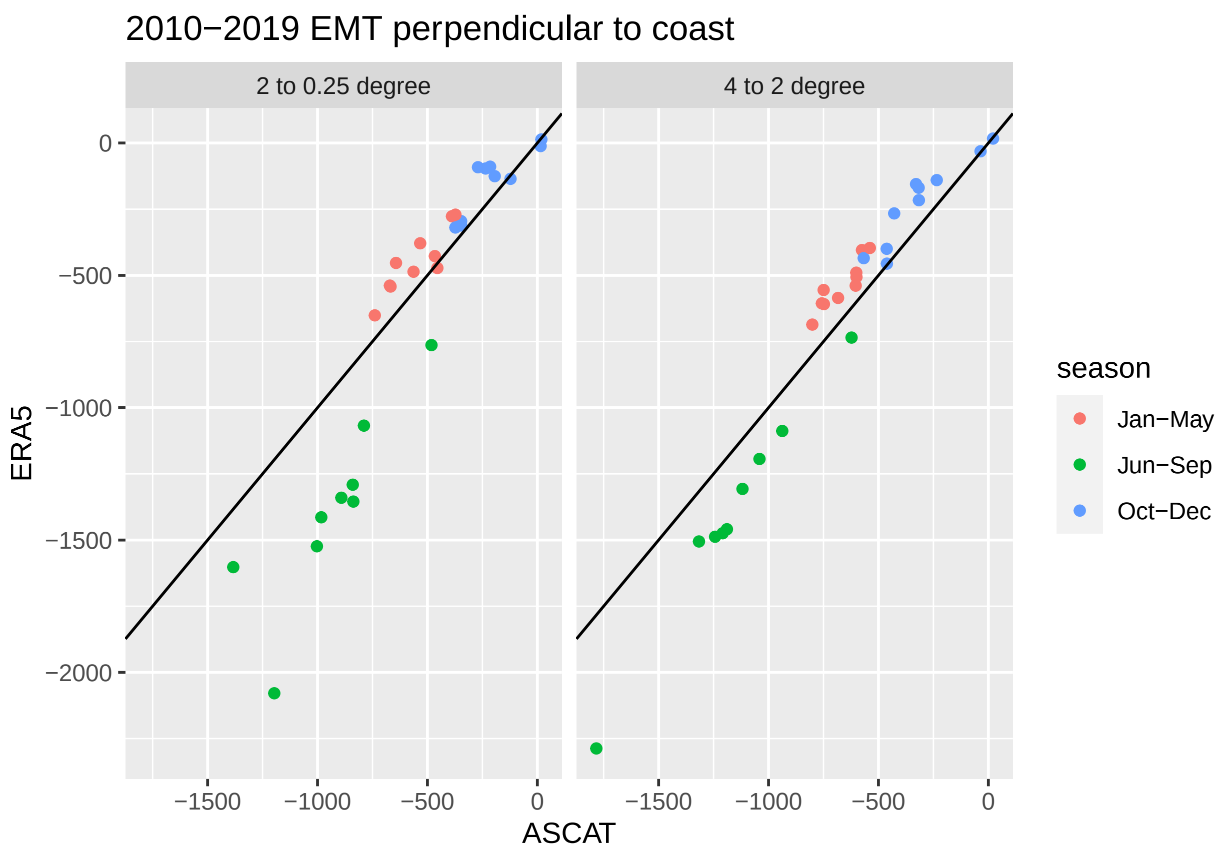
```
b=as(as(indiashp2, "SpatialLinesDataFrame"), "SpatialPointsDataFrame")
b=subset(b, Line.NR==129)
coastcoord=coordinates(b)
coastcoord <- coastcoord[coastcoord[,"x"]<77.5,]
coastlats = unique(windmon2$latitude)
coastlons = c()
for(i in coastlats){
  tmp <- coastcoord[,"y"]
  coastlons <- c(coastlons, min(coastcoord[,"x"][which(abs(tmp-i)==min(abs(tmp-i)))]))
}
```

Define the coast as 2 to 0.25 degrees offshore.

```
dfl2 <- dfl %>%
  pivot_wider(names_from = "data", values_from = "value")
dfl2.coast1 <- subset(dfl2, latitude >= 8 & latitude <= 13)
dfl2.coast2 <- subset(dfl2, latitude >= 8 & latitude <= 13)
for(i in 1:length(coastlats)){
  thelat <- coastlats[i]
  thelon <- coastlons[i]
```

```r
  dfl2.coast1 <- dfl2.coast1[!(dfl2.coast1$latitude==thelat &
    (dfl2.coast1$longitude<(thelon-2) | dfl2.coast1$longitude>=(thelon-0.25))),]
  dfl2.coast2 <- dfl2.coast2[!(dfl2.coast2$latitude==thelat &
    (dfl2.coast2$longitude<(thelon-4) | dfl2.coast2$longitude>=(thelon-2))),]
}
dfl2.coast1$coast <- "2 to 0.25 degree"
dfl2.coast2$coast <- "4 to 2 degree"
dfl2.coast <- rbind(dfl2.coast1, dfl2.coast2)
dfl2.coast$month <- format(as.Date(dfl2.coast$date), "%m")
dfl2.coast$year <- format(as.Date(dfl2.coast$date), "%Y")
dfl2.coast.mean <- dfl2.coast %>% group_by(year, month, name, coast) %>%
  summarize(FNMOC = mean(FNMOC, na.rm=TRUE),
            ASCAT = mean(ASCAT, na.rm=TRUE),
            ERA5 = mean(ERA5, na.rm=TRUE))
dfl2.coast.mean$season <- NA
dfl2.coast.mean$season[as.numeric(dfl2.coast.mean$month) %in% 6:9] <- "Jun-Sep"
dfl2.coast.mean$season[as.numeric(dfl2.coast.mean$month) %in% 10:12] <- "Oct-Dec"
dfl2.coast.mean$season[as.numeric(dfl2.coast.mean$month) %in% 1:5] <- "Jan-May"
```

```r
dfl2.coast.season <- dfl2.coast.mean %>% group_by(year, season, name, coast) %>%
  summarize(FNMOC = mean(FNMOC, na.rm=TRUE),
            ASCAT = mean(ASCAT, na.rm=TRUE),
            ERA5 = mean(ERA5, na.rm=TRUE))
ggplot(subset(dfl2.coast.season, name=="EMTperp" & ASCAT<1000),
       aes(x=ASCAT, y=ERA5, col=season)) + geom_point() + geom_abline() +
  facet_wrap(~coast) +
  ggtitle("2010-2019 EMT perpendicular to coast")
```



Conclusion

For EMT perpendicular to the coast, there is a linear relationship between the ERA5 product versus ASCAT. But the values from ERA5 are biased downward (more negative) in the summer monsoon (Jun-Sep) period. The bias is stronger when using values closer to the coast and suggests that it is due to the nearshore gaps in the ASCAT data seen in the raster plot. The missing raster points nearshore would be those with lower EMT values in the summer. However the generally linear relationship supports using the ERA5 product and the bias would not affect the analysis since the models include an intercept.

Functions in R

```
upwell <-  function (ektrx, ektry, coast_angle)
{
    pi <- 3.1415927
    degtorad <- pi/180
    alpha <- (360 - coast_angle) * degtorad
    s1 <- cos(alpha)
    t1 <- sin(alpha)
    s2 <- -1 * t1
    t2 <- s1
    perp <- (s1 * ektrx) + (t1 * ektry)
    para <- (s2 * ektrx) + (t2 * ektry)
    return(perp/10)
```

```
}

Cd <-  function (wind.speed)
{
    wind.orig <- wind.speed
    wind.speed[is.na(wind.speed)] <- -Inf
    Cd <- rep(NA, length(wind.orig))
    Cd[wind.speed <= 1] <- 2.18
    Cd[wind.speed < 3 & wind.speed > 1] <- 0.62 + 1.56/wind.speed[wind.speed <
        3 & wind.speed > 1]
    Cd[wind.speed >= 3 & wind.speed < 10] <- 1.14
    Cd[wind.speed >= 10] <- 0.49 + 0.065 * wind.speed[wind.speed >=
        10]
    Cd[is.na(wind.orig)] <- NA
    return(Cd * 0.001)
}

EMTperp <-  function (ektrx, ektry, coast_angle)
{
    pi <- 3.1415927
    degtorad <- pi/180
    alpha <- (180 - coast_angle) * degtorad
    s1 <- cos(alpha)
    t1 <- sin(alpha)
    s2 <- -1 * t1
    t2 <- s1
    perp <- (s1 * ektrx) + (t1 * ektry)
    para <- (s2 * ektrx) + (t2 * ektry)
    return(list(perp = perp, para = para))
}

getEMT <-  function (dat, coast_angle = NULL)
{
    if (!any(colnames(dat) == "u"))
        stop("need u in data")
    if (!any(colnames(dat) == "v"))
        stop("need v in data")
    for (i in c("uv_mag", "tau", "taux", "tauy", "EMT", "ektrx",
        "ektry", "upi")) if (any(colnames(dat) == i)) {
        warning(paste0(i, " is already in data. Changing to ",
            i, "_orig\n"))
        colnames(dat)[colnames(dat) == i] <- paste0(i, "_orig")
    }
    pa <- 1.22
    omega <- 7.272205e-05
    f <- 2 * omega * sin(pi * dat$latitude/180)
    uv_mag <- sqrt(dat$u^2 + dat$v^2)
    tau <- pa * Cd(uv_mag) * uv_mag * uv_mag
    tauy <- pa * Cd(uv_mag) * uv_mag * dat$v
    taux <- pa * Cd(uv_mag) * uv_mag * dat$u
    EMT <- tau/f
    EMTy <- -1 * taux/f
    EMTx <- tauy/f
```

```
    dat$uv_mag <- uv_mag
    dat$tau <- tau
    dat$taux <- taux
    dat$tauy <- tauy
    dat$EMT <- EMT
    dat$ektrx <- EMTx
    dat$ektry <- EMTy
    dat$We <- getEPump(dat)$We
    if (!missing(coast_angle)) {
        dat$EMTperp <- EMTperp(dat$ektrx, dat$ektry, coast_angle)$perp
        dat$upi <- upwell(dat$ektrx, dat$ektry, coast_angle)
    }
    return(dat)
}

getwindfromP <-  function (dat)
{
    if (is.null(attr(dat, "resolution")))
        stop("need resolution in the data.frame")
    if (!any(c("P_msl", "p_msl") %in% colnames(dat)))
        stop("need P_msl or p_msl (pressure) in data")
    for (i in c("u", "v", "P")) if (any(colnames(dat) == i)) {
        warning(paste0(i, " is already in data. Changing to ",
            i, "_orig", "\n"))
        colnames(dat)[colnames(dat) == i] <- paste0(i, "_orig")
    }
    res <- attr(dat, "resolution")
    lons <- sort(unique(dat$longitude))
    lats <- sort(unique(dat$latitude))
    dat$ug <- NA
    dat$vg <- NA
    dat$P <- 100 * dat[, tolower(colnames(dat)) == "p_msl"]
    h <- pi * res/180
    pa <- 1.22
    omega <- 7.272205e-05
    f <- 2 * omega * sin(pi * dat$latitude/180)
    R <- 6371 * 1000
    if (length(lats) > 2)
        for (lat in (min(lats) + res):(max(lats) - res)) {
            dPdlat <- (dat$P[dat$latitude == (lat + res)] - dat$P[dat$latitude ==
                (lat - res)])/(2 * h)
            constant <- (f * pa * R)[dat$latitude == lat]
            ug <- -1 * dPdlat/constant
            dat$ug[dat$latitude == lat] <- ug
        }
    if (length(lons) > 2)
        for (lon in (min(lons) + res):(max(lons) - res)) {
            dPdlon <- (dat$P[dat$longitude == (lon + res)] -
                dat$P[dat$longitude == (lon - res)])/(2 * h)
            constant <- (f * pa * R * cos(pi * dat$latitude/180))[dat$longitude ==
                lon]
            vg <- dPdlon/constant
            dat$vg[dat$longitude == lon] <- vg
        }
```

```
    ang <- pi * 30/360
    u <- 0.7 * (cos(ang) * dat$ug - sin(ang) * dat$vg)
    v <- 0.7 * (sin(ang) * dat$ug + cos(ang) * dat$vg)
    dat$u <- u
    dat$v <- v
    return(dat)
}

getdata <-  function (id, pars = NULL, lat = c(7, 15), lon = c(70, 78), date = NULL,
    altitude = 10, alt.name = "altitude", eserver = "https://coastwatch.pfeg.noaa.gov/erddap")
{
    url <- paste0(eserver, "/info/", id, "/index.csv")
    meta <- read.csv(url)
    if (!missing(date) && length(date) == 1)
        date <- c(date, date)
    if (missing(date))
        date <- c(meta$Value[meta$Attribute.Name == "time_coverage_start"],
            meta$Value[meta$Attribute.Name == "time_coverage_end"])
    lat1 <- lat[1]
    lat2 <- lat[2]
    lon1 <- lon[1]
    lon2 <- lon[2]
    fil <- paste0(id, "-", lat1, "-", lat2, "-", lon1, "-", lon2,
        "-", stringr::str_sub(date[1], 1, 10), "-", stringr::str_sub(date[2],
            1, 10), ".csv")
    dfil <- file.path(here::here(), "SupplementFiles", fil)
    if (!file.exists(dfil)) {
        if (missing(pars)) {
            pars <- unique(meta$Variable.Name)
            pars <- pars[!(pars %in% c("NC_GLOBAL", "time", "latitude",
                "longitude", alt.name))]
        }
        alttag <- ifelse(alt.name %in% meta$Variable.Name, paste0("[(",
            altitude, "):1:(", altitude, ")]"), "")
        val <- paste0("[(", date[1], "):1:(", date[2], ")]",
            alttag, "[(", lat1, "):1:(", lat2, ")][(", lon1,
            "):1:(", lon2, ")]")
        val2 <- paste0(pars, val, collapse = ",")
        url <- paste0(eserver, "/griddap/", id, ".csv?", val2)
        download.file(url, destfile = dfil)
        cat("data saved to", fil, "\n")
    }
    else {
        cat("data read from", fil, "\n")
    }
    dat <- read.csv(file = dfil, stringsAsFactors = FALSE)
    dat <- dat[-1, ]
    for (i in 2:ncol(dat)) dat[[i]] <- as.numeric(dat[[i]])
    dat$time <- as.POSIXlt(dat$time, format = "%Y-%m-%dT%H:%M:%OSZ",
        tz = "UTC")
    attr(dat, "resolution") <- min(abs(diff(dat$latitude))[abs(diff(dat$latitude)) !=
        0], na.rm = TRUE)
    cat(paste0("data ", id, " date ", date[1], "-", date[2],
        ", latitude ", lat1, "-", lat2, ", longitude ", lon1,
```

```
        "-", lon2), "\n")
    dat
}
```