Package ‘coastMDT’

January 5, 2017

Title Estimate a coastal MDT based on various input files

Version 0.0.1

Date 2016-09-16

Author Karina Nielsen, Ole B. Andersen, Per Knudsen, Christopher W. Hughes, and Mederic Gravelle

Maintainer Karina Nielsen <karni@space.dtu.dk>

Description More about what it does (maybe more than one line)

License GNU LESSER GENERAL PUBLIC LICENSE

Imports ncdf4,
smoothie,
fields,
FNN,
raster,
sp,
akima

URL https://github.com/cavios/coastMDT

LazyData TRUE

BugReports https://github.com/cavios/coastMDT/issues

R topics documented:

compareWithTG *Extract MDT values at TG positions*

Description

The function extracts MDT values at the position of the tide gauges, and compares the MDT values of the field with MDT values based on the tide gauges.

Usage

```
compareWithTG(TG, dat, lonlim, latlim, boxlon = 3, boxlat = 3)
```

Arguments

`TG`	Data frame or matrix with tide gauge information. The dimension of TG is N x 4, where N is the number of tide gauges and the 4 columns are; PSMSL station id, latitude, longitude, MDT value at the tide gauges.
`dat`	Matrix[lon,lat] with MDT values
`lonlim`	Vector of length 2 with the longitude data grid limits, c(lonlim[1],lonlim[2]). The limits must be given in whole degrees.
`latlim`	Vector of length 2 with the longitude data grid limits, c(lonlim[1],lonlim[2]). The limits must be given in whole degrees.
`boxlon`	The number ((2 x boxlon) +1) of grid cells in the longitude direction, that is used to estimate the altimetry based MDT value at the coast.
`boxlat`	The number ((2 x boxlat) +1) of grid cells in the latitude direction, that is used to estimate the altimetry based MDT value at the coast.

Details

Besides the list, that is returned. A plot of the difference between the altimetry and the tide gauges MDT values is automatically generated

Value

A list that includes: mean: The mean values of the field in the box, defined by boxlon and boxlat at each tide gauge position. sd: The standard deviation of the field in the box, defined by boxlon and boxlat at each tide gauge position. bias: The bias between the mean field values and the tide gauge MDT values. diff: The difference between the mean field values and the bias corrected tide gauge MDT values RMS: The RMS of the mean field values and the tide gauge MDT values

Examples

```
#load data
data(landmask8) #land mask
data(difmss15eig6c4r)
data(MDT_EGM08_2003_2007)
TG<-MDT_EGM08_2003_2007
#region of interest
lonlim<-c(275,300)
latlim<-c(20,55)
#sub grids
mask<-getSubGrid(landmask8,lonlim,latlim)
rawUS<-getSubGrid(difmss15eig6c4r,lonlim,latlim)
idraw<-which(mask$g==0,arr.ind=TRUE)
rawUS$g[idraw]<-NA
TGsub<-getSubTG(TG,lonlim,latlim)
#Compare with tide gauges
res<-compareWithTG(TGsub,rawUS$g,lonlim,latlim)
```

dacCor5Y_2003_2007 *dacCor5Y_2003_2007*

Description

Dynamic atmosphere corrections for the reference period 2003-2007. The corrections are given on the following grid:

2880 = longitudes (0.5+(0,1,2....2879))/8. degrees 1440 = latitudes (0.5+(0,1,2....1439))/8. -90 degrees

These numbers should be added to ssh in metres to give ib-corrected ssh

They are derived from the DAC, as provided by aviso at 6-hour and 1/4 degree resolution.

dacCor5Y_2003_2007 represents the 5-year average of the period 2003 to 2007.

Usage

```
data("dacCor5Y_2003_2007")
```

Format

A matrix[lon,lat] of dimension 2880 x 1440

Details

...

Source

...

References

Created by Christopher W. Hughes `<cwh@noc.ac.uk>`

Examples

```
data(dacCor5Y_2003_2007)
```

---

dDTU15MSS_ref2003_2007

dDTU15MSS_ref2003_2007

Description

Grid to transform the DTU15 MSS (DTU15MSS) to the MSS of the period 2003-2007. The grid should be added. dDTU15MSS_ref2003_2007 is a 2880 x 1440 matrix.

The first cell is longitude 0 to 1/8 degree, latitude -90 to -90+1/8 degree cells are all 1/8 by 1/8 degree order is east to west, then south to north

Usage

```
data("dDTU15MSS_ref2003_2007")
```

Format

A matrix[lon,lat] of dimension 2880 x 1440

Details

...

Source

...

References

Created by Ole B. Andersen `<oa@space.dtu.dk>`

Examples

```
data(dDTU15MSS_ref2003_2007)
```

`difmss15eig6c4r` *difmss15eig6c4r*

Description

difmss15eig6c4r is an unfiltered MDT based on the MSS DTU15 and the geoid model Eigen6c4r. difmss15eig6c4r is a 2880 x 1440 matrix.

The first cell is longitude 0 to 1/8 degree, latitude -90 to -90+1/8 degree cells are all 1/8 by 1/8 degree order is east to west, then south to north

Usage

```
data("difmss15eig6c4r")
```

Format

A matrix[lon,lat] of dimension 2880 x 1440

Details

...

Source

...

References

Created by Per Knudsen

Examples

```
data(difmss15eig6c4r)
```

DTU15MSS *DTU15MSS*

Description

DTU15MSS is the DTU15 mean sea surface (MSS). DTU15MSS is a 2880 x 1440 matrix.

The first cell is longitude 0 to 1/8 degree, latitude -90 to -90+1/8 degree cells are all 1/8 by 1/8 degree order is east to west, then south to north

Usage

```
data("DTU15MSS")
```

Format

A matrix[lon,lat] of dimension 2880 x 1440

Details

...

Source

...

References

Created by Ole B. Andersen `<oa@space.dtu.dk>`

Examples

```
data(DTU15MSS)
```

eigen6c4r *eigen6c4r*

Description

eigen6c4r is a geoid model based on EIGEN-6C4. Eigen6c4r is a 2880 x 1440 matrix.

The first cell is longitude 0 to 1/8 degree, latitude -90 to -90+1/8 degree cells are all 1/8 by 1/8 degree order is east to west, then south to north

Usage

```
data("eigen6c4r")
```

Format

A matrix[lon,lat] of dimension 2880 x 1440

Details

...

Source

...

References

F\"orste, Christoph; Bruinsma, Sean.L.; Abrikosov, Oleg; Lemoine, Jean-Michel; Marty, Jean Charles; Flechtner, Frank; Balmino, G.; Barthelmes, F.; Biancale, R. (2014): EIGEN-6C4 The latest combined global gravity field model including GOCE data up to degree and order 2190 of GFZ Potsdam and GRGS Toulouse. GFZ Data Services. http://doi.org/10.5880/icgem.2015.1

Examples

```
data(eigen6c4r)
```

`getBoxMean` *Helper function: Find mean value in a box*

Description

Helper function: Find mean value in a box

Usage

```
getBoxMean(dat, id, boxlon = 4, boxlat = 4)
```

Arguments

`dat`	Matrix[lon,lat] with MDT values
`id`	Matrix[N,2] with row and column id, reprecenting the center of the box
`boxlon`	Integer. The number ((2 x boxlon) +1) of grid cells in the longitude direction, that is used to estimate the altimetry based MDT value at the coast.
`boxlat`	Integer. The number ((2 x boxlat) +1) of grid cells in the latitude direction, that is used to estimate the altimetry based MDT value at the coast.

Details

...

Value

list with mean and sd values

getCoastLine *Helper function to getLandVal: Extract the coast line from the land mask*

Description

Helper function to getLandVal: Extract the coast line from the land mask

Usage

```
getCoastLine(mask, land = 0, water = 1)
```

Arguments

`mask`	Matrix[lon,lat] with land mask values.
`land`	Integer value for land. Default is 0.
`water`	Integer value for land. Default is 1.

Details

...

Value

List with the elements; matrix g[lon,lat], which contains the location of the land value which has a water neighbor. Matrix id which contains two columns; row no and col no of the matrix g where a land value is identified as coast.

getLandComb *Helper funtion to getLandVal: Finds land values based on tide gauges and altimetry*

Description

Helper funtion to getLandVal: Finds land values based on tide gauges and altimetry

Usage

```
getLandComb(polyCoast, TG, TGcorr, dat, lonlim, latlim, boxlon = 4,
  boxlat = 4)
```

Arguments

`polyCoast` Matrix[lon,lat], the out put of the function polygonizeCoast. The matrix contains the coastlines of the region defined by lonlim and latlim, where the integer values represents the coast id.

`TG` Data frame or matrix with tide gauge information. The dimension of TG is N x 4, where N is the number of tide gauges and the 4 columns are; PSMSL station id, latitude, longitude, MDT value at the tide gauge.

`TGcorr` Vector with bias corrected tide gauge values. Obtained from the helper function getTGVal.

`dat` Matrix[lon,lat] with MDT values

`lonlim` Vector of length 2 with the longitude data grid limits, c(lonlim[1],lonlim[2]). The limits must be given in whole degrees.

`latlim` Vector of length 2 with the longitude data grid limits, c(lonlim[1],lonlim[2]). The limits must be given in whole degrees.

`boxlon` Integer. The number ((2 x boxlon) +1) of grid cells in the longitude direction, that is used to estimate the altimetry based MDT value at the coast.

`boxlat` Integer. The number ((2 x boxlat) +1) of grid cells in the latitude direction, that is used to estimate the altimetry based MDT value at the coast.

Details

...

Value

Matrix[lon,lat] with MDT land values at the coast line, defined by polyCoast

`getLandInfo` *Helper function to getLandVal: Estimate land MDT values at the coastline based on altimetry*

Description

Helper function to getLandVal: Estimate land MDT values at the coastline based on altimetry

Usage

```
getLandInfo(mycoast, mask, dat, boxlon = 4, boxlat = 4)
```

Arguments

`mycoast`	Matrix[N,2] with row and column values of the coast line. mycoast is the out put of helper function getCoastLine
`mask`	Matrix[lon,lat] representing the land mask, where land=0 and water=1
`dat`	Matrix[lon,lat] with MDT values
`boxlon`	Integer. The number ((2 x boxlon) +1) of grid cells in the longitude direction, that is used to estimate the altimetry based MDT value at the coast.
`boxlat`	Integer. The number ((2 x boxlat) +1) of grid cells in the latitude direction, that is used to estimate the altimetry based MDT value at the coast.

Details

...

Value

Matrix[N,4]; row id, col id, mean MDT value, sd of MDT

`getLandVal` *Estimates MDT land values based based on altimetry, tide gauges of both*

Description

Estimates MDT land values based based on altimetry, tide gauges of both

Usage

```
getLandVal(dat, mask, lonlim, latlim, TG = NULL, type = "alt",
  intMethod = "lin", boxlon = 4, boxlat = 4)
```

Arguments

`dat`	Matrix[lon,lat] with MDT values
`mask`	Matrix[lon,lat] representing the land mask, where land=0 and water=1
`lonlim`	Vector of length 2 with the longitude data grid limits, c(lonlim[1],lonlim[2]). The limits must be given in whole degrees
`latlim`	Vector of length 2 with the longitude data grid limits, c(lonlim[1],lonlim[2]). The limits must be given in whole degrees
`TG`	Data frame or matrix with tide gauge information. The dimension of TG is N x 4, where N is the number of tide gauges and the 4 columns are; PSMSL station id, latitude, longitude, MDT value at the tide gauge

`type` Character string representing the data to be used when the land values are estimated. type="tg": tide gauge data is used to estimate the MDT land values, type="alt": altimetry data is used, and type="both": both altimetry and tide gauge data is used.

`intMethod` Character string describing the interpolation method used. "lin": linear interpolation and "nn": nearest neighbor interpolation

`boxlon` Integer. The number ((2 x boxlon) +1) of grid cells in the longitude direction, that is used to estimate the altimetry based MDT value at the coast.

`boxlat` Integer. The number ((2 x boxlat) +1) of grid cells in the latitude direction, that is used to estimate the altimetry based MDT value at the coast.

Details

...

Value

Matrix[lon,lat] with land values

Examples

```
#load data
data(landmask8) #land mask
data(difmss15eig6c4r)
data(MDT_EGM08_2003_2007)
TG<-MDT_EGM08_2003_2007
#region of interest
lonlim<-c(275,300)
latlim<-c(20,55)
#sub grids
mask<-getSubGrid(landmask8,lonlim,latlim)
rawUS<-getSubGrid(difmss15eig6c4r,lonlim,latlim)
idraw<-which(mask$g==0,arr.ind=TRUE)
rawUS$g[idraw]<-NA
TGsub<-getSubTG(TG,lonlim,latlim)
#Estimation of land data
mylandTG<-getLandVal(rawUS$g,TGsub,mask$g,lonlim,latlim,type="tg",intMethod="lin")
```

`getSubGrid` *Extract sub grid*

Description

Extract sub grid

Usage

```
getSubGrid(grid, lonlim, latlim, res = 0.125, glonlim = c(0 + (res/2), 360 -
  (res/2)), glatlim = c(-90 + (res/2), 90 - (res/2)))
```

Arguments

grid — Input grid[longitude,latitude] of type matrix.

lonlim — Vector of length two containing the longitude limits of the sub grid. The limits must be given with the smallest longitude first for example c(270,300), except when the 0 longitude id crossed. In this case for example c(355,10). Acceptable values are between 0 and 360.

latlim — Vector of length two containing the latitude limits of the sub grid.

res — The resolution of the input grid in decimal degrees.

glonlim — Vector of length two containing the longitude limits of the input grid. The default is c(0,360).

glatlim — Vector of length two containing the latitude limits of the input grid. The default is c(-90,90).

Details

...

Value

List with the elements; matrix g (sub grid), vector lon (longitudes), vector lat (latitudes) ##'

Examples

```
data(landmask8)
out<-getSubGrid(landmask8,c(280,300),c(30,60))
image(out$lon,out$lat,out$g)
```

getSubTG — *Find subset of Tide gauges*

Description

Find subset of Tide gauges

Usage

```
getSubTG(TG, lonlim, latlim)
```

Arguments

TG — Data frame or matrix with tide gauge information. The dimension of TG is N x 4, where N is the number of tide gauges and the 4 columns are; PSMSL station id, latitude, longitude, MDT value at the tide gauge.

lonlim — Vector of length 2 with the longitude data grid limits, c(lonlim[1],lonlim[2]). The limits must be given in whole degrees.

latlim — Vector of length 2 with the longitude data grid limits, c(lonlim[1],lonlim[2]). The limits must be given in whole degrees.

Value

subset of TG

`getTGCoast` *Helper function to getLandVal: Identify Coast lines with tide gauges*

Description

Helper function to getLandVal: Identify Coast lines with tide gauges

Helper function: Find index of coast line with Tide gauges

Usage

```
getTGCoast(polyCoast, TG, lonlim, latlim)

getTGCoast(polyCoast, TG, lonlim, latlim)
```

Arguments

`polyCoast` Matrix[lon,lat], the out put of the function polygonizeCoast. The matrix contains the coastlines of the region defined by lonlim and latlim, where the integer values represents the coast id.

`TG` Data frame or matrix with tide gauge information. The dimension of TG is N x 4, where N is the number of tide gauges and the 4 columns are; PSMSL station id, latitude, longitude, MDT value at the tide gauge.

`lonlim` Vector of length 2 with the longitude data grid limits, c(lonlim[1],lonlim[2]). The limits must be given in whole degrees.

`latlim` Vector of length 2 with the longitude data grid limits, c(lonlim[1],lonlim[2]). The limits must be given in whole degrees

`polyCoast` Matrix[lon,lat], the out put of the function polygonizeCoast. The matrix contains the coastlines of the region defined by lonlim and latlim, where the integer values represents the coast id.

`TG` Data frame or matrix with tide gauge information. The dimension of TG is N x 4, where N is the number of tide gauges and the 4 columns are; PSMSL station id, latitude, longitude, MDT value at the tide gauge.

`lonlim` Vector of length 2 with the longitude data grid limits, c(lonlim[1],lonlim[2]). The limits must be given in whole degrees.

`latlim` Vector of length 2 with the longitude data grid limits, c(lonlim[1],lonlim[2]). The limits must be given in whole degrees.

Details

...

Value

Vector with coast id for the tide gauges.

coastid for each TG

`getTGid` *Helper function: Finds row and col id of Tide gauges*

Description

Helper function: Finds row and col id of Tide gauges

Usage

```
getTGid(TG, lonlim, latlim)
```

Arguments

`TG`	Data frame or matrix with tide gauge information. The dimension of TG is N x 4, where N is the number of tide gauges and the 4 columns are; PSMSL station id, latitude, longitude, MDT value at the tide gauge.
`lonlim`	Vector of length 2 with the longitude data grid limits, c(lonlim[1],lonlim[2]). The limits must be given in whole degrees.
`latlim`	Vector of length 2 with the longitude data grid limits, c(lonlim[1],lonlim[2]). The limits must be given in whole degrees.

Value

Matrix[N,2] with row and coulmn id for the tide gauges

`getTGVal` *Helper function to getLandVal:*

Description

Extract MDT values at TG positions and estimates a potential bias between the tide gauges and the model based MDT.

Usage

```
getTGVal(TG, dat, mask, lonlim, latlim, boxlon = 4, boxlat = 4)
```

Arguments

`TG` Data frame or matrix with tide gauge information. The dimension of TG is N x 4, where N is the number of tide gauges and the 4 columns are; PSMSL station id, latitude, longitude, MDT value at the tide gauge.

`dat` Matrix[lon,lat] with MDT values

`lonlim` Vector of length 2 with the longitude data grid limits, c(lonlim[1],lonlim[2]). The limits must be given in whole degrees.

`latlim` Vector of length 2 with the longitude data grid limits, c(lonlim[1],lonlim[2]). The limits must be given in whole degrees.

`boxlon` Integer. The number ((2 x boxlon) +1) of grid cells in the longitude direction, that is used to estimate the altimetry based MDT value at the coast.

`boxlat` Integer. The number ((2 x boxlat) +1) of grid cells in the latitude direction, that is used to estimate the altimetry based MDT value at the coast.

Details

...

Value

list(TGland=out,bias=bias). TGland is data frame with 4 columns; row id, col id, corrected tide gauge value, sd of boxmean value of modeled MDT

`ibCor5Y_2003_2007` *ibCor5Y_2003_2007*

Description

Inverse barometer correction for the 5 year reference period given on the following grid:

2880 = longitudes (0.5+(0,1,2....2879))/8. degrees 1440 = latitudes (0.5+(0,1,2....1439))/8. -90 degrees

These numbers should be added to ssh in metres to give ib-corrected ssh

They are derived from monthly-mean sea-level pressure from the era-interim analysis, as provided by ECMWF at 1/4 degree resolution. The conversion from pressure in Pa to sea level correction in m is given by correction = -1.e-4*(0.99*(p-101100.)-0.974*(pglob-101100.)) where p is pressure and pglob is global-ocean-average pressure at the same time.

(the reason for the different coefficients, 0.99 and 0.974, is that global ocean average pressure does produce a small sea level change due to compressibility of seawater - without this effect, the formula would reduce to -1.e-4*0.99*(p-pglob))

ibCor5Y_2003_2007 represents the 5-year average of the period 2003 to 2007.

NOTE: Gibbs fringes can be seen in both these products, slightly larger in the ib than the dac, but are typically at the 1 mm level or less (the largest ocean values occur by the Pacific South American coast).

Usage

```
data("ibCor5Y_2003_2007")
```

Format

A matrix[lon,lat] of dimension 2880 x 1440

Details

...

Source

...

References

Created by Christopher W. Hughes <cwh@noc.ac.uk>

Examples

```
data(ibCor5Y_2003_2007)
```

iterativeAveSmoother *Iterative box filter*

Description

This function `iterativeAveSmoother` ....

Usage

```
iterativeAveSmoother(dat, mask, land, latlim, radius = 0.15/0.83, nit = 10,
  res = 0.125)
```

Arguments

`dat`	Matrix[lon,lat] to be filtered
`mask`	Matrix[lon,lat] containing land/ocean values. Land=0, water=1.
`land`	Matrix[lat,lon] containing land values
`latlim`	Vector of length 2 with the longitude data grid limits, c(lonlim[1],lonlim[2]). The limits must be given in whole degrees.
`radius`	Filter radius. Default is radius=0.15/0.83
`nit`	Number of iterations of the box filter. Default is nit=10
`dlat`	Grid spacing of the matrix dat. Default is dlat=0.125

Details

...

Value

Matrix[lat,lon] with filtered MDT values.

`landmask8` *landmask8*

Description

landmasks from GEBCO2014, on a 1/8 degree grid.

landmask8 is a 2880 x 1440 matrix

first cell is longitude 0 to 1/8 degree, latitude -90 to -90+1/8 degree cells are all 1/8 by 1/8 degree order is east to west, then south to north

value is 1 for any cell which is 50 0 for any cell which is less than 50

isolated ocean points have been removed (excluding Black Sea and Sea of Marmara)

Usage

```
data(landmask8)
```

Format

A matrix of dimension 2880 x 1440

Details

points below sea level, but enclosed by land (e.g. Caspian Sea, Dead Sea) are here classed as land.

GEBCO2014 original data have been modified by hand to allow the Sea of Marmara and the Black Sea to be connected. This involved converting 1 point in the Dardanelles (strait connecting Sea of Marmara to the Mediterranean), and 3 points in the Bosphorus (strait connecting Sea of Marmara to the Black Sea), from land to ocean. For reference, these points are:

i values (counting in range 1 to 43200) = 3167, 3484, 3486, 3486 j values (counting in range 1 to 21600) = 15618,15725,15726,15729

Source

...

References

Create from GEBCO2014 by Chris Hughes

Examples

```
data(landmask8)
```

MDT_EGM08_2003_2007 *MDT_EGM08_2003_2007*

Description

Estimated MDT values at selected tide gauges. These are based on the estimated mean sea level at the gauges and the EGM 2008 geoid model. The MDT estimates are in the .... tide system.

Usage

```
data(MDT_EGM08_2003_2007)
```

Format

A data frame with 240 observations on the following 4 variables.

`PSMSLid` a numeric vector

`Lat` a numeric vector

`Lon` a numeric vector

`MDT` a numeric vector

Source

...

References

Create by Mederic Gravelle

Examples

```
data(MDT_EGM08_2003_2007)
```

MDT_eigen6c4r_2003_2007 *MDT_eigen6c4r_2003_2007*

Description

Estimated MDT values at selected tide gauges. These are based on the estimated mean sea level at the gauges and the eigen6c4r geoid model. The MSS estimates are relative to the WGS84 ellipsoid.

Usage

```
data(MDT_eigen6c4r_2003_2007)
```

Format

A data frame with 240 observations on the following 4 variables.

`PSMSLid` a numeric vector

`Lat` a numeric vector

`Lon` a numeric vector

`MDT` a numeric vector

Source

...

References

Create by Mederic Gravelle and Ole B. Andersen

Examples

```
data(MDT_eigen6c4r_2003_2007)
```

`mean2TF_AddThis` *mean2TF_AddThis*

Description

Grid to go from mean tide to tide free. The grid should be added. mean2TF_AddThis is a 2880 x 1440 matrix.

Usage

```
data("mean2TF_AddThis")
```

Format

A matrix[lon,lat] of dimension 2880 x 1440

Details

...

Source

...

References

Created by Ole B. Andersen `<oa@space.dtu.dk>`

Examples

```
data(mean2TF_AddThis)
```

`plotMDT` *Plot MDT grid*

Description

Plot MDT grid

Usage

```
plotMDT(dat, zlim, lonlim, latlim, addContour = TRUE, res = 0.125,
  conlev = 0.05, ...)
```

Arguments

`dat`	Matrix[lon,lat] with MDT values.
`zlim`	Range of the MDT values given as a Vector of length 2.
`lonlim`	Vector of length 2 with the longitude data grid limits, c(lonlim[1],lonlim[2]). The limits must be given in whole degrees
`latlim`	Vector of length 2 with the longitude data grid limits, c(lonlim[1],lonlim[2]). The limits must be given in whole degrees.
`addContour`	Bolean; To add a contour plot. Default is TRUE
`res`	Grid spacing
`conlev`	The spacing between contour lines given in meters. The default is 0.05.
`...`	Additional arguments to image.plot from fields

Details

...

`polygonizeCoast` *Helper function to getLandVal: Turns land mask matrix into polygons*

Description

Helper function to getLandVal: Turns land mask matrix into polygons

Usage

```
polygonizeCoast(mask, landVal = 0)
```

Arguments

`mask`	Matrix[lon,lat] with land mask. Land=0 (default) and water=1.
`landVal`	integeer representing the land value in the mask

Details

...

Value

Matrix[lon,lat] whith coast line ids ##'

`readncdf1var` *Read net cdf file with one variable*

Description

This function `readncdf1var` ....

Usage

```
readncdf1var(filename)
```

Arguments

`filename`	String containing the filename

Details

...

Value

Matrix var[longitude,latitude] ##'

Examples

```
## Not run:
mydat<-readncdf1var('landmask8_anyland.nc')

## End(Not run)
```

TF2mean_AddThis *TF2mean_AddThis*

Description

Grid to go from tide free to mean tide. The grid should be added. TF2mean_AddThis is a 2880 x 1440 matrix.

Usage

```
data("TF2mean_AddThis")
```

Format

A matrix[lon,lat] of dimension 2880 x 1440

Details

...

Source

...

References

Created by Ole B. Andersen <oa@space.dtu.dk>

Examples

```
data(TF2mean_AddThis)
```

`wgs2topCorr` *Function that estiamtes the height difference between WGS84 and Topex ellipsoids*

Description

Function that estiamtes the height difference between WGS84 and Topex ellipsoids

Usage

```
wgs2topCorr(phi)
```

Arguments

`phi`	The latitude in degrees.

Details

...

Value

An array with the height differences.

Index