Package 'wmm'

May 8, 2026

Type Package

Title World Magnetic Model

Version 1.1.1

Maintainer Will Frierson <will.frierson@gmail.com>

Description Calculate magnetic field at a given location and time according to the World Magnetic Model (WMM). Both the main field and secular variation components are returned. This functionality is useful for physicists and geophysicists who need orthogonal components from WMM. Currently, this package supports annualized time inputs between 2000 and 2025. If desired, users can specify which WMM version to use, e.g., the original WMM2015 release or the recent out-of-cycle WMM2015 release. Methods used to implement WMM, including the Gauss coefficients for each release, are described in the following publications: Chulliat et al (2020) <doi:10.25923/ytk1-yx35>, Chulliat et al (2019) <doi:10.25921/xhr3-0t19>, Chulliat et al (2015) <doi:10.7289/V5TB14V7>, Maus et al (2010) <https://www.ngdc.noaa.gov/geomag/WMM/data/WMMReports/WMM2010_Report.pdf>, McLean et al (2004) <https://www.ngdc.noaa.gov/geomag/WMM/data/WMMReports/TRWMM_2005.pdf>, and Macmillian et al (2000) <https://www.ngdc.noaa.gov/geomag/WMM/data/WMMReports/wmm2000.pdf>.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 2.10)

RoxygenNote 6.1.1

Suggests testthat (>= 2.0.1), data.table (>= 1.12.2)

URL https://github.com/wfrierson/wmm

BugReports https://github.com/wfrierson/wmm/issues

Language en-US

NeedsCompilation no

Author Will Frierson [aut, cre]

Repository CRAN

Date/Publication 2021-09-06 04:10:02 UTC

Contents

`.CalcLegendre` *Compute Associated Legendre Functions Given Sequence of (degree, order) Indices*

Description

Procedure that computes the associated Legendre function, $P_{n,m}(\mu)$, given a sequence of (degree, order) indices and function argument μ. This is computed via a closed-form equation.

Usage

```
.CalcLegendre(mu)
```

Arguments

`mu`	Function argument to $P_{n,m}(\mu)$

Details

The underlying equation used is:

$P(x,n,m) = (-1)^m * 2^n * (1-x^2)^(m/2) * sum(form <= k <= n : k!/(k-m)! * x^(k-m) * choose(n,k) * choose((n+k-1

.CalcLegendreComponents

Compute Legendre Components

Description

Function that computes the components of the associated Legendre function, $P_{n,m}(\mu)$, only dependent on (degree, order) indices. This function is only used to precompute values.

Usage

```
.CalcLegendreComponents(n, m)
```

Arguments

n	degree of associated Legendre function
m	order of associated Legendre function

Details

The underlying equation used is:

$P(x, n, m) = (-1)^m * 2^n * (1-x^2)^(m/2) * sum(form <= k <= n : k!/(k-m)! * x^(k-m) * choose(n, k) * choose((n+k-1

.CalcPolynomialComponents

Calculate Polynomial Components for Associated Legendre Function

Description

Function that computes the polynomial components of mu that are paired with the output of .CalcLegendreComponents to create the indiviudal components of the associated Legendre function, $P_{n,m}(\mu)$.

Usage

```
.CalcPolynomialComponents(mu)
```

Arguments

mu	Function argument to $P_{n,m}(\mu)$

Details

The underlying equation used is:

$P(x, n, m) = (-1)^m * 2^n * (1-x^2)^(m/2) * sum(form <= k <= n : k!/(k-m)! * x^(k-m) * choose(n, k) * choose((n+k-1

.CalculateGaussCoef *Lookup Table for Gauss coefficients g & h*

Description

Find Gauss coefficient $g_{n,m}(t)$ consistent with the World Magnetic Model.

Usage

```
.CalculateGaussCoef(t, t0, wmmVersion = "derived")
```

Arguments

t	Annualized date time. E.g., 2015-02-01 = (2015 + 32/365) = 2015.088
t0	Annualized reference time associated with t
wmmVersion	String representing WMM version to use. Must be consistent with time and one of the following: 'derived', 'WMM2000', 'WMM2005', 'WMM2010', 'WMM2015', 'WMM2015v2', 'WMM2020'. Default 'derived' value will infer the latest WMM version consistent with time.

Value

vector of Gauss coefficients, $g_{n,m}(t)$ and $h_{n,m}(t)$

.CalculateGeocentricFieldSum *Calculate sum of geocentric field components*

Description

Calculate sum of geocentric field components

Usage

```
.CalculateGeocentricFieldSum(legendreTable, gaussCoef, radius, lon, latGC,
  deltaLatitude)
```

Arguments

legendreTable	data.table modified by .CalculateSchmidtLegendreDerivative
gaussCoef	Gauss coefficients as calculated by .CalculateGaussCoef
radius	Radius of curvature of prime vertical at given geodetic latitude
lon	GPS longitude
latGC	GPS latitude, geocentric
deltaLatitude	(Geocentric Latitude - Geodetic Latitude) in decimal degrees

.CalculateMagneticElements

Calculate Expected Magnetic Elements from WMM2020

Description

Calculate the magnetic elements (i.e., horizontal intensity, total intensity, inclination, declination, and their secular variation) for given magnetic orthogonal components

Usage

```
.CalculateMagneticElements(orthComps)
```

Arguments

orthComps	named list containing magnetic orthogonal components

Value

Expected magnetic components from WMM2020. list

.CalculateMagneticField

Calculate Expected Magnetic Field from WMM2020

Description

Calculate the magnetic field for a given location and time using the fitted spherical harmonic model from the 2020 World Magnetic Model.

Usage

```
.CalculateMagneticField(lon, latGD, latGC, radius, time,
  wmmVersion = "derived")
```

Arguments

lon	GPS longitude
latGD	GPS latitude, geodetic
latGC	GPS latitude, geocentric
radius	Radius of curvature of prime vertical at given geodetic latitude
time	Annualized date time. E.g., 2015-02-01 = (2015 + 32/365) = 2015.088
wmmVersion	String representing WMM version to use. Must be consistent with time and one of the following: 'derived', 'WMM2005', 'WMM2010', 'WMM2015', 'WMM2015v2', 'WMM2020'. Default 'derived' value will infer the latest WMM version consistent with time.

Value

Expected magnetic field from WMM2020, $m^{WMM}_{\lambda_t,\varphi_t,h_t,t}$. `list`

`.CalculateRadiusCurvature`

Radius of curvature of prime vertical

Description

Calculate radius of curvature of prime vertical at given geodetic latitude.

Usage

```
.CalculateRadiusCurvature(latitudeGD)
```

Arguments

`latitudeGD`	Geodetic latitude in decimal degrees

Value

Radius of curvature of prime vertical at given geodetic latitude

---

`.CheckBlackoutZone` *Check if given horizontal intensity triggers a blackout zone*

Description

Check if given horizontal intensity triggers a blackout zone

Usage

```
.CheckBlackoutZone(h)
```

Arguments

`h`	horizontal intensity, `numeric`

Value

`warning` if warranted

.CheckVersionWMM *Check if given time is consistent with available WMM versions*

Description

Check if given time is consistent with available WMM versions

Usage

```
.CheckVersionWMM(t, wmmVersion)
```

Arguments

t	Annualized date time. E.g., 2015-02-01 = (2015 + 32/365) = 2015.088
wmmVersion	String representing WMM version to use. Must be consistent with time and one of the following: 'derived', 'WMM2000', 'WMM2005', 'WMM2010', 'WMM2015', 'WMM2015v2'.

.ConvertGeocentricToGeodeticFieldComponents *Geocentric Coordinates to Geodetic Coordinates*

Description

Convert Geocentric Coordinates to Geodetic Coordinates.

Usage

```
.ConvertGeocentricToGeodeticFieldComponents(xGeocentric, yGeocentric,
  zGeocentric, deltaLat)
```

Arguments

xGeocentric	X-coordinate in geocentric system
yGeocentric	Y-coordinate in geocentric system
zGeocentric	Z-coordinate in geocentric system
deltaLat	(Geocentric Latitude - Geodetic Latitude) in decimal degrees

Value

Vector of length 3 representing geodetic coordinates consistent with given geocentric data

.ConvertGeodeticToGeocentricGPS *Convert from Geodetic to Geocentric Coordinates*

Description

Convert geodetic coordinates to geocentric coordinates

Usage

```
.ConvertGeodeticToGeocentricGPS(latitudeGD, height)
```

Arguments

latitudeGD	Geodetic latitude in decimal degrees
height	Height in meters above ellipsoid (not mean sea level)

Value

List with first element as geocentric latitude in decimal degrees and second element as geocentric radius

.DeriveVersionInfo *Derive WMM version based on given time*

Description

Derive WMM version based on given time

Usage

```
.DeriveVersionInfo(t)
```

Arguments

t	Annualized date time. E.g., 2015-02-01 = (2015 + 32/365) = 2015.088

Value

List of reference year and compatible WMM versions inferred from time.

`GetMagneticFieldWMM` *Calculate Expected Magnetic Field from WMM*

Description

Function that takes in geodetic GPS location and annualized time, and returns the expected magnetic field from WMM.

Usage

```
GetMagneticFieldWMM(lon, lat, height, time, wmmVersion = "derived")
```

Arguments

`lon`	GPS longitude
`lat`	GPS latitude, geodetic
`height`	GPS height in meters above ellipsoid
`time`	Annualized date time. E.g., 2015-02-01 = (2015 + 32/365) = 2015.088; optionally an object (length 1) of class 'POSIXt' or 'Date'
`wmmVersion`	String representing WMM version to use. Must be consistent with `time` and one of the following: 'derived', 'WMM2000', 'WMM2005', 'WMM2010', 'WMM2015', 'WMM2015v2', 'WMM2020'. Default 'derived' value will infer the latest WMM version consistent with `time`.

Value

`list` of calculated main field and secular variation vector components in nT and nT/yr, resp. The magnetic element intensities (i.e., horizontal and total intensities, h & f) are in nT and the magnetic element angles (i.e., inclination and declination, i & d) are in degrees, with their secular variation in nT/yr and deg/yr, resp.: `x`, `y`, `z`, `xDot`, `yDot`, `zDot`, `h`, `f`, `i`, `d`, `hDot`, `fDot`, `iDot`, `dDot`

Examples

```
GetMagneticFieldWMM(
   lon = 240,
   lat = -80,
   height = 1e5,
   time = 2022.5,
   wmmVersion = 'WMM2020'
)

## Expected output
# x = 5814.9658886215 nT
# y = 14802.9663839328 nT
# z = -49755.3119939183 nT
# xDot = 28.0381961827 nT/yr
# yDot = 1.3970624624 nT/yr
```

```
# zDot = 85.6309533031 nT/yr
# h = 15904.1391483373 nT
# f = 52235.3588449608 nT
# i = -72.27367 deg
# d = 68.55389 deg
# hDot = 11.5518244235 nT/yr
# fDot = -78.0481471753 nT/yr
# iDot = 0.04066726 deg/yr
# dDot = -0.09217566 deg/yr

## Calculated output
#$x
#[1] 5814.966

#$y
#[1] 14802.97

#$z
#[1] -49755.31

#$xDot
#[1] 28.0382

#$yDot
#[1] 1.397062

#$zDot
#[1] 85.63095

#$h
#[1] 15904.14

#$f
#[1] 52235.36

#$i
#[1] -72.27367

#$d
#[1] 68.55389

#$hDot
#[1] 11.55182

#$fDot
#[1] -78.04815

#$iDot
#[1] 0.04066726

#$dDot
#[1] -0.09217566
```

wmm *wmm: R Implementation of World Magnetic Model*

Description

The wmm package calculates magnetic field at a given location and time according to the World Magnetic Model.

WMM functions

This package has 1 exported function, GetMagneticFieldWMM, which returns a list of:

- Main field and secular variation vector components in nT and nT/yr, resp.
- Magnetic element intensities (i.e., horizontal and total intensities, h & f) in nT with their secular variation in nT/yr
- Magnetic element angles (i.e., inclination and declination, i & d) in degrees with their secular variation in deg/yr

GetMagneticFieldWMM(lambda_t, phi_t, h_t, t) = (x, y, z, xDot, yDot, zDot, h, f, i, d, hDot, fDot, iDot, dDot)

Acknowledgments

Thanks to:

- The WMM team past, present, and future for making the Gauss coefficients public domain
- Alex Breeze for tech reviewing the original version of this code, years ago

Index