Package ‘TCHazaRds’

August 25, 2025

Type Package

Title Tropical Cyclone (Hurricane, Typhoon) Spatial Hazard Modelling

Version 1.1.4

Date 2025-08-25

Maintainer Julian O'Grady <julian.ogrady@csiro.au>

Description Methods for generating modelled parametric Tropical Cyclone (TC) spatial hazard fields and time series output at point locations from TC tracks. R's compatibility to simply use fast 'cpp' code via the 'Rcpp' package and the wide range spatial analysis tools via the 'terra' package makes it an attractive open source environment to study 'TCs'. This package estimates TC vortex wind and pressure fields using parametric equations originally coded up in 'python' by 'TCRM' <https://github.com/GeoscienceAustralia/tcrm> and then coded up in 'Cuda' 'cpp' by 'TCwindgen' <https://github.com/CyprienBosserelle/TCwindgen>.

URL https://github.com/AusClimateService/TCHazaRds

License GPL (>= 3)

Imports Rcpp (>= 1.0.7), terra, utils, stats, geosphere, ncdf4, methods, sp, rasterVis, raster, latticeExtra

LinkingTo Rcpp

RoxygenNote 7.3.2

Encoding UTF-8

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

Language en-AU

NeedsCompilation yes

Author Julian O'Grady [aut, cre] (ORCID: <https://orcid.org/0000-0003-3552-9193>)

Repository CRAN

Date/Publication 2025-08-25 04:00:02 UTC

Contents

beta_modelsR *Compute the Exponential TC beta Profile-Curvature Parameter*

Description

Compute the Exponential TC beta Profile-Curvature Parameter

Usage

```
beta_modelsR(betaModel, vMax, rMax, cPs, eP, vFms, TClats, dPdt, rho = 1.15)
```

Arguments

betaModel	0=Powell (2005), 1=Holland (2008),2=Willoughby & Rahn (2004),3=Vickery & Wadhera (2008),4=Hubbert (1991)
vMax	maximum wind speed m/s. see vMax_modelsR
rMax	radius of maximum winds (km). see rMax_modelsR
cPs	Tropical cyclone central pressure (hPa)
eP	Background environmental pressure (hPa)
vFms	Forward speed of the storm m/s
TClats	Tropical cyclone central latitude
dPdt	rate of change in central pressure over time, hPa per hour from Holland 2008
rho	density of air

Value

exponential beta parameter

Examples

```
beta_modelsR(0,10,10,960,1013,3,-15,1)
```

DoubleHollandPressureProfile *Double Holland Pressure Profile*

Description

Pressure profile at grid points

Usage

```
DoubleHollandPressureProfile(rMax, rMax2, dP, cP, beta, R)
```

Arguments

`rMax`	radius of maximum winds in km
`rMax2`	radius of outer radial winds in km
`dP`	pressure differential, environmental less TC central pressure in hPa
`cP`	TC central pressure in hPa
`beta`	exponential term for Holland vortex
`R`	vector of distances from grid points to TC centre in km

Value

vector of pressures. //@example DoubleHollandPressureProfile(20,20,980,1.2,50)

`DoubleHollandPressureProfilePi` *Double Holland Pressure Profile Time Series*

Description

Pressure profile time series at a grid point

Usage

```
DoubleHollandPressureProfilePi(rMax, rMax2, dP, cP, beta, R)
```

Arguments

`rMax`	radius of maximum winds in km
`rMax2`	radius of outer radial winds in km
`dP`	pressure differential, environmental less TC central pressure in hPa
`cP`	TC central pressure in hPa
`beta`	exponential term for Holland vortex
`R`	vector of distances from grid points to TC centre in km

Value

vector of pressures. //@example DoubleHollandPressureProfilePi(20,20,980,1.2,50)

DoubleHollandWindProfile

Double Holland Wind Profile

Description

McConochie *et al*'s double Holland vortex model based on Cardone *et al*, 1994.This application is the Coral Sea adaptation of the double vortex model and it can also be used for concentric eye - wall configurations.

Usage

```
DoubleHollandWindProfile(f, vMax, rMax, rMax2, dP, cP, rho, beta, R)
```

Arguments

f	single coriolis parameter at the centre of TC in hz
vMax	maximum wind velocity calculation in m/s
rMax	radius of maximum winds in km
rMax2	radius of outer radial winds in km
dP	pressure differential, environmental less TC central pressure in hPa
cP	TC central pressure in hPa
rho	density of air in Kg/m3
beta	exponential term for Holland vortex
R	vector of distances from grid points to TC centre in km

Value

array with two columns for velocity and then vorticity. //@example DoubleHollandWindProfile(-1e-4,20,20,10,980,1.15,1.2,50)

DoubleHollandWindProfilePi

Double Holland Wind Profile Time Series

Description

Wind profile time series at a grid point. McConochie *et al*'s double Holland vortex model based on Cardone *et al*, 1994.This application is the Coral Sea adaptation of the double vortex model and it can also be used for concentric eye - wall configurations.

Usage

```
DoubleHollandWindProfilePi(f, vMax, rMax, rMax2, dP, cP, rho, beta, R)
```

Arguments

`f`	single coriolis parameter at the centre of TC in hz
`vMax`	maximum wind velocity calculation in m/s
`rMax`	radius of maximum winds in km
`rMax2`	radius of outer radial winds in km
`dP`	pressure differential, environmental less TC central pressure in hPa
`cP`	TC central pressure in hPa
`rho`	density of air in Kg/m3
`beta`	exponential term for Holland vortex
`R`	vector of distances from grid points to TC centre in km

Value

array with two columns for velocity and then vorticity. //@example DoubleHollandWindProfilePi(-1e-4,20,20,10,980,1.15,1.2,50)

`HollandPressureProfile` *Holland Pressure Profile*

Description

Pressure profile at grid points

Usage

```
HollandPressureProfile(rMax, dP, cP, beta, R)
```

Arguments

`rMax`	radius of maximum winds in km
`dP`	pressure differential, environmental less TC central pressure in hPa
`cP`	TC central pressure in hPa
`beta`	exponential term for Holland vortex
`R`	vector of distances from grid points to TC centre in km

Value

vector of pressures. //@example HollandPressureProfile(20,20,980,1.2,50)

HollandPressureProfilePi *Holland Pressure Profile Time Series*

Description

Pressure profile time series at a grid point.

Usage

```
HollandPressureProfilePi(rMax, dP, cP, beta, R)
```

Arguments

rMax	radius of maximum winds in km
dP	pressure differential, environmental less TC central pressure in hPa
cP	TC central pressure in hPa
beta	exponential term for Holland vortex
R	vector of distances from grid points to TC centre in km

Value

vector of pressures. //@example HollandPressureProfilePi(20,20,980,1.2,50)

HollandWindProfile *Holland Wind Profile*

Description

wind profile at grid points

Usage

```
HollandWindProfile(f, vMax, rMax, dP, rho, beta, R)
```

Arguments

f	single coriolis parameter at the centre of TC in hz
vMax	maximum wind velocity calculation in m/s
rMax	radius of maximum winds in km
dP	pressure differential, environmental less TC central pressure in hPa
rho	density of air in Kg/m3
beta	exponential term for Holland vortex
R	vector of distances from grid points to TC centre in km

Value

array with two columns for velocity and then vorticity. //@example HollandWindProfile(-1e-4,20,20,10,1.15,1.2,50)

`HollandWindProfilePi` *Holland Wind Profile Time Series*

Description

wind profile time series at a grid point

Usage

```
HollandWindProfilePi(f, vMax, rMax, dP, rho, beta, R)
```

Arguments

`f`	single coriolis parameter at the centre of TC in hz
`vMax`	maximum wind velocity calculation in m/s
`rMax`	radius of maximum winds in km
`dP`	pressure differential, environmental less TC central pressure in hPa
`rho`	density of air in Kg/m3
`beta`	exponential term for Holland vortex
`R`	vector of distances from grid points to TC centre in km

Value

array with two columns for velocity and then vorticity. //@example HollandWindProfilePi(-1e-4,20,20,10,1.15,1.2,50)

`HubbertWindField` *Hubbert Wind Field*

Description

Grid point vortex Wind field, wind vectors. Hubbert, G.D., G.J.Holland, L.M.Leslie and M.J.Manton, 1991: A Real - Time System for Forecasting Tropical Cyclone Storm Surges. *Weather and Forecasting*, **6 * *, 86 - 97

Usage

```
HubbertWindField(f, rMax, vFm, thetaFm, Rlam, V, surface)
```

Arguments

`f`	single coriolis parameter at the centre of TC in hz
`rMax`	radius of maximum winds in km
`vFm`	input forward velocity of TC
`thetaFm`	input forward direction of TC
`Rlam`	two columns for distances and direction from grid points to TC centre in km
`V`	velocity profile
`surface`	equals one if winds are reduced from the gradient level to the surface, otherwise gradient winds.

Value

array with two columns for zonal and meridional wind speed vector-components. //@example HubbertWindField(-1e-4,20,2,10,rbind(c(50,35),c(45,40)),c(20,20))

`HubbertWindFieldPi` *Hubbert Wind Field Time Series*

Description

Time series vortex Wind, wind vectors. Hubbert, G.D., G.J.Holland, L.M.Leslie and M.J.Manton, 1991: A Real - Time System for Forecasting Tropical Cyclone Storm Surges. *Weather and Forecasting*, **6 * *, 86 - 97

Usage

```
HubbertWindFieldPi(f, rMax, vFm, thetaFm, Rlam, V, surface)
```

Arguments

`f`	single coriolis parameter at the centre of TC in hz
`rMax`	radius of maximum winds in km
`vFm`	input forward velocity of TC
`thetaFm`	input forward direction of TC
`Rlam`	two columns for distances and direction from grid points to TC centre in km
`V`	velocity profile
`surface`	equals one if winds are reduced from the gradient level to the surface, otherwise gradient winds.

Value

array with two columns for zonal and meridional wind speed vector-components. //@example HubbertWindFieldPi(-1e-4,20,2,10,rbind(c(50,35),c(45,40)),c(20,20)) `inlandWindDecay` *Reduce Winds Overland*

Description

Reduce Winds Overland

Usage

```
inlandWindDecay(d, a = c(0.66, 1, 0.4))
```

Arguments

`d`	inland distance in km
`a`	three parameter of decay model a1,a2,a3

Value

a reduction factor Km

Examples

```
inlandWindDecay(10)
```

`JelesnianskiWindProfile` *Jelesnianski Wind Profile*

Description

wind profile at grid points

Usage

```
JelesnianskiWindProfile(f, vMax, rMax, R)
```

Arguments

`f`	single coriolis parameter at the centre of TC in hz
`vMax`	maximum wind velocity calculation in m/s
`rMax`	radius of maximum winds in km
`R`	vector of distances from grid points to TC centre in km

Value

array with two columns for velocity and then vorticity. //@example JelesnianskiWindProfile(-1e-4,20,20,50)

`JelesnianskiWindProfilePi`
Jelesnianski Wind Profile Time Series

Description

wind profile time series at a grid point

Usage

```
JelesnianskiWindProfilePi(f, vMax, rMax, R)
```

Arguments

`f`	single coriolis parameter at the centre of TC in hz
`vMax`	maximum wind velocity calculation in m/s
`rMax`	radius of maximum winds in km
`R`	vector of distances from grid points to TC centre in km

Value

array with two columns for velocity and then vorticity. //@example JelesnianskiWindProfilePi(-1e-4,20,20,50)

`KepertVerticalWindField`
Kepert Vertical Wind Field (u, v, Ks, w)

Description

As your KepertWindField but also computes vertical velocity w(r) = (1/r) * dQ/dr where Q(r) = r*C*Vg*(Vg + 2*vs) / (f + Vg/r + dVg/dr). Derivatives use 3-point stencils with radii r - dr, r, r + dr.

Usage

```
KepertVerticalWindField(
  rMax,
  vMax,
  vFm,
  thetaFm,
  f,
  Rlam,
  VZ,
  surface,
  dr_m = 10
)
```

Arguments

rMax	radius of maximum winds in km
vMax	maximum wind velocity in m/s
vFm	forward speed of TC (m/s)
thetaFm	forward direction of TC (deg)
f	single coriolis parameter (1/s)
Rlam	two columns: [radius_km, azimuth_deg] from grid point to TC centre
VZ	two columns: [Vi (m/s), Zi (1/s)]
surface	equals 1 for surface winds (reduced from gradient level), otherwise gradient winds.
dr_m	finite-difference step in metres (default 10 m)

Value

NumericMatrix with columns: 1) u (m/s), 2) v (m/s), 3) Ks (-), 4) w (m/s)

KepertWindField *Kepert Wind Field*

Description

Grid point vortex Wind field, wind vectors. Kepert, J., 2001: The Dynamics of Boundary Layer Jets within the Tropical Cyclone Core.Part I : Linear Theory.J.Atmos.Sci., 58, 2469 - 2484

Usage

```
KepertWindField(rMax, vMax, vFm, thetaFm, f, Rlam, VZ, surface)
```

Arguments

rMax	radius of maximum winds in km
vMax	maximum wind velocity calculation in m/s
vFm	input forward velocity of TC
thetaFm	input forward direction of TC
f	single coriolis parameter at the centre of TC in hz
Rlam	two columns for distances and Cartesian direction clocwise from the x axis from grid points to TC centre in km
VZ	array two columns velocity then vorticity
surface	equals one if winds are reduced from the gradient level to the surface, otherwise gradient winds.

Value

array with two columns for zonal and meridional wind speed vector-components. //@example KepertWindField(20,20,2,10,-1e-4,rbind(c(50,35),c(45,40)),rbind(c(20,2),c(22,3)),surface=1)

KepertWindFieldPi *Kepert Wind Field*

Description

Time series vortex Wind, wind vectors. Kepert, J., 2001: The Dynamics of Boundary Layer Jets within the Tropical Cyclone Core.Part I : Linear Theory.J.Atmospheric.Science., 58, 2469 - 2484

Usage

```
KepertWindFieldPi(rMax, vMax, vFm, thetaFm, f, Rlam, VZ, surface)
```

Arguments

`rMax`	radius of maximum winds in km
`vMax`	maximum wind velocity calculation in m/s
`vFm`	input forward velocity of TC
`thetaFm`	input forward direction of TC
`f`	single coriolis parameter at the centre of TC in hz
`Rlam`	two columns for distances and direction from grid points to TC centre in km
`VZ`	array two columns velocity then vorticity
`surface`	equals one if winds are reduced from the gradient level to the surface, otherwise gradient winds.

Value

array with two columns for zonal and meridional wind speed vector-components. //@example KepertWindField(20,20,2,10,-1e-4,rbind(c(50,35),c(45,40)),rbind(c(20,2),c(22,3)))

land_geometry *Calculate the Geometric Parameters for Terrestrial Wind*

Description

Returns geometric data to compute wind fields.

Usage

```
land_geometry(dem, inland_proximity, returnpoints = FALSE)
```

Arguments

`dem`	SpatRaster object, digital elevation model
`inland_proximity`	SpatRaster object, distance from the coast inland
`returnpoints`	Return SpatVector of points or SpatRaster

Value

SpatVector with attributes or SpatRaster

Abbreviated attribute	description	units
dem	Digital Elevation Model	m
lat	Latitude	degs
lon	Longitude	degs
slope	slope of terrain	radians
aspect	DEM aspect	radians
inlandD	distance inland from coast	m
f	Coriolis parameter	hz
dzdx	land gradient in x direction	radians
dzdy	land gradient in y direction	radians

Examples

```
require(terra)
dem <- rast(system.file("extdata/DEMs/YASI_dem.tif", package="TCHazaRds"))
land <- dem; land[land > 0] = 0
inland_proximity = distance(land,target = 0)
GEO_land = land_geometry(dem,inland_proximity)
plot(GEO_land)
```

`McConochieWindField` *McConochie Wind Field*

Description

Grid point vortex Wind field, wind vectors. McConochie, J.D., T.A.Hardy and L.B.Mason, 2004: Modelling tropical cyclone over - water wind and pressure fields. Ocean Engineering, 31, 1757 - 1782.

Usage

```
McConochieWindField(rMax, vMax, vFm, thetaFm, Rlam, V, f, surface)
```

Arguments

`rMax`	radius of maximum winds in km
`vMax`	maximum wind velocity calculation in m/s
`vFm`	input forward velocity of TC
`thetaFm`	input forward direction of TC
`Rlam`	two columns for distances and direction from grid points to TC centre in km
`V`	velocity profile
`f`	coriolis parameter at the centre of TC in hz
`surface`	equals one if winds are reduced from the gradient level to the surface, otherwise gradient winds.

Value

array with two columns for zonal and meridional wind speed vector-components. //@example McConochieWindField(-1e-4,20,2,10,rbind(c(50,35),c(45,40)),c(20,20))

`McConochieWindFieldPi` *McConochie Wind Field Time Series*

Description

Time series vortex Wind, wind vectors. McConochie, J.D., T.A.Hardy and L.B.Mason, 2004: Modelling tropical cyclone over - water wind and pressure fields. Ocean Engineering, 31, 1757 - 1782.

Usage

```
McConochieWindFieldPi(rMax, vMax, vFm, thetaFm, Rlam, V, f, surface)
```

Arguments

`rMax`	radius of maximum winds in km
`vMax`	maximum wind velocity calculation in m/s
`vFm`	input forward velocity of TC
`thetaFm`	input forward direction of TC
`Rlam`	two columns for distances and direction from grid points to TC centre in km
`V`	velocity profile
`f`	coriolis parameter at the centre of TC in hz
`surface`	equals one if winds are reduced from the gradient level to the surface, otherwise gradient winds.

Value

array with two columns for zonal and meridional wind speed vector-components. //@example McConochieWindFieldPi(-1e-4,20,2,10,rbind(c(50,35),c(45,40)),c(20,20))

NewHollandWindProfile *New Holland Wind Profile Time Series*

Description

Wind profile time series at a grid point. Holland et al. 2010. In this version, the exponent is allowed to vary linearly outside the radius of maximum wind. I.e. rather than take the square root, the exponent varies around 0.5.Currently this version does not have a corresponding vorticity profile set up in wind Vorticity, so it cannot be applied in some wind field modelling.

Usage

```
NewHollandWindProfile(f, rMax, rMax2, dP, rho, R, vMax, beta)
```

Arguments

f	single coriolis parameter at the centre of TC in hz
rMax	radius of maximum winds in km
rMax2	radius of outer 17.5ms winds in km
dP	pressure differential, environmental less TC central pressure in hPa
rho	density of air in Kg/m3
R	vector of distances from grid points to TC centre in km
vMax	maximum wind velocity calculation in m/s
beta	exponential term for Holland vortex

Value

array with two columns for velocity and then vorticity. //@example NewHollandWindProfile(-1e-4,20,20,1.15,-14,50,1.3)

NewHollandWindProfilePi *New Holland Wind Profile Time Series*

Description

Wind profile time series at a grid point. Holland et al. 2010. In this version, the exponent is allowed to vary linearly outside the radius of maximum wind. I.e. rather than take the square root, the exponent varies around 0.5.Currently this version does not have a corresponding vorticity profile set up in wind Vorticity, so it cannot be applied in some wind field modelling.

Usage

```
NewHollandWindProfilePi(f, rMax, rMax2, dP, rho, R, vMax, beta)
```

Arguments

`f`	single coriolis parameter at the centre of TC in hz
`rMax`	radius of maximum winds in km
`rMax2`	radius of outer 17ms winds in km
`dP`	pressure differential, environmental less TC central pressure in hPa
`rho`	density of air in Kg/m3
`R`	vector of distances from grid points to TC centre in km
`vMax`	maximum wind velocity calculation in m/s
`beta`	exponential term for Holland vortex

Value

array with two columns for velocity and then vorticity. //@example NewHollandWindProfilePi(-1e-4,20,20,1.15,-14,50,1.3)

`predict_rmax` *predict_rmax*

Description

Predicts the radius of maximum winds (rmax) based on the radius of 17.5 m/s winds (rMax175ms) using the Chavas and Knaff (2022) model.

Usage

```
predict_rmax(rMax175ms, vMax, TClats)
```

Arguments

`rMax175ms`	Numeric. A vector of radius of 17.5 m/s winds (in km).
`vMax`	Numeric. A vector of maximum wind speeds (m/s).
`TClats`	Numeric. A vector of latitudes of tropical cyclones (in degrees).

Value

A vector of predicted rmax values (in km).

Examples

```
rMax175ms <- c(100, 120, 140)
vMax <- c(50, 55, 60)
TClats <- c(20, 25, 30)
predict_rmax(rMax175ms, vMax, TClats)
```

`RankineWindProfilePi` *Rankine Wind Profile Time Series*

Description

wind profile time series at a grid point

Usage

```
RankineWindProfilePi(f, vMax, rMax, R)
```

Arguments

`f`	single coriolis parameter at the centre of TC in hz
`vMax`	maximum wind velocity calculation in m/s
`rMax`	radius of maximum winds in km
`R`	vector of distances from grid points to TC centre in km

Value

array with two columns for velocity and then vorticity. //@example RankineWindProfilePi(-1e-4,20,20,50)

`Rdist` *TC Distance and Direction From Output Grid Points*

Description

Grid points distance and direction to TC.

Usage

```
Rdist(Gridlon, Gridlat, TClon, TClat)
```

Arguments

`Gridlon`	vector of Grid point longitudes
`Gridlat`	vector of Grid point latitudes
`TClon`	single TC longitude
`TClat`	single TC latitude

Value

two columns for distance in km and cartesian direction in degrees, counter clockwise from the x axis. //@example Rdist(c(144,145),c(-11,-12),142,-14)

RdistPi — *TC Track Distance and Direction From Output Grid Point*

Description

Grid point time series of TC distance and direction.

Usage

```
RdistPi(Gridlon, Gridlat, TClon, TClat)
```

Arguments

`Gridlon`	single Grid point longitude
`Gridlat`	single Grid point latitude
`TClon`	vector of TC longitudes
`TClat`	vector of TC latitudes

Value

two columns for distance in km and cartesian direction in degrees, counterclockwise from the x axis. //@example RdistPi(142,-14,c(144,145),c(-11,-12))

returnBearing — *Return the Bearing for Line Segments*

Description

Return the Bearing for Line Segments

Usage

```
returnBearing(x)
```

Arguments

`x`	spatial vector with line segments (two connected points)

Value

array of bearings see geosphere::bearing, i.e the Forward direction of the storm geographic bearing, positive clockwise from true north

Examples

```
### IBTRACS HAS the WRONG BEARING!!
require(terra)
northwardTC <- vect(cbind(c(154,154),c(-26.1,-26)),"lines",crs="epsg:4283") #track line segment
easthwardTC <- vect(cbind(c(154,154.1),c(-26,-26)),"lines",crs="epsg:4283") #track line segment
southhwardTC <- vect(cbind(c(154,154),c(-26,-26.1)),"lines",crs="epsg:4283") #track line segment
westwardTC <- vect(cbind(c(154.1,154),c(-26,-26)),"lines",crs="epsg:4283") #track line segment
returnBearing(northwardTC)
returnBearing(easthwardTC)
returnBearing(southhwardTC)
returnBearing(westwardTC)
```

---

rMax175ms_solver *rMax175ms_solver*

Description

A helper function for numerically solving the radius of 17.5 m/s winds using the Chavas and Knaff (2022) model. This function is called by ‘uniroot‘ to compute the difference between the guessed and actual rmax values.

Usage

```
rMax175ms_solver(rMax175ms_m, vMax, rmax_predict_m, TClats)
```

Arguments

rMax175ms_m	Numeric. Guessed radius of 17.5 m/s winds in meters.
vMax	Numeric. Maximum wind speed (m/s).
rmax_predict_m	Numeric. Target radius of maximum winds in meters.
TClats	Numeric. Latitude of the tropical cyclone in degrees.

Value

The difference between the guessed rmax and the target rmax.

Examples

```
rMax175ms_solver(100000, 50, 36000, 20)
```

`rMax2_modelsR` *rMax2_modelsR*

Description

Numerically solves for the radius of 17.5 m/s winds (rMax175ms) using the Chavas and Knaff (2022) model and ‘uniroot‘.

Usage

```
rMax2_modelsR(rMax2Model, rMax, vMax, TClats)
```

Arguments

`rMax2Model`	TC outer radius of 17.5m/s winds model 1=’150km’, 2=Chavas and Knaff(2022)
`rMax`	Numeric. A vector of radius of maximum winds (km).
`vMax`	Numeric. A vector of maximum wind speeds (m/s).
`TClats`	Numeric. A vector of latitudes of tropical cyclone centre in degrees.

Value

A vector of predicted rMax175ms values (in km).

Examples

```
rMax <- c(30, 36, 40)
vMax <- c(50, 55, 60)
TClats <- c(20, 25, 30)
rMax2_modelsR(2,rMax, vMax, TClats)
```

`rMax_modelsR` *Compute the Tropical Cyclone Radius of Maximum Winds*

Description

Compute the Tropical Cyclone Radius of Maximum Winds

Usage

```
rMax_modelsR(
  rMaxModel,
  TClats,
  cPs,
  eP,
  R175ms = 150,
  dPdt = NULL,
  vFms = NULL,
  rho = 1.15,
  vMax = NULL
)
```

Arguments

`rMaxModel`	0=Powell et.al.(2005),1=McInnes et.al.(2014),2=Willoughby & Rahn (2004), 3=Vickery & Wadhera (2008), 4=Takagi & Wu (2016), 5 = Chavas & Knaff (2022).
`TClats`	Tropical cyclone central latitude (nautical degrees)
`cPs`	Tropical cyclone central pressure (hPa)
`eP`	Background environmental pressure (hPa)
`R175ms`	radius of 17.5m/s wind speeds (km)
`dPdt`	rate of change in central pressure over time, hPa per hour from Holland 2008
`vFms`	Forward speed of the storm m/s
`rho`	density of air
`vMax`	maximum wind speed m/s. see `vMax_modelsR`

Value

radius of maximum winds (km)

Examples

```
rMax_modelsR(0,-14,950,1013,200,0,0,1.15)
```

`TCHazaRdsWindField` *Compute the Wind and Pressure Spatial Hazards Field Associated with TCs Single Time Step.*

Description

Compute the Wind and Pressure Spatial Hazards Field Associated with TCs Single Time Step.

Usage

```
TCHazaRdsWindField(
  GEO_land,
  TC,
  paramsTable,
  return_vars = c("Pr", "Uw", "Vw", "Sw", "Dw", "R", "lam"),
  returnWaves = NULL
)
```

Arguments

`GEO_land`	SpatVector or dataframe hazard geometry generated with land_geometry
`TC`	SpatVector or data.frame of Tropical cyclone track parameters for a single time step.
`paramsTable`	Global parameters to compute TC Hazards.
`return_vars`	character(). Variables to return. Default: core winds/pressure. Options: c("Pr","Uw","Vw","Sw","Dw","R
`returnWaves`	DEPRECATED. Use return_vars including 'Hs0','Tp0','Dp0' instead

Value

SpatRaster with the following attributes

abbreviated variable	description	units
P	Atmospheric pressure	hPa
Uw	Meridional wind speed	m/s
Vw	Zonal wind speed	m/s
Ww	Vertical wind speed	m/s
Sw	Wind speed	m/s
Dw	The direction from which wind originates	deg clockwise from true north
Hs0	Deep water significant wave height	m
Tp0	Deep water Peak wave period	s
Dp0	The peak direction in which wave are heading	deg clockwise from true north
R	The radial distance to the TC centre	km
lam	The radial direction to the TC centre	deg

Examples

```
require(terra)
dem <- rast(system.file("extdata/DEMs/YASI_dem.tif", package="TCHazaRds"))
land <- dem; land[land > 0] = 0
inland_proximity = distance(land,target = 0)
GEO_land = land_geometry(dem,inland_proximity)

TCi = vect(cbind(c(154,154),c(-26.1,-26)),"lines",crs="epsg:4283") #track line segment
TCi$PRES = 950
TCi$RMAX = 40
TCi$VMAX = 60
```

```
TCi$B = 1.4
TCi$ISO_TIME = "2022-10-04 20:00:00"
TCi$LON = geom(TCi)[1,3]
TCi$LAT = geom(TCi)[1,4]
TCi$STORM_SPD = perim(TCi)/(3*3600) #m/s
TCi$thetaFm = 90-returnBearing(TCi)
#OR
TC <- vect(system.file("extdata/YASI/YASI.shp", package="TCHazaRds"))
TC$PRES <- TC$BOM_PRES
TCi = TC[47]
plot(dem);lines(TCi,lwd = 4,col=2)

paramsTable = read.csv(system.file("extdata/tuningParams/defult_params.csv",package = "TCHazaRds"))
#calculate the wind hazard
HAZ = TCHazaRdsWindField(GEO_land,TCi,paramsTable)
plot(HAZ)

#require(rasterVis) #pretty spatial vector plot
#ats = seq(0, 80, length=9)
#UV = as(c(HAZ["Uw"],HAZ["Vw"]),"Raster") #need to convert back to raster
#vectorplot(UV, isField='dXY', col.arrows='white', aspX=0.002,aspY=0.002,at=ats ,
#colorkey=list( at=ats), par.settings=viridisTheme)
```

`TCHazaRdsWindFields` *Compute the Wind and Pressure Spatial Hazards Field Associated with TC track.*

Description

Compute the Wind and Pressure Spatial Hazards Field Associated with TC track.

Usage

```
TCHazaRdsWindFields(
  outdate = NULL,
  GEO_land,
  TC,
  paramsTable,
  outfile = NULL,
  overwrite = FALSE,
  returnWaves = NULL,
  return_vars = c("Pr", "Uw", "Vw", "Sw", "Dw")
)
```

Arguments

`outdate`	array of POSITx date times to linearly interpolate TC track
`GEO_land`	SpatVector or dataframe hazard geometry generated with land_geometry

`TC`	SpatVector of Tropical cyclone track parameters for a single time step
`paramsTable`	Global parameters to compute TC Hazards
`outfile`	character. Output netcdf filename
`overwrite`	TRUE/FALSE, option to overwrite outfile
`returnWaves`	DEPRECATED. Use return_vars including 'Hs0','Tp0','Dp0' instead
`return_vars`	character(). Variables to return. Default: core winds/pressure. Options: c("Pr","Uw","Vw","Sw","Dw","R

Value

SpatRasterDataset with the following attributes.

abbreviated variable	description	units
P	Atmospheric pressure	hPa
Uw	Meridional wind speed	m/s
Vw	Zonal wind speed	m/s
Ww	Vertical wind speed	m/s
Sw	Wind speed	m/s
Dw	The direction from which wind originates	deg clockwise from true north
Hs0	Deep water significant wave height	m
Tp0	Deep water Peak wave period	s
Dp0	The peak direction in which wave are heading	deg clockwise from true north
R	The radial distance to the TC centre	km
lam	The radial direction to the TC centre	deg

Examples

```
require(terra)
dem <- rast(system.file("extdata/DEMs/YASI_dem.tif", package="TCHazaRds"))
land <- dem; land[land > 0] = 0
inland_proximity = distance(land,target = 0)
GEO_land = land_geometry(dem,inland_proximity)

TCi = vect(cbind(c(154,154),c(-26.1,-26)),"lines",crs="epsg:4283") #track line segment
TCi$PRES = 950
TCi$RMAX = 40
TCi$VMAX = 60
TCi$B = 1.4
TCi$ISO_TIME = "2022-10-04 20:00:00"
TCi$LON = geom(TCi)[1,3]
TCi$LAT = geom(TCi)[1,4]
TCi$STORM_SPD = perim(TCi)/(3*3600) #m/s
TCi$thetaFm = 90-returnBearing(TCi)
#OR
TC <- vect(system.file("extdata/YASI/YASI.shp", package="TCHazaRds"))
TC$PRES <- TC$BOM_PRES
plot(dem);lines(TC,lwd = 4,col=2)

paramsTable = read.csv(system.file("extdata/tuningParams/defult_params.csv",package = "TCHazaRds"))
```

```
#calculate the wind hazard

outdate = seq(strptime(TC$ISO_TIME[44],"%Y-%m-%d %H:%M:%S",tz="UTC"),
              strptime(TC$ISO_TIME[46],"%Y-%m-%d %H:%M:%S",tz="UTC"),
              3600*3)
HAZi = TCHazaRdsWindFields(outdate=outdate,GEO_land=GEO_land,TC=TC,paramsTable=paramsTable)
plot(min(HAZi$Pr))
```

`TCHazaRdsWindProfile` *Compute the Wind and Pressure Spatial Hazards Profile Associated with TCs Single Time Step.*

Description

Compute the Wind and Pressure Spatial Hazards Profile Associated with TCs Single Time Step.

Usage

```
TCHazaRdsWindProfile(GEO_land, TC, paramsTable)
```

Arguments

GEO_land	SpatVector or dataframe hazard geometry generated with land_geometry
TC	SpatVector or data.frame of Tropical cyclone track parameters for a single time step.
paramsTable	Global parameters to compute TC Hazards.

Value

SpatRaster with the following attributes

abbreviated attribute	description	units
P	Atmospheric pressure	hPa
Uw	Meridional wind speed	m/s
Vw	Zonal wind speed	m/s
Sw	Wind speed	m/s
Dw	Wind direction	deg clockwise from true north

Examples

```
require(terra)
dem <- rast(system.file("extdata/DEMs/YASI_dem.tif", package="TCHazaRds"))
land <- dem; land[land > 0] = 0
inland_proximity = distance(land,target = 0)
GEO_land = land_geometry(dem,inland_proximity)
```

```
TCi = vect(cbind(c(154,154),c(-26.1,-26)),"lines",crs="epsg:4283") #track line segment
TCi$PRES = 950
TCi$RMAX = 40
TCi$VMAX = 60
TCi$B = 1.4
TCi$ISO_TIME = "2022-10-04 20:00:00"
TCi$LON = geom(TCi)[1,3]
TCi$LAT = geom(TCi)[1,4]
TCi$STORM_SPD = perim(TCi)/(3*3600) #m/s
TCi$thetaFm = 90-returnBearing(TCi)
#OR
TC <- vect(system.file("extdata/YASI/YASI.shp", package="TCHazaRds"))
TC$PRES <- TC$BOM_PRES
TCi = TC[47]
TCi$thetaFm = 90-returnBearing(TCi)

#extract a profile/transect at right angles (90 degrees) from the TC heading/bearing direction
pp <- TCProfilePts(TC_line = TCi,bear=TCi$thetaFm+90,length =100,step=1)
#plot(dem);lines(TCi,lwd = 4,col=2)
#points(pp)
GEO_land_v = extract(GEO_land,pp,bind=TRUE,method = "bilinear")

paramsTable = read.csv(system.file("extdata/tuningParams/defult_params.csv",package = "TCHazaRds"))
#calculate the wind hazard
HAZ = TCHazaRdsWindProfile(GEO_land_v,TCi,paramsTable)
#plot(HAZ$radialdist,HAZ$Sw,type="l",xlab = "Radial distance [km]",ylab = "Wind speed [m/s]");grid()
#plot(HAZ,"Sw",type="continuous")
```

`TCHazaRdsWindTimeSereies` *Compute the Wind Hazards Associated Over the Period of a TCs Event at one Given Location*

Description

Compute the Wind Hazards Associated Over the Period of a TCs Event at one Given Location

Usage

```
TCHazaRdsWindTimeSereies(
  outdate = NULL,
  GEO_land = NULL,
  TC,
  paramsTable,
  returnWaves = FALSE
)
```

Arguments

`outdate`	array of POSITx date times to linearly interpolate TC track,optional.
`GEO_land`	dataframe hazard geometry generated with land_geometry
`TC`	SpatVector of Tropical cyclone track parameters
`paramsTable`	Global parameters to compute TC Hazards.
`returnWaves`	Return ocean wave parameters (default = FALSE)

Details

The function calculates wind speed and direction time series from a tropical cyclone track using various wind profile models.

Value

list() containing a timeseries

abbreviated attribute	description	units
date	POSIX data time object of TC or outdate if provided	as.POSIX
P	Atmospheric pressure	hPa
Uw	Meridional wind speed	m/s
Vw	Zonal wind speed	m/s
Sw	Wind speed	m/s
R	distance to TC centre	m
rMax	radius of maximum wind	km
vMax	TC maximum velocity	m/s
b	TC wind profile exponent	-
CP	TC central Pressure	hPa
dPdt	change in TC CP per hour	hPa/hr
vFm	velocity of TC forward motion	m/s
Hs0	Deep water significant wave height	m
Tp0	Deep water Peak wave period	s
Dp0	The peak direction in which wave are heading	deg clockwise from true north.

Examples

```
GEO_land = data.frame(dem=0,lons = 147,lats=-18,f=-4e-4,inlandD = 0)

require(terra)
TCi <- vect(system.file("extdata/YASI/YASI.shp", package="TCHazaRds"))
TCi$PRES <- TCi$BOM_PRES

paramsTable = read.csv(system.file("extdata/tuningParams/defult_params.csv",package = "TCHazaRds"))
HAZts = TCHazaRdsWindTimeSereies(GEO_land=GEO_land,TC=TCi,paramsTable = paramsTable)
main =  paste(TCi$NAME[1],TCi$SEASON[1],"at",GEO_land$lons,GEO_land$lats)
#with(HAZts,plot(date,Sw,format = "%b-%d %H",type="l",main = main,ylab = "Wind speed [m/s]"))
```

TCpoints2lines *Convert Points to Line Segments*

Description

This function converts a set of point geometries into line segments. The input vector must be a set of points, and the function will draw line segments between consecutive points. An additional point is extrapolated from the last two points to ensure the final segment is complete.

Usage

```
TCpoints2lines(pts_v)
```

Arguments

pts_v	A ‘SpatVector‘ of points (from the ‘terra‘ package).

Value

A ‘SpatVector‘ containing line geometries created from the input points.

Examples

```
library(terra)
# Create example points
pts <- vect(matrix(c(1, 1, 2, 2, 3, 3), ncol=2), type="points")
# Convert points to line segments
TClines <- TCpoints2lines(pts)
```

TCProfilePts *Transect points from a origin through a point or with a bearing and to the opposite side.*

Description

Transect points from a origin through a point or with a bearing and to the opposite side.

Usage

```
TCProfilePts(
  TC_line,
  Through_point = NULL,
  bear = NULL,
  length = 200,
  step = 2
)
```

Arguments

`TC_line`	origin of the transect
`Through_point`	a point to pass through
`bear`	the bearing
`length`	the length of the transect in Km
`step`	the spacing of the transect in Km

Value

spatial vector of transect profile points with distances in Km (negative for left hand side)

Examples

```
require(terra)
TCi <- vect(cbind(c(154.1,154),c(-26.1,-26)),"lines",crs="epsg:4283") #track line segment
TCi$PRES <- 950
TCi$RMAX <- 40
TCi$B <- 1.4
TCi$RMAX2 <- 90
TCi$ISO_TIME <- "2022-10-04 20:00:00"
TCi$LON <- geom(TCi)[1,3]
TCi$LAT <- geom(TCi)[1,4]
TCi$STORM_SPD <- perim(TCi)/(3*3600) #m/s
TCi$thetaFm <- 90-returnBearing(TCi)
#Through_point <- isd[isd$OID==isdsi]
pp <- TCProfilePts(TC_line = TCi,Through_point=NULL,bear=TCi$thetaFm+90,length =100,step=10)
plot(pp,"radialdist",type="continuous")
lines(TCi,col=2)
```

`TCvectInterp` *Temporally Interpolate Along a Tropical Cyclone Track And Compute Along-Track Parameters*

Description

Temporally Interpolate Along a Tropical Cyclone Track And Compute Along-Track Parameters

Usage

```
TCvectInterp(outdate = NULL, TC, paramsTable)
```

Arguments

`outdate`	POSIX times to be interpolated to. The output date in "YYYY-MM-DD" format. Default is NULL.
`TC`	SpatVector of Tropical cyclone track parameters
`paramsTable`	Global parameters to compute TC Hazards.

Value

SpatVector of Tropical cyclone track parameters

Examples

```
require(terra)
TCi <- vect(system.file("extdata/YASI/YASI.shp", package="TCHazaRds"))
TCi$PRES <- TCi$BOM_PRES
TCi$PRES[is.na(TCi$PRES)] = 1010
outdate = seq(strptime(TCi$ISO_TIME[1],"%Y-%m-%d %H:%M:%S",tz="UTC"),
strptime(rev(TCi$ISO_TIME)[1],"%Y-%m-%d %H:%M:%S",tz="UTC"),3600)
paramsTable = read.csv(system.file("extdata/tuningParams/defult_params.csv",package = "TCHazaRds"))
TCii = TCvectInterp(outdate = outdate,TC=TCi,paramsTable = paramsTable)
```

`tunedParams` *Update Parameter List to Calibrated Values*

Description

Update Parameter List to Calibrated Values

Usage

```
tunedParams(
  paramsTable,
  infile = system.file("extdata/tuningParams/QLD_modelSummaryTable.csv", package =
    "TCHazaRds")
)
```

Arguments

`paramsTable`	Global parameters to compute TC Hazards.
`infile`	File containing tuning parameters in a .csv. Default for QLD calibration.

Value

list of params with updated tuning wind parameters.

Examples

```
paramsTable <- read.csv(system.file("extdata/tuningParams/defult_params.csv",package = "TCHazaRds"))

tunedParams(paramsTable)
```

update_Track *Calculate Additional TC Parameters, and temporally Interpolate Along a Tropical Cyclone Track*

Description

Calculate Additional TC Parameters, and temporally Interpolate Along a Tropical Cyclone Track

Usage

```
update_Track(
  outdate = NULL,
  indate,
  TClons,
  TClats,
  vFms,
  thetaFms,
  cPs,
  rMaxModel,
  vMaxModel,
  betaModel,
  rMax2Model,
  eP,
  rho = NULL,
  RMAX,
  VMAX,
  B,
  RMAX2
)
```

Arguments

`outdate`	POSIX times to be interpolated to
`indate`	POSIX input times
`TClons`	input central TC longitude
`TClats`	input central TC latitude
`vFms`	input forward velocity of TC
`thetaFms`	input forward direction
`cPs`	central pressure
`rMaxModel`	empirical model for radius of maximum wind calculation (rMax in km)
`vMaxModel`	empirical model for maximum wind velocity calculation (vMax in m/s)
`betaModel`	empirical model for TC shape parameter beta (dimensionless Beta)
`rMax2Model`	empirical model for radius of outer 17.5ms wind calculation (rMax2 in km)
`eP`	background environmental pressure (hPa)

rho	air density
RMAX	If params rMaxModel value is NA, use input TC$RMAX
VMAX	If params rMaxModel value is NA, use input TC$VMAX
B	If params rMaxModel value is NA, use input TC$B
RMAX2	If params rMax2Model value is NA, use input TC$RMAX2

Value

list of track data inclining the rMax vMax and Beta.

Examples

```
paramsTable <- read.csv(system.file("extdata/tuningParams/defult_params.csv",package = "TCHazaRds"))
params <- array(paramsTable$value,dim = c(1,length(paramsTable$value)))
colnames(params) <- paramsTable$param
params <- data.frame(params)
require(terra)
TCi <- vect(system.file("extdata/YASI/YASI.shp", package="TCHazaRds"))
TCi$PRES <- TCi$BOM_PRES
TCi$RMAX <- TCi$BOM_RMW*1.852 #convert from nautical miles to km
TCi$VMAX <- TCi$BOM_WIND*1.94 #convert from knots to m/s
TCi$B <- 1.4
TCi$RMAX2 <- 150
t1 <- strptime("2011-02-01 09:00:00","%Y-%m-%d %H:%M:%S", tz = "UTC") #first date in POSIX format
t2 <- strptime(rev(TCi$ISO_TIME)[1],"%Y-%m-%d %H:%M:%S", tz = "UTC") #last date in POSIX format
outdate <- seq(t1,t2,"hour") #array sequence from t1 to t2 stepping by "hour"
# defult along track parameters are calculated
TCil = update_Track(outdate = outdate,
                    indate = strptime(TCi$ISO_TIME,"%Y-%m-%d %H:%M:%S", tz = "UTC"),
                    TClons = TCi$LON,
                    TClats = TCi$LAT,
                    vFms=TCi$STORM_SPD,
                    thetaFms=TCi$thetaFm,
                    cPs=TCi$PRES,
                    rMaxModel=params$rMaxModel,
                    vMaxModel=params$vMaxModel,
                    betaModel=params$betaModel,
                    rMax2Model = params$rMaxModel,
                    eP = params$eP,
                    rho = params$rhoa,
                    RMAX = TCi$RMAX,
                    VMAX = TCi$VMAX,
                    B = TCi$B,
                    RMAX2 = TCi$RMAX2
                    )
# 'observed' along tack parameters are calculated (#Model = NA)
TCil = update_Track(outdate = outdate,
                    indate = strptime(TCi$ISO_TIME,"%Y-%m-%d %H:%M:%S", tz = "UTC"),
                    TClons = TCi$LON,
                    TClats = TCi$LAT,
                    vFms=TCi$STORM_SPD,
```

```
thetaFms=TCi$thetaFm,
cPs=TCi$PRES,
rMaxModel=NA,
vMaxModel=NA,
betaModel=NA,
rMax2Model = NA,
eP = params$eP,
rho = params$rhoa,
RMAX = TCi$RMAX,
VMAX = TCi$VMAX,
B = TCi$B,
RMAX2 = TCi$RMAX2
)
```

---

vMax_modelsR *Compute the Tropical Cyclone Maximum Wind Speeds*

Description

Compute the Tropical Cyclone Maximum Wind Speeds

Usage

```
vMax_modelsR(
  vMaxModel,
  cPs,
  eP,
  vFms = NULL,
  TClats = NULL,
  dPdt = NULL,
  beta = 1.3,
  rho = 1.15
)
```

Arguments

vMaxModel	0=Arthur (1980),1=Holland (2008),2=Willoughby & Rahn (2004).3=Vickery & Wadhera (2008),4=Atkinson and Holliday (1977)
cPs	Tropical cyclone central pressure (hPa)
eP	Background environmental pressure (hPa)
vFms	Forward speed of the storm m/s
TClats	Tropical cyclone central latitude
dPdt	rate of change in central pressure over time, hPa per hour from Holland 2008
beta	exponential term for Holland vortex
rho	density of air

Value

maximum wind speed m/s.

Examples

```
vMax_modelsR(vMaxModel=1,cPs=950,eP=1010,vFms = 1,TClats = -14,dPdt = .1)
```

Index