

Package ‘RFOC’

July 21, 2025

Type Package

Title Graphics for Spherical Distributions and Earthquake Focal Mechanisms

Version 3.4-10

Date 2023-09-06

Depends R ($\geq$ 3.1.0)

Imports RPMG, GEOmap, RSEIS, MASS, fields

Author Jonathan M. Lees [aut, cre],
Keehoon Kim [ctb]

Maintainer Jonathan M. Lees <jonathan.lees@unc.edu>

Description Graphics for statistics on a sphere, as applied to geological fault data, crystallography, earthquake focal mechanisms, radiation patterns, ternary plots and geographical/geological maps. Non-double couple plotting of focal spheres and source type maps are included for statistical analysis of moment tensors.

License GPL ($\geq$ 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2023-09-06 21:52:35 UTC

Contents

RFOC-package	4
addmecpoints	6
addPT	7
addPTarrows	7
addsmallcirc	8
AlongGreat	9
alpha95	10
AXpoint	12
bang	13
Beachfoc	14
Bfocvec	15

BOXarrows3D	17
cirtics	19
CONVERTSDR	20
cross.prod	21
CROSSL	22
doNonDouble	23
EApont	24
egl	25
fancyarrows	27
faultplane	28
FixDip	29
flipnodal	30
foc.color	31
foc.icolor	32
FOCangles	33
focleg	34
focpoint	35
forcerighthand	36
getCMT	37
GetRake	39
GetRakeSense	40
getUWfocs	42
HAMMERprojXY	43
hudson.net	44
hudson.plot	45
imageP	47
imageSCALE	48
imageSH	49
imageSV	50
inverseTAPE	51
jimbo	52
JMAT	52
justfocXY	53
KAMCORN	55
lowplane	56
m2tk	57
makeblock3D	58
makenet	59
MapNonDouble	60
mc2cart	63
mijedr	64
MomentDist	65
MRake	66
net	67
nipXY	68
nodalLines	70
normal.fault	71
pcirc	72

pglyph3D	72
phong3D	74
PKAM	76
plotfoc	77
plotmanyfoc	79
plotMEC	80
PlotPlanes	81
PlotPTsmooth	82
PlotTernfoc	84
PLTcirc	86
pnet	87
polyfoc	88
Pradfoc	89
Preflect	90
prepFOCS	91
printMEC	92
PROJ3D	92
PTaxes	94
PTXY2	95
qpoint	97
radiateP	98
radiateSH	99
radiateSV	100
radP	101
radSH	102
radSV	103
rakelegend	105
readCMT	106
RectDense	108
REFLECT	109
rotateFoc	110
Rotfocphi	111
RotTP	112
ROTX	113
rotx3	114
ROTY	115
roty3	116
ROTZ	117
rotz3	118
SDRfoc	119
ShadowCLVD	120
SHradfoc	122
SourceType	123
spherefocgeo	124
spline.arrow	126
StrikeDip	128
strikeslip.fault	130
SVradfoc	131

TapeBase	132
TapePlot	133
TEACHFOC	134
ternfoc.point	135
ternfocgeo	136
testrightHAND	138
thrust.fault	139
tk2uv	140
to.spherical	141
TOCART.DIP	142
tocartL	143
TOSPHERE	144
TOSPHERE.DIP	145
tosphereL	146
TP2XYZ	147
TRANmat	148
Vmoments	149
widdenMoments	149
Wnet	150
Wpoint	151
xsecmanyfoc	152
Z3Darrow	154

Index	156
--------------	------------

RFOC-package

Calculates and plot Earthquake Focal Mechanisms

Description

Graphics for statistics on a sphere, as applied to geological fault data, crystallography, earthquake focal mechanisms, radiation patterns, ternary plots and geographical/geological maps. Given strike-dip-rake or a set of fault planes, focal planes, RFOC creates structures for manipulating and plotting earthquake focal mechanisms as individual plots or distributed spatially maps.

RFOC can be used for analysis of plane orientation, geologic structure, distribution of stress and strain analyses.

Details

Visualize focal mechanisms in a number of modes, including: beachball plots, radiation plots, fault planes and ternary diagrams. Shows spatial distribution of spherically distributed data.

Author(s)

Jonathan M. Lees Maintainer: Jonathan M. Lees <jonathan.lees@unc.edu>

References

J. M. Lees. Geotouch: Software for three and four dimensional GIS in the earth sciences. *Computers and Geosciences*, 26(7):751–761, 2000.

K.~Aki and P.~G. Richards. *Quantitative seismology*. University Science Books, Sausalito, Calif., 2nd edition, 2002.

Snyder, John P., 1987, Map Projections-a working manual, USGS-Professional Paper, 383p.

C. Frohlich. Triangle diagrams: ternary graphs to display similarity and diversity of earthquake focal mechanisms. *Physics of the Earth and Planetary Interiors*, 75:193-198, 1992.

See Also

RSEIS, GEOMap, zoeppritz

Examples

```
##### plot one focal mechanism:
M = SDRfoc(-25, 34, 16,u = FALSE, ALIM = c(-1, -1, +1, +1), PLOT=TRUE)
```

```
##### plot many P-axes:
paz = rnorm(100, mean=297, sd=100)
pdip = rnorm(100, mean=52, sd=20)
net()
focpoint(paz, pdip, col='red', pch=3, lab="", UP=FALSE)
```

```
#####
```

```
#### Show many Focal mechanisms on a plot:
```

```
Z1 = c(159.33,51.6,206,18,78,
161.89,54.5,257,27,133,
170.03,53.57,-44,13,171,
154.99,50.16,-83,19,-40,
151.09,47.15,123,23,-170,
176.31,51.41,-81,22,122,
153.71,46.63,205,28,59,
178.39,51.21,-77,16,126,
178.27,51.1,-86,15,115,
177.95,51.14,-83,25,126,
178.25,51.18,215,16,27
)
```

```
MZ = matrix(Z1, ncol=5, byrow=TRUE)
```

```
plot(MZ[,1], MZ[,2], type='n', xlab="LON", ylab="LAT", asp=1)
```

```
for(i in 1:length(MZ[,1]))
{
paste(MZ[i,3], MZ[i,4], MZ[i,5])
}
```

```

MEC = SDRfoc(MZ[i,3], MZ[i,4], MZ[i,5], u=FALSE, ALIM=c(-1,-1, +1, +1), PLOT=FALSE)
fcol = foc.color(foc.icolor(MEC$rake1), pal=1)
justfocXY(MEC, x=MZ[i,1], y =MZ[i,2] , focsiz=0.5, fcol =fcol , fcolback = "white", xpd = TRUE)

}

```

addmecpoints

Add points to Focal Mech

Description

Add a standard set of points to a Focal Mechanism

Usage

```
addmecpoints(MEC, pch = 5)
```

Arguments

MEC	MEC structure list
pch	plotting character

Value

Graphical Side effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

SDRfoc, focpoint

Examples

```

MEC= SDRfoc(12,34,-120)
addmecpoints(MEC)

```

addPT	<i>Add P-T Axis to focal plot</i>
-------	-----------------------------------

Description

Add Pressure and tension Axes to focal mechanism

Usage

```
addPT(MEC, pch = 5)
```

Arguments

MEC	MEC structure
pch	plotting character

Value

Graphical Side Effect

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

addPTarrows

Examples

```
MEC =SDRfoc(65,25,13, u=FALSE, ALIM=c(-1,-1, +1, +1), PLOT=FALSE)
Beachfoc(MEC)
addPT(MEC, pch = 5)
```

addPTarrows	<i>Add fancy 3D arrows</i>
-------------	----------------------------

Description

Illustrate Pressure and Tension axis on Focal Plot using 3D arrows

Usage

```
addPTarrows(MEC)
```

Arguments

MEC Mechanism Structure

Value

Graphical Side Effects

Note

This function looks better when plotting the upper hemisphere

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

focpoint, BOXarrows3D,Z3Darrow

Examples

```
MEC = SDRfoc(65,25,13, u=TRUE, ALIM=c(-1,-1, +1, +1), PLOT=TRUE)

addPTarrows(MEC)
```

addsmallcirc

Small Circle on Stereonet

Description

Calculate and plot small circle on Stereo net at arbitrary azimuth, orientation and conical angle

Usage

```
addsmallcirc(az, iang, alphadeg, BALL.radius = 1, N = 100, add = TRUE, ...)
```

Arguments

az	Azimuth of axis
iang	angle of dip, degrees
alphadeg	width of cone in degrees
BALL.radius	size of sphere
N	Number of points to calculate
add	logical, TRUE=add to existing plot
...	graphical parameters

Details

Given the azimuth and dip of a vector, plot the small circle around the pole with conical angle alphadeg

Value

LIST:

x x-coordinates

y y-coordinates

Note

alphadeg is the radius of the conic projection

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

net

Examples

```
net()
addsmallcirc(65, 13, 20, BALL.radius = 1, N = 100, add = TRUE)
addsmallcirc(165, 73, 5.6, BALL.radius = 1, N = 100, add = TRUE)
```

AlongGreat

Get Points Along Great Circle

Description

Using a Starting LAT-LON, return points along an azimuth

Usage

```
AlongGreat(LON1, LAT1, km1, ang, EARTHRADE = 6371)
```

Arguments

LON1	Longitude, point
LAT1	Latitude, point
km1	Kilometers in direction ang
ang	Direction from North
EARTHRAID	optional earth radius, default = 6371

Details

Returns LAT-LON points along a great circle, so many kilometers away in a specified direction

Value

LIST:

lat	Latitude, destination point
lon	Longitude, destination point
distdeg	distance in degrees
distkm	distance in km

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

Examples

```
london = c(51.53333, -0.08333333)
AlongGreat(london[2], london[1], 450, 56)
```

alpha95

95 percent confidence for Spherical Distribution

Description

Calculates conical projection angle for 95% confidence bounds for mean of spherically distributed data.

Usage

```
alpha95(az, iang)
```

Arguments

az	vector of azimuths, degrees
iang	vector of dips, degrees

Details

Program calculates the cartesian coordinates of all poles, sums and returns the resultant vector, its azimuth and length (R). For N points, statistics include:

$$K = \frac{N - 1}{N - R}$$

$$S = \frac{81^\circ}{\sqrt{K}}$$

$$\kappa = \frac{\log(\frac{\epsilon_1}{\epsilon_2})}{\log(\frac{\epsilon_2}{\epsilon_3})}$$

$$\alpha_{95} = \cos^{-1} \left[1 - \frac{N - R}{R} \left(20^{\frac{1}{N-1}} - 1 \right) \right]$$

where ϵ 's are the relevant eigenvalues of matrix MAT and angles are in degrees.

Value

LIST:

Ir	resultant inclination, degrees
Dr	resultant declination, degrees
R	resultant sum of vectors, normalized
K	K-dispersion value
S	spherical variance
Alph95	95% confidence angle, degrees
Kappa	log ratio of eigenvectors
E	Eigenvectors
MAT	matrix of cartesian vectors

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

Davis, John C., 2002, Statistics and data analysis in geology, Wiley, New York, 637p.

See Also

addsmallicirc

Examples

```
paz = rnorm(100, mean=297, sd=10)
pdip = rnorm(100, mean=52, sd=8)
ALPH = alpha95(paz, pdip)

##### draw stereonet
net()
##### add points
focpoint(paz, pdip, col='red', pch=3, lab="", UP=FALSE)
##### add 95 percent confidence bounds
addsmalldisc(ALPH$Dr, ALPH$Ir, ALPH$Alph95, BALL.radius = 1, N = 25,
add = TRUE, lwd=1, col='blue')

##### second example:
paz = rnorm(100, mean=297, sd=100)
pdip = rnorm(100, mean=52, sd=20)
ALPH = alpha95(paz, pdip)

net()
focpoint(paz, pdip, col='red', pch=3, lab="", UP=FALSE)

addsmalldisc(ALPH$Dr, 90-ALPH$Ir, ALPH$Alph95, BALL.radius = 1, N = 25,
add = TRUE, lwd=1, col='blue')
```

AXpoint

Extract Axis pole on Stereonet

Description

Interactive extract axis point on Stereonet

Usage

```
AXpoint(UP = TRUE, col=2, n=1)
```

Arguments

UP	logical, TRUE=upper hemisphere
col	plotting color
n	maximum number to locate, default=unlimited

Details

Program uses locator to create a vector of poles. Points outside the focal sphere ($r > 1$) are ignored. If n is missing, locator continues until stopped (middle mouse in linux, stop in windows).

Value

phiang	azimuth angle, degrees
dip	dip angle, degrees
x	x-coordinate of cartesian vector
y	y-coordinate of cartesian vector
z	z-coordinate of cartesian vector
gx	x-coordinate of prjection
gy	y-coordinate of prjection

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

locator, qpoint, EApoint

Examples

```
##### this is interactive
## Not run:
net()
Z = AXpoint(UP = TRUE)
## click in steronet
Z

## End(Not run)
```

bang

Angle between two 2D normalized vectors

Description

Calculates the angle between two 2D normalized vectors using dot and cross product

Usage

```
bang(x1, y1, x2, y2)
```

Arguments

x1	x coordinate of first normalized vector
y1	y coordinate of first normalized vector
x2	x coordinate of second normalized vector
y2	y coordinate of second normalized vector

Details

The sign of angle is determined by the sign of the cross product of the two vectors.

Value

angle in radians

Note

Vectors must be normalized prior to calling this routine. Used mainly for vectors on the unit sphere.

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

Examples

```
v1 = c(5,3)
v2 = c(6,1)

a1 = c(5,3)/sqrt(v1[1]^2+v1[2]^2)
a2 = c(6,1)/sqrt(v2[1]^2+v2[2]^2)

plot(c(0, v1[1],v2[1] ) , c(0, v1[2],v2[2]), type='n', xlab="x", ylab="y" )
text(c(v1[1],v2[1]) , c(v1[2],v2[2]), labels=c("v1", "v2"), pos=3, xpd=TRUE)

arrows(0, 0, c(v1[1],v2[1] ), c(v1[2],v2[2]))

B = 180*bang(a1[1], a1[2], a2[1], a2[2])/pi
title(paste(sep=" ", "Angle from V1 to V2=",format(B, digits=2)) )
```

Beachfoc

Plot a BeachBall Focal Mechanism

Description

Plots a focal mechanism in beachball style

Usage

```
Beachfoc(MEC, fcol = gray(0.9), fcolback = "white", ALIM = c(-1, -1, +1, +1))
```

Arguments

MEC	Mechanism Structure
fcol	color for the filled portion of the beachball
fcolback	color for the background portion of the beachball, default='white'
ALIM	Bounding box for beachball, default=c(-1, -1, +1, +1)

Details

Beachfoc is run after MEC is set using SDRfoc. Options for plotting the beachball in various modes are controlled by flags set in MEC

Value

Used for its graphical side effect

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

References

K. Aki and P. G. Richards. *Quantitative seismology*. University Science Books, Sausalito, Calif., 2nd edition, 2002. Keiiti Aki, Paul G. Richards. ill. ; 26 cm.

See Also

CONVERTSDR, SDRfoc, justfocXY

Examples

```
MEC = SDRfoc(65,25,13, u=FALSE, ALIM=c(-1,-1, +1, +1), PLOT=TRUE)

Beachfoc(MEC, fcol=MEC$fcol, fcolback="white")
```

Bfocvec

Angles for Ternary plot

Description

Calculates Angles for determining ternary distribution of faults based on P-T axis orientation.

Usage

```
Bfocvec(Paz, Pdip, Taz, Tdip)
```

Arguments

Paz	vector of azimuths, degrees
Pdip	vector of dips, degrees
Taz	vector of azimuths, degrees
Tdip	vector of dips, degrees

Details

This calculation is based on Froelich's paper.

Value

LIST:

Bdip	azimuths, degrees
Baz	dips, degrees

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

C. Frohlich. Triangle diagrams: ternary graphs to display similarity and diversity of earthquake focal mechanisms. *Physics of the Earth and Planetary Interiors*, 75:193-198, 1992.

See Also

ternfoc.point

Examples

```
Msdr = CONVERTSDR(55.01, 165.65, 29.2 )
MEC = MRake(Msdr$M)
MEC$UP = FALSE
az1 = Msdr$M$az1
dip1 = Msdr$M$d1
az2 = Msdr$M$az2
dip2 = Msdr$M$d2
BBB = Bfocvec(az1, dip1, az2, dip2)
V = ternfoc.point(BBB$Bdip, Msdr$M$pd, Msdr$M$td )
```

`BOXarrows3D`*Create a 3D Arrow structure*

Description

Create and project and plot 3D arrows with viewing Matrix.

Usage

```
BOXarrows3D(x1, y1, z1, x2, y2, z2, aglyph = NULL, Rview = ROTX(0),  
col = grey(0.5), border = "black", len = 0.7, basethick = 0.05,  
headlen = 0.3, headlip = 0.02)
```

Arguments

<code>x1</code>	x-coordinates of base of arrows
<code>y1</code>	y-coordinates of base of arrows
<code>z1</code>	z-coordinates of base of arrows
<code>x2</code>	x-coordinates of head of arrows
<code>y2</code>	y-coordinates of head of arrows
<code>z2</code>	z-coordinates of head of arrows
<code>aglyph</code>	glyph structure, default is Z3Darrow
<code>Rview</code>	Viewing matrix
<code>col</code>	fill color
<code>border</code>	Border color
<code>len</code>	Length
<code>basethick</code>	thickness of the base
<code>headlen</code>	thickness of the head
<code>headlip</code>	width of the overhanging lip

Details

Arrows point from base to head.

Value

Used for graphical side effects.

Note

Any 3D glyph structure can be used

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

Z3Darrow

Examples

```
## Not run:
#### animate 10 random arrow vectors

L = list(x1 = runif(10, min=-2, max=2),
        y1 = runif(10, min=-2, max=2),
        z1=runif(10, min=-4, max=4),
        x2 = runif(10, min=-2, max=2),
        y2 = runif(10, min=-2, max=2),
        z2=runif(10, min=-4, max=4)
        )
headlen = .3
len = .7
basethick = 0.05
headlip = .02
aglyph = Z3Darrow(len = len , basethick =basethick , headlen =headlen , headlip=headlip )

r1 = 8
theta = seq(from=0, to=2*360, length=200)
mex = r1*cos(theta*pi/180)
mey = r1*sin(theta*pi/180)
mez = seq(from=r1, to =0 , length=length(mex))
## mez=rep(r1, length=length(mex))

angz = atan2(mey, mex)*180/pi
angx = atan2(sqrt(mex^2+mey^2), mez)*180/pi
pal=c("red", "blue", "green")

## aglyph = gblock

for(j in 1:length(angz))
{
  Rview = ROTZ(angz[j])
  plot(c(-4,4), c(-4,4), type='n', asp=1); grid()

  BOXarrows3D(L$x1,L$y1,L$z1, L$x2,L$y2,L$z2, aglyph=aglyph, Rview=Rview, col=pal)

  Sys.sleep(.1)
}

## End(Not run)
```

cirtics	<i>Draw circular ticmarks</i>
---------	-------------------------------

Description

Draw circular ticmarks

Usage

```
cirtics(r = 1, dr = 0.02, dang = 10, ...)
```

Arguments

r	radius
dr	length of tics
dang	angle between tics
...	graphical parameters

Value

graphical side effects

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

Examples

```
phi = seq(from = 0, to = 2 * pi, length = 360)
x = cos(phi)
y = sin(phi)
plot(x, y, col = 'blue', asp = 1, type = 'l')
cirtics(r = 1, dr = 0.02, dang = 10, col = 'red')
```

 CONVERTSDR

Convert Strike-Dip-Rake to MEC structure

Description

Takes Strike-Dip-Rake and creates planes and pole locations for MEC structure

Usage

CONVERTSDR(strike, dip, rake)

Arguments

strike	angle, degrees, strike of down dip directin
dip	angle, degrees, dip is measured from the horizontal NOT from the NADIR
rake	angle, degrees

Details

input is strike dip and rake in degrees

Value

LIST:

strike	strike
dipdir	dip
rake	rake
F	list(az, dip) of F-pole
G	list(az, dip) of G-pole
U	list(az, dip) of U-pole
V	list(az, dip) of V-pole
P	list(az, dip) of P-pole
T	list(az, dip) of T-pole
M	list(az1=0, d1=0, az2=0, d2=0, uaz=0, ud=0, vaz=0, vd=0, paz=0, pd =0, taz=0, td=0)

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

BeachFoc

Examples

```
s=65
d=25
r=13

mc = CONVERTSDR(s,d,r )
```

cross.prod	<i>Vector Cross Product</i>
------------	-----------------------------

Description

Vector Cross Product with list as arguments and list as values

Usage

```
cross.prod(B, A)
```

Arguments

B	list of x,y,z
A	list of x,y,z

Value

LIST	
x, y, z	vector of cross product

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

RSEIS::xprod

Examples

```
B1 = list(x=4, y=9, z=2)
B2 = list(x=2,y=-5,z=4)

cross.prod(B1, B2)
```

CROSSL*Vector Cross Product*

Description

returns cross product of two vectors in list format

Usage

```
CROSSL(A1, A2)
```

Arguments

A1	list x,y,z
A2	list x,y,z

Value

List	
x, y, z	input vector
az	azimuth, degrees
dip	dip, degrees

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

RSEIS::xprod

Examples

```
A1 = list(x=1,y=2, z=3)
A2 = list(x=12,y=-2, z=-5)

N = CROSSL(A1, A2)
```

doNonDouble

*Plot Non-double Couple Moment***Description**

Plot Non-double Couple Moment

Usage

```
doNonDouble(moments, sel = 1, col=rgb(1, .75, .75))
```

Arguments

moments	list of moments: seven elements. See details.
sel	integer vector, index of moments to plot
col	color, either a single color, rgb, or a color palette

Details

Plot, sequentially the moments using the CLVD (non-double couple component. The first element of the list is the integer index of the event. The next six elements are the moments in the following order, c(Mxx, Myy, Mzz, Mzy, Mxz, Mxy) .

If the data is in spherical coordinates, one must switch the sign of the Mrp and Mtp components, so:

```
Mrr = Mzz
Mtt = Mxx
Mpp = Myy
Mrt = Mxz
Mrp = -Myz
Mtp = -Mxy
```

Value

Side effects

Note

If events are read in using spherical rather than cartesian coordinates need a conversion:

```
Mrr = Mzz
Mtt = Mxx
Mpp = Myy
Mrt = Mxz
Mrp = -Myz
Mtp = -Mxy
```

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

Ekstrom, G.; Nettles, M. and DziewoDski, A. The Global CMT Project 2004-2010: centroid-moment tensors for 13,017 earthquakes Physics of the Earth and Planetary Interiors, 2012.

See Also

MapNonDouble, ShadowCLVD, angles, nodalLines, PTaxes

Examples

```
mo = list(n=1, m1=1.035675e+017, m2=-1.985852e+016,
m3=-6.198052e+014, m4=1.177936e+017, m5=-7.600627e+016,
m6=-3.461405e+017)

moments = cbind(mo$n, mo$m1, mo$m2, mo$m3, mo$m4, mo$m5, mo$m6)

doNonDouble(moments)
```

EApoint

Equal-area point stereonet

Description

Interactive locator to calculate x,y orientation, dip coordinates and plots on an equalarea stereonet

Usage

EApoint()

Details

Used for returning a set of strike/dip angles on Equal-area stereonet plot.

Value

LIST:

phi	orientation, degrees
iang	angle of dip, degrees
x	x-coordinate
y	y-coordinate

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

qpoint, focpoint

Examples

```
##### this is interactive

### collect points with locator()
## Not run:
net()
eps = EApoint()

### plot results
net()
qpoint(eps$phi , eps$iang)

## End(Not run)
```

egl

Tungurahua Cartesian Moment Tensors

Description

Cartesian moment tensors from Tungurahua Volcano, Ecuador

Usage

```
data(egl)
```

Format

A list of 84 moment tensors, each element consists of: lam1, lam2, lam3, vec1, vec2,vec3, ratio, force.

Source

See below

References

Kim, K., Lees, J.M. and Ruiz, M., (2014) Source mechanism of Vulcanian eruption at Tungurahua Volcano, Ecuador, derived from seismic moment tensor inversions, *J. Geophys. Res.*, February, 2014. Vol. 119(2): pp. 1145-1164.

Examples

```

data(egl)

typ1=c(2,4,7,12,13,16,17,18,19,20,24,25,26,27,
28,29,30,31,33,34,35,36,37,38,40,43,50,
59,62,73,74,77,8,79,80,81,83,84)
typ12=c(5,6,8,9,10,11,14,15,22,42,46,47,48,49,
51,52,53,54,55,56,57,58,60,61,63,72,82)

evtns=1:84

par(mfrow=c(1,2))
T1 = TapeBase()
TapePlot(T1)

for(i in 1:length(egl))
{
i1 = egl[[i]]

E1 = list(values=c(i1$lam1, i1$lam2, i1$lam3),
vectors = cbind(i1$vec1, i1$vec2, i1$vec3))

testrightHAND(E1$vectors)

E1$vectors = forcerighthand(E1$vectors)

mo=sort(E1$values,decreasing=TRUE)
# M=sum(mo)/3
# Md=mo-M
h = SourceType(mo)
h$dip = 90-h$phi

h1 = HAMMERprojXY(h$dip*pi/180, h$lam*pi/180)

if(i %in% typ1) { col="red" }else{col="blue" }
points(h1$x, h1$y, pch=21, bg=col )

}

par(mai=c(0,0,0,0))
hudson.net()
for(i in 1:length(typ1))
{
egv=egl[[typ1[i]]]
m=c(egv$lam1, egv$lam2, egv$lam3)
col='red'
hudson.plot(m=m,col=col)
}

for(i in 1:length(typ12))

```

```
{
  egv=egl[[typ12[i]]]
  m=c(egv$lam1, egv$lam2, egv$lam3)
  col='blue'
  hudson.plot(m=m, col=col, lwd=2)
}
```

fancyarrows

Make fancy arrows

Description

Create and plot fancy arrows. Aspect ratio must be set to 1-1 for these arrows to plot correctly.

Usage

```
fancyarrows(x1, y1, x2, y2, thick = 0.08,
            headlength = 0.4, headthick = 0.2, col = grey(0.5),
            border = "black")
```

Arguments

x1	x tail coordinate
y1	y tail coordinate
x2	x head coordinate
y2	y head coordinate
thick	thickness of arrow
headlength	length of head
headthick	thickness of head
col	fill color
border	color of border

Value

Graphical side effects.

Note

fancyarrows only work if the aspect ratio is set to 1. See example below.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

TEACHFOC

Examples

```

    thick = 0.01; headlength = 0.2; headthick = 0.1

x = runif(10, -1, 1)
y = runif(10, -1, 1)

##### MUST set asp=1 here
plot(x,y, asp=1)

fancyarrows(rep(0, 10) , rep(0, 10) ,x, y,
thick =thick , headlength = headlength,
headthick =headthick)

```

faultplane

fault plane projection on focal sphere

Description

given azimuth and dip of fault mechanism, calculate and plot the fault plane.

Usage

```
faultplane(az, dip, col = par("col"), PLOT = TRUE, UP = FALSE, lwd=2, lty=1, ...)
```

Arguments

az	degrees, strike of the plane (NOT down dip azimuth)
dip	degrees, dip from horizontal
col	color for line
PLOT	option for adding to plot
UP	upper or lower hemisphere
lwd	Line Width
lty	Line Type
...	graphical parameters

Details

Azimuth is the strike in degrees, not the down dip azimuth as described in other routines.

Value

list of points along fault plane

x coordinates on focal sphere

y coordinates on focal sphere

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

Beachfoc

Examples

```
gcol='black'
border='black'
ndiv=36
phi = seq(0,2*pi, by=2*pi/ndiv);
x = cos(phi);
y = sin(phi);

plot(x,y, type='n', asp=1)
lines(x,y, col=border)
lines(c(-1,1), c(0,0), col=gcol)
lines(c(0,0), c(-1,1), col=gcol)

faultplane(65, 34)
```

FixDip

Fix Dip Angle

Description

Fix az, dip angles so they fall in correct quadrant.

Usage

FixDip(A)

Arguments

List:

A **az** azimuthm angle, degrees
dip dip angle, degrees

Details

Quadrants are determined by the sine and cosine of the dip angle: $co = \cos(dip)$ $si = \sin(dip)$
 $quad[co \geq 0 \ \& \ si \geq 0] = 1$ $quad[co < 0 \ \& \ si \geq 0] = 2$ $quad[co < 0 \ \& \ si < 0] = 3$ $quad[co \geq 0 \ \& \ si < 0]$
 $= 4$

Value

List:

az	azimuthm angle, degrees
dip	dip angle, degrees

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

RPMG::fmod

Examples

```
B = list(az=231, dip = -65)
```

```
FixDip(B)
```

flipnodal

Flip Nodal Fault Plane

Description

Switch a focal mechanism so the auxilliary plane is the nodal plane.

Usage

```
flipnodal(s1, d1, r1)
```

Arguments

s1	Strike
d1	Dip
r1	Rake

Details

Fufunction is used for orienting a set of fault planes to line up according to a geologic interpretation.

Value

List:

s1	Strike
d1	Dip
r1	Rake

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

Examples

```
s=65
d=25
r=13

mc = CONVERTSDR(s,d,r )

mc2 = flipnodal(s, d, r)
```

foc.color

*Get color of Focal Mechansim***Description**

Based on the rake angle, focal styles are assigned an index and assigned a color by foc.color

Usage

```
foc.color(i, pal = 0)
```

Arguments

i	index to list of focal rupture styles
pal	vector of colors

Details

Since the colors used by focal programs are arbitrary, this routines allows one to change the coloring scheme easily.

foc.icolor returns an index that is used to get the color associated with that style of faulting

Value

Color for plotting, either a name or HEX RGB

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

foc.icolor

Examples

```
fcolors=c("DarkSeaGreen", "cyan1", "SkyBlue1", "RoyalBlue", "GreenYellow", "orange", "red")
foc.color(3, fcolors)
```

foc.icolor

Get Fault Style

Description

Use Rake Angle to determine style of faulting

Usage

```
foc.icolor(rake)
```

Arguments

rake degrees, rake angle of fault plane

Details

The styles are determined by the rake angle

strikeslip $\text{abs}(\text{rake}) \leq 15.0$ or $\text{abs}((180.0 - \text{abs}(\text{rake}))) \leq 15.0$

rev-obl strk-slp $(\text{rake} \geq 15.0 \text{ and } \text{rake} < 45)$ or $(\text{rake} \geq 135 \text{ and } \text{rake} < 165)$

oblique reverse $(\text{rake} \geq 45.0 \text{ and } \text{rake} < 75)$ or $(\text{rake} \geq 105 \text{ and } \text{rake} < 135)$

reverse $\text{rake} \geq 75.0 \text{ and } \text{rake} < 105.0$

norm-oblq strk-slp $(\text{rake} < -15.0 \text{ and } \text{rake} \geq -45)$ or $(\text{rake} < -135 \text{ and } \text{rake} \geq -165)$

oblq norm $(\text{rake} < -45.0 \text{ and } \text{rake} \geq -75)$ or $(\text{rake} < -105 \text{ and } \text{rake} \geq -135)$

normal $\text{rake} < -75.0 \text{ and } \text{rake} \geq -105$

Value

index (1-6)

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

foc.color

Examples

```
foc.icolor(25)
```

FOCangles

Angles for focal planes

Description

Angles for focal planes

Usage

```
FOCangles(m)
```

Arguments

m moment tensor

Details

Used in MapNonDouble and doNonDouble

Value

vector of 6 angles, 3 for each plane

Note

Lower Hemisphere.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

MapNonDouble, doNonDouble, PTaxes, nodalLines

Examples

```
mo = list(n=1, m1=1.035675e+017, m2=-1.985852e+016,
          m3=-6.198052e+014, m4=1.177936e+017,
          m5=-7.600627e+016, m6=-3.461405e+017)

moments = cbind(mo$n, mo$m1, mo$m2, mo$m3, mo$m4, mo$m5, mo$m6)

di = dim(moments)
number.of.events = di[1]
moment_11 = moments[,2]
moment_22 = moments[,3]
moment_33 = moments[,4]
moment_23 = moments[,5]
moment_13 = moments[,6]
moment_12 = moments[,7]

i = 1
m=matrix( c(moment_11[i],moment_12[i],moment_13[i],
            moment_12[i],moment_22[i],moment_23[i],
            moment_13[i],moment_23[i],moment_33[i]), ncol=3, byrow=TRUE)

angles.all = FOCangles(m)
print(angles.all)
```

focleg	<i>Fault style descriptor</i>
--------	-------------------------------

Description

Get character string describing type of fault from its style index

Usage

```
focleg(i)
```

Arguments

i index to vector of focal styles

Value

character string used for setting text on plots

Note

String of characters:

STRIKESLIP Strike slip fault

REV-OBL STRK-SLP Reverse Oblique strike-slip fault

REVERSE Reverse fault

NORM-OBLQ STRKS LP Normal Oblique strike-slip fault

OBLQ NORM Oblique Normal fault

NORMAL Normal fault

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

foc.icolor, foc.color

Examples

```
focleg(2)
```

focpoint

add point on focal sphere

Description

Add points on equal-area focal plot

Usage

```
focpoint(az1, dip1, col = 2, pch = 5, lab = "", cex=1, UP = FALSE, PLOT = TRUE, ...)
```

Arguments

az1	degrees, azimuth angle
dip1	degrees, dip angle
col	color
pch	plot character for point
lab	text label for point
cex	Character Size
UP	upper or lower hemisphere
PLOT	logical, PLOT=TRUE add points to current plot
...	graphical parameters

Value

List of x,y coordinates on the plot

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

Beachfoc, addmecpoints

Examples

```
### create focal mech
ALIM=c(-1,-1, +1, +1)
s=65
d=25
r=13
mc = CONVERTSDR(s,d,r )
MEC = MRake(mc$M)
MEC$UP = FALSE
MEC$icol = foc.icolor(MEC$rake1)
MEC$ileg = focleg(MEC$icol)
MEC$fcol = foc.color(MEC$icol)
MEC$CNVRG = NA
MEC$LIM = ALIM

### plot focal mech
Beachfoc(MEC, fcol=MEC$fcol, fcolback="white")

### now add the F anf G axes
focpoint(MEC$F$az, MEC$F$dip, pch=5, lab="F", UP=MEC$UP)
focpoint(MEC$G$az, MEC$G$dip, pch=5, lab="G", UP=MEC$UP)
```

forcerighthand

Force Right-Hand System

Description

Force Right-Hand System

Usage

forcerighthand(U)

Arguments

U 3 by 3 matrix

Details

Flip vectors so they form a right handed system

Value

matrix

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

testrightHAND

Examples

```
Mtens = c(-0.412, 0.084, 0.328 ,0.398, -1.239, 1.058)
M1 = matrix(c(Mtens[1], Mtens[4], Mtens[5], Mtens[4], Mtens[2],
Mtens[6], Mtens[5],Mtens[6], Mtens[3]), ncol=3, nrow=3, byrow=TRUE)
E1 = eigen(M1)
testrightHAND(E1$vectors)

E1$vectors = forcerighthand(E1$vectors)

testrightHAND(E1$vectors)
```

getCMT

Read CMT

Description

Read and reformat CMT solutions downloaded from the web.

Usage

```
getCMT(fn, skip=1)
```

Arguments

fn	character file name
skip	number of lines to skip (e.g. header)

Details

Data can be extracted from web site: <http://www.globalcmt.org/CMTsearch.html>

The file must be cleaned prior to scanning - on download from the web site there are extra lines on top and bottom of file. Delete these. Leave one line on the top that describes the columns. Data is separated by blanks. The files have a mixture of dates - some with 7 component dates (YYMMDD) and others with 14 components YYYYMMDDHHMM these are read in separately. Missing hours and minutes are set to zero.

Value

list of CMT solution data:

lon	lon of epicenter
lat	lat of epicenter
str1	strike of fault plane
dip1	dip of fault plane
rake1	rake of fault plane
str2	strike of auxilliary plane
dip2	dip of auxilliary plane
rake2	rake of auxilliary plane
sc	scale?
iexp	exponent?
name	name, includes the date
Elat	exploding latitude, set to lat initially
Elon	exploding longitude, set to lon initially
jd	julian day
yr	year
mo	month
dom	day of month

Note

Use ExplodeSymbols or explode to get new locations for expanding the plotting points.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

<http://www.globalcmt.org/CMTsearch.html>

G. Ekstrom. Rapid earthquake analysis utilizes the internet. Computers in Physics, 8:632-638, 1994.

See Also

ExplodeSymbols, spherefocgeo, ternfocgeo

Examples

```
## Not run:

g = getCMT("/home/lees/aleut.cmt")

pg = prepFOCS(g)

plot(range(pg$LONS), range(pg$LATS), type = "n", xlab = "LON",
      ylab = "LAT", asp = 1)

for (i in 1:length(pg$LATS)) {
  mc = CONVERTSDR(g$str1[i], g$dip1[i], g$rake1[i])
  MEC <- MRake(mc$M)
MEC$UP = FALSE
  Fcol <- foc.color(foc.icolor(MEC$rake1), pal = 1)
  justfocXY(MEC, x = pg$LONS[i], y = pg$LATS[i], focsiz = 0.4,
            fcol = Fcol, xpd = FALSE)
}

## End(Not run)
```

GetRake

Calculate Rake angles

Description

Calculates rake angles for fault and auxilliary planes

Usage

```
GetRake(az1, dip1, az2, dip2, dir)
```

Arguments

az1	azimuth in degrees of fault plane 1
dip1	dip in degrees of fault plane 1
az2	azimuth in degrees of auxilliary plane 2
dip2	dip in degrees of auxilliary plane 2
dir	polarity

Details

uses output of CONVERTSDR or MEC structure

Value

list of angles for fault plane and auxillary plane

```
az1, dip1, rake1, dipaz1
    strike, dip rake and downdip direction for plane 1
az2, dip2, rake2, dipaz2
    strike, dip rake and downdip direction for plane 2
```

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

GetRakeSense, CONVERTSDR, Beachfoc, justfocXY

Examples

```
GetRake(345.000000, 25.000000, 122.000000, 71.000000, 1)
```

GetRakeSense	<i>Get Rake Sense</i>
--------------	-----------------------

Description

Get the sense of the focal mechanism rake, from the U, V, P, T vectors

Usage

```
GetRakeSense(uaz, upl, vaz, vpl, paz, ppl, taz, tpl)
```

Arguments

uaz	Azimuth of U vector
upl	dip of U vector
vaz	Azimuth of V vector
vpl	dip of V vector
paz	Azimuth of P vector
ppl	dip of P vector
taz	Azimuth of T vector
tpl	dip of T vector

Value

1, 0 to make sure the region of the T-axis is shaded and the P-axis is blank.

Note

The convention is for the T-axis to be shaded, so this subroutine determines the order of the polygons to be plotted so that the appropriate regins are filled.

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

GetRake

Examples

```
mc =CONVERTSDR(65,25,13)
```

```
angsense = GetRakeSense(mc$U$az, mc$U$dip, mc$V$az, mc$V$dip,mc$P$az, mc$P$dip,mc$T$az, mc$T$dip)
```

getUWfocs

Get UW focals

Description

Get UW focal mechanisms from a file. These are often called A and M cards

Usage

```
getUWfocs(amfile)
```

Arguments

amfile	character, file name
--------	----------------------

Details

UW focal mechanisms are stored as A and M cards. The A card describes the hypocenter the M card describes the focal mechanism.

Value

List:

lon	numeric, longitude
lat	numeric, latitude
str1	numeric, strike of plane 1
dip1	numeric, dip of plane 1
rake1	numeric, rake of plane 1
str2	numeric, strike of plane 2
dip2	numeric, dip of plane 2
rake2	numeric, rake of plane 2
sc	character, some GMT info for scale
iexp	character, some GMT info for scale
name	character, name
yr	numeric, year
mo	numeric, month
dom	numeric, day of month
jd	numeric, julian day
hr	numeric, hour
mi	numeric, minute
se	numeric, second
z	numeric, depth
mag	numeric, magnitude

Note

Uses UW2 format, so full 4 digit year is required

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

http://www.unc.edu/~leesj/XM_DOC/xm_hypo.doc.html

See Also

getCMT

Examples

```
## Not run:
##### uwpickfile is an ascii format file from University of Washington
G1 = getUWfocs(uwpickfile)

plot(G1$lon, G1$lat)

MEKS = list(lon=G1$lon, lat=G1$lat, str1=G1$str1,
dip1=G1$dip1, rake1=G1$rake1, dep=G1$z, name=G1$name)

## utm projection
PROJ = GEOMap::setPROJ(type=2, LAT0=mean(G1$lat) , LON0=mean(G1$lon) )

XY = GEOMap::GLOB.XY(G1$lat, G1$lon, PROJ)

plot(range(XY$x), range(XY$y), type='n', asp=1)

plotmanyfoc(MEKS, PROJ, focsiz=0.05)

## End(Not run)
```

Description

Hammer Equal Area projection

Usage

```
HAMMERprojXY(phi, lam)
```

Arguments

phi	Latitude, radians
lam	Longitude, radians

Value

list:

x	coordinate for plotting
y	coordinate for plotting

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

Examples

```
HAMMERprojXY(-25*pi/180, -16*pi/180)
```

hudson.net

Hudson Net Plot

Description

Plot a Hudson plot as preparation for plotting T-k values for focal mechanisms.

Usage

```
hudson.net(add = FALSE, POINTS = TRUE, TEXT = TRUE,
           colint = "grey", colext = "black")
```

Arguments

add	logical, TRUE=add to existing plot
POINTS	logical, TRUE=add points
TEXT	logical, TRUE=add points
colint	color for interior lines
colext	color for exterior lines

Details

Draws a T-k plot for moment tensors

Value

Graphical Side effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

Hudson, J.A., Pearce, R.G. and Rogers, R.M., 1989. Source time plot for inversion of the moment tensor, J. Geophys. Res., 94(B1), 765-774.

See Also

hudson.plot

Examples

```
hudson.net()

Mtens <- c(-0.412, 0.084, 0.328 ,0.398, -1.239, 1.058)

M1 <- matrix(c(Mtens[1], Mtens[4], Mtens[5], Mtens[4],
Mtens[2], Mtens[6], Mtens[5],Mtens[6], Mtens[3]), ncol=3, nrow=3,
byrow=TRUE)

E1 <- eigen(M1)

hudson.plot(E1$values)
```

hudson.plot

Hudson Source Type Plot

Description

Hudson Source Type Plot

Usage

```
hudson.plot(m, col = "red", pch = 21, lwd = 2, cex = 1, bg="white")
```

Arguments

m	vector of eigen values, sorted
col	color
pch	plotting char
lwd	line width
cex	character expansion
bg	background color for filled symbols

Details

Add to existing Hudson net

Value

Side effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

Hudson, J.A., Pearce, R.G. and Rogers, R.M., 1989. Source time plot for inversion of the moment tensor, J. Geophys. Res., 94(B1), 765-774.

See Also

hudson.net

Examples

```
hudson.net()

Mtens <- c(-0.412, 0.084, 0.328 ,0.398, -1.239, 1.058)

M1 <- matrix(c(Mtens[1], Mtens[4], Mtens[5], Mtens[4],
Mtens[2], Mtens[6], Mtens[5],Mtens[6],
Mtens[3]), ncol=3, nrow=3, byrow=TRUE)

E1 <- eigen(M1)

hudson.plot(E1$values)
```

imageP	<i>P-wave radiation pattern</i>
--------	---------------------------------

Description

Amplitude of P-wave radiation pattern from Double-Couple earthquake

Usage

```
imageP(phiS, del, lam, SCALE = FALSE, UP = FALSE, col = NULL)
```

Arguments

phiS	strike
del	dip
lam	lambda
SCALE	logical, TRUE=add scale on side of plot
UP	upper/lower hemisphere
col	color

Details

This program calls radP to calculate the radiation pattern and it plots the result using the standard image function

Value

Used for the graphical side effect

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

References

K.~Aki and P.~G. Richards. *Quantitative seismology*. University Science Books, Sausalito, Calif., 2nd edition, 2002.

See Also

radP, SDRfoc

Examples

```
MEC =SDRfoc(65,25,13, u=FALSE, ALIM=c(-1,-1, +1, +1), PLOT=FALSE)
imageP(MEC$az1, MEC$dip1, MEC$rake1, SCALE=TRUE, UP=MEC$UP, col=rainbow(100) )
```

imageSCALE

add scale on side of image

Description

add scale to side of an image plot

Usage

```
imageSCALE(z, col, x, y = NULL, size = NULL, digits = 2,  
labels = c("breaks", "ranges"), nlab = 10)
```

Arguments

z	elevation matrix
col	palette for plotting
x	x location on plot
y	y location on plot
size	length of scale
digits	digits on labels
labels	breaks to be plotted
nlab	number of breaks to be plotted

Value

Used for graphical side effect

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

Examples

```
data(volcano)  
image(volcano, col=rainbow(100) )  
  
imageSCALE(volcano, rainbow(100), 1.015983, y = 0.874668,  
size = .01, digits =  
2, labels = "breaks", nlab = 20)
```

imageSH	<i>P-wave radiation pattern</i>
---------	---------------------------------

Description

Amplitude of SH-wave radiation pattern from Double-Couple earthquake

Usage

```
imageSH(phiS, del, lam, SCALE = FALSE, UP = FALSE, col = NULL)
```

Arguments

phiS	strike
del	dip
lam	lambda
SCALE	logical, TRUE=add scale on side of plot
UP	upper/lower hemisphere
col	color

Details

This program calls radP to calculate the radiation pattern and it plots the result using the standard image function

Value

Used for the graphical side effect

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

References

K.~Aki and P.~G. Richards. *Quantitative seismology*. University Science Books, Sausalito, Calif., 2nd edition, 2002.

See Also

radSH, SDRfoc

Examples

```
MEC =SDRfoc(65,25,13, u=FALSE, ALIM=c(-1,-1, +1, +1), PLOT=FALSE)
imageSH(MEC$az1, MEC$dip1, MEC$rake1, SCALE=TRUE, UP=MEC$UP, col=rainbow(100) )
```

imageSV	<i>P-wave radiation pattern</i>
---------	---------------------------------

Description

Amplitude of SV-wave radiation pattern from Double-Couple earthquake

Usage

```
imageSV(phiS, del, lam, SCALE = FALSE, UP = FALSE, col = NULL)
```

Arguments

phiS	strike
del	dip
lam	lambda
SCALE	logical, TRUE=add scale on side of plot
UP	upper/lower hemisphere
col	color

Details

This program calls radP to calculate the radiation pattern and it plots the result using the standard image function

Value

Used for the graphical side effect

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

References

K.~Aki and P.~G. Richards. *Quantitative seismology*. University Science Books, Sausalito, Calif., 2nd edition, 2002.

See Also

radSV, SDRfoc

Examples

```
MEC =SDRfoc(65,25,13, u=FALSE, ALIM=c(-1,-1, +1, +1), PLOT=FALSE)
imageSV(MEC$az1, MEC$dip1, MEC$rake1, SCALE=TRUE, UP=MEC$UP, col=rainbow(100) )
```

inverseTAPE	<i>Inverse Moment Tensor</i>
-------------	------------------------------

Description

Inverse moment tensor from Tape angles.

Usage

inverseTAPE(GAMMA, BETA)

Arguments

GAMMA	Longitude, degrees
BETA	CoLatitude, degrees

Details

Uses Tape and Tape lune angles to estimate the moment tensor. This function is the inverse of the SourceType calculation. There are two solutions to the systems of equations.

Vectors are scaled by the maximum value.

Value

Moment tensor list:

Va	vector, First solution
Vb	vector, First solution

Note

The latitude is the CoLatitude.

Either vector can be used as a solution.

Orientation of moment tensor is not preserved in the lune plots.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

Tape, W., and C. Tape (2012), A geometric comparison of source-type plots for moment tensors, Geophys. J. Int., 190, 499-510.

See Also

SourceType

Examples

```

lats = seq(from = -80, to = 80, by=10)
lons = seq(from=-30, to=30, by=10)

i = 3
j = 3
u = inverseTAPE( lons[i], 90-lats[j] )

```

jimbo

Moment Tensors from the Harvard CMT

Description

Moment Tensors from the Harvard CMT

Usage

```
data(jimbo)
```

Format

A list of 9 moment tensors from the Kamchatka region.

Source

<http://www.globalcmt.org/CMTsearch.html>

References

Ekstrom, G.; Nettles, M. & DziewoDski, A. The Global CMT Project 2004-2010: centroid-moment tensors for 13,017 earthquakes *Physics of the Earth and Planetary Interiors*, 2012.

JMAT

Vertical Rotation matrix

Description

Vertical Rotation matrix

Usage

```
JMAT(phi)
```

Arguments

phi angle, degrees

Details

First rotate to plan, then within plane rotate to view angle.

Value

3 by 3 matrix

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

ROTX, ROTZ, ROTY

Examples

```
phi = 18  
MAT = JMAT(phi)  
v1 = c(1,1,0)  
v2 = MAT
```

justfocXY

Plot focal mechanism

Description

Add simple focal mechanisms to plot

Usage

```
justfocXY(MEC, x = x, y = y, focsiz=1 , fcol = gray(0.9),  
          fcolback = "white", xpd = TRUE)
```

Arguments

MEC	MEC structure
x	x-coordinate of center
y	y-coordinate of center
focsiz	size of focal sphere in inches
fcol	color of shaded region
fcolback	color of background region
xpd	logical, whether to extend the plot beyond, or to clip

Details

This routine can be used to add focal mechanisms on geographic map or other plot.

Value

Used for graphical side effect

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

SDRfoc, foc.color

Examples

```
#### read in some data:

Z1 = c(159.33,51.6,206,18,78,
161.89,54.5,257,27,133,
170.03,53.57,-44,13,171,
154.99,50.16,-83,19,-40,
151.09,47.15,123,23,-170,
176.31,51.41,-81,22,122,
153.71,46.63,205,28,59,
178.39,51.21,-77,16,126,
178.27,51.1,-86,15,115,
177.95,51.14,-83,25,126,
178.25,51.18,215,16,27
)

MZ = matrix(Z1, ncol=5, byrow=TRUE)

plot(MZ[,1], MZ[,2], type='n', xlab="LON", ylab="LAT", asp=1)

for(i in 1:length(MZ[,1]))
{
```

```
paste(MZ[i,3], MZ[i,4], MZ[i,5])

MEC = SDRfoc(MZ[i,3], MZ[i,4], MZ[i,5], u=FALSE, ALIM=c(-1,-1, +1, +1), PLOT=FALSE)
fcol = foc.color(foc.icolor(MEC$rake1), pal=1)
justfocXY(MEC, x=MZ[i,1], y =MZ[i,2] , focsiz=.5, fcol =fcol , fcolback = "white", xpd = TRUE)

}
```

KAMCORN

SDR data from the Harvard CMT catalog

Description

Strike-Dip-Rake and Locations of Harvard CMT catalog for the intersection of the Kamchataka and Aleutian arcs

Usage

```
data(KAMCORN)
```

Format

The format is: chr "KAMCORN"

Details

The data is selected from the CMT catalog. Parameters are extracted from the normal distribution. Format of the list of data save in KAMCORN is: list(LAT=0 , LON =0 , DEPTH=0 , STRIKE=0 , DIP=0 , RAKE=0)

Source

<http://www.globalcmt.org/CMTsearch.html>

References

G. Ekstrom. Rapid earthquake analysis utilizes the internet. Computers in Physics, 8:632-638, 1994.

Examples

```

data(KAMCORN)
plot(KAMCORN$LON, KAMCORN$LAT, xlab="LON", ylab="LAT" ,
      main="Kamchatka-Aleutian Inersection", asp=1)

#####
Paz =vector()
Pdip =vector()
Taz =vector()
Tdip =vector()
h = vector()
v = vector()

IFcol = vector()
Fcol = vector()

for(i in 1:10)
{
  Msdr = CONVERTSDR(KAMCORN$STRIKE[i],
                    KAMCORN$DIP[i], KAMCORN$RAKE[i] )
  MEC = MRake(Msdr$M)
  MEC$UP = FALSE
  IFcol[i] = foc.icolor(MEC$rake1)
  Fcol[i] = foc.color(IFcol[i], 1)

  az1 = Msdr$M$az1
  dip1 = Msdr$M$d1
  az2 = Msdr$M$az2
  dip2 = Msdr$M$d2
  BBB = Bfocvec(az1, dip1, az2, dip2)
  V = ternfoc.point(BBB$Bdip, Msdr$M$pd, Msdr$M$td )
  Paz[i] = Msdr$M$paz
  Pdip[i] = Msdr$M$pd
  Taz[i] = Msdr$M$taz
  Tdip[i] = Msdr$M$td
  h[i] = V$h
  v[i] = V$v

  justfocXY( MEC, fcol = Fcol[i], KAMCORN$LON[i],
             KAMCORN$LAT[i] , focsiz = 0.4 )
}

```

lowplane

Plot one Fault plane on stereonet

Description

takes azimuth and dip and projects the greaat circle on the focla sphere

Usage

```
lowplane(az, dip, col = par("col"), UP = FALSE, PLOT = TRUE)
```

Arguments

az	degrees, azimuth of strike of plane
dip	degrees, dip
col	color of plane
UP	upper/lower hemisphere
PLOT	add to plot

Details

Here azimuth is measured from North, and represents the actual strike of the fault line.

Value

list of x,y coordinates of plane

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

net

Examples

```
net()
lowplane(65,23)
```

m2tk

Moment tensor to T-k

Description

Moment tensor to T-k

Usage

```
m2tk(m0)
```

Arguments

m0 moment tensor eigenvalues, sorted decending

Details

Convert 3 eigen values of a moment tensor to T-k coordinates

Value

list(t, k)

Author(s)

Keehoon Kim<keehoon@live.unc.edu> Jonathan M. Lees<jonathan.lees@unc.edu>

References

Hudson

See Also

tk2uv, hudson.net, hudson.plot

Examples

```
v = c(2,-1,-1)
m2tk(v)
```

makeblock3D

Make a 3D block Structure

Description

Given vertices of a 3D block, create a glyph structure (faces and normals)

Usage

```
makeblock3D(block1)
```

Arguments

block1 matrix of vertices

Value

glyph structure list

aglyph list of faces (x,y,z)

anorm Normals to faces

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

ROTZ, ROTY, ROTX, BOXarrows3D, Z3Darrow, TRANmat

Examples

```
block1 = matrix(c(0,0,0,
  1,0,0,
  1,0.5,0,
  0,0.5,0,
  0,0,-2,
  1,0,-2,
  1,0.5,-2,
  0,0.5,-2), byrow=TRUE, ncol=3)

Bblock1 = makeblock3D(block1)
```

makenet	<i>Equal-Angle Stereonet</i>
---------	------------------------------

Description

Creates but does not plot an Equal-Angle (Schmidt) Stereonet

Usage

```
makenet()
```

Value

- list of x,y, values for drawing lines
- | | |
|----|-----------------------------|
| x1 | x-coordinate start of lines |
| y1 | y-coordinate start of lines |
| x2 | x-coordinate end of lines |
| y2 | y-coordinate end of lines |

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

References

Snyder, John P., 1987, Map Projections-a working manual, USGS-Professional Paper, 383p. pages 185-186

See Also

net, pnet

Examples

```
MN = makenet()

pnet(MN)
```

MapNonDouble

Map moment tensors

Description

Plot moment tensors on map

Usage

```
MapNonDouble(Locs, moments, sel = 1, siz = 0.2,
col=rgb(1, .75, .75), PLANES = TRUE, add = FALSE, LEG=FALSE)
```

Arguments

Locs	Locations, x,y
moments	list of moments: seven elements. See details.
sel	integer, index of which to plot
siz	size to plot, inches
col	color, either a single color, rgb, or a color palette.
PLANES	logical, whether to add nodal planes, default=TRUE
add	logical, whether to add to plot, default=FALSE
LEG	logical, whether to add focal mech legend based on color coding, default=FALSE

Details

Moment tensors are added to an existing plot. The first element of the list is the integer index of the event. The next six elements are the moments in the following order, c(Mxx, Myy, Mzz, Mzy, Mxz, Mxy) .

If the data is in spherical coordinates, one must switch the sign of the Mrp and Mtp components, so:

```
Mrr = Mzz
Mtt = Mxx
Mpp = Myy
Mrt = Mxz
Mrp = -Myz
Mtp = -Mxy
```

A color palette can be provided for some details of the radiation patterns, e.g. col=rainbow(12). If col is NULL, the colors will be chosen according to focal.color from RFOC, based on rake of first nodal plane.

If col is NULL, then the colors are set by foc.color and it is appropriate to add a legend.

Value

list:

FOC	matrix, focal mechanism angles (strike, dip rake)
LAB	matrix, x-y location for labels

Note

If events are read in using spherical rather than cartesian coordinates need a conversion:

```

Mrr = Mzz
Mtt = Mxx
Mpp = Myy
Mrt = Mxz
Mrp = -Myz
Mtp = -Mxy

```

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

Ekstrom, G.; Nettles, M. & DziewoDski, A. The Global CMT Project 2004-2010: centroid-moment tensors for 13,017 earthquakes Physics of the Earth and Planetary Interiors, 2012.

See Also

doNonDouble, ShadowCLVD, angles, nodalLines, PTaxes, focal.color, foc.icolor

Examples

```

## Not run:

library(maps)
library(GEOmap)

##### load the data
data(widdenMoments)

##### to read in the data from a file,
## GG = scan("widdenMoments.txt",sep=" ",
## what=list(ID=0,Event="",Lat=0,Long=0,Depth=0,Mw=0,ML=0,DC=0,
## CLVD=0,ISO=0,VR=0,nsta=0,Mxx=0,Mxy=0,Mxz=0,
## Myy=0,Myz=0,Mzz=0,Mo=0,Ftest=0) )

GG = widdenMoments
Locs = list(y=GG$Lat,x=GG$Long)

```

```

ef = 1e20
moments = cbind(GG$ID, ef*GG$Mxx, ef*GG$Myx,
ef*GG$Mzz, ef*GG$Myz, ef*GG$Mxz,ef*GG$Mxy)

UTAH = map('state', region = c('utah'), plot=FALSE )

m1on = mean(UTAH$x, na.rm=TRUE)
m1at = mean(UTAH$y, na.rm=TRUE)

Gutah = maps2GEOmap(UTAH)

##### for mercator projection
PROJ = GEOmap::setPROJ(type = 1, LAT0 = m1at , LON0 = m1on)
Glocs = GEOmap::GLOB.XY(Locs$y, Locs$x, PROJ )
##### for UTM projection
PROJ = GEOmap::setPROJ(type = 2, LAT0 = m1at , LON0 = m1on)
Glocs = GEOmap::GLOB.XY(Locs$y, Locs$x, PROJ )

LIM1at = expandbound(Gutah$POINTS$lat)
LIM1on = expandbound(Gutah$POINTS$lon)

PLAT = pretty(LIM1at)
PLON = pretty(LIM1on)

##### plot the map

##### Utah is a little rectangular
dev.new(width=9, height=12)

plotGEOmapXY(Gutah,
LIM = c(min(PLON), min(PLAT) , max(PLON) , max(PLAT)) ,
PROJ=PROJ, axes=FALSE, xlab="", ylab="" )

### add tic marks
kbox = GEOmap::GLOB.XY(PLAT,PLON, PROJ)

sqrTICXY(kbox , PROJ, side=c(1,2,3,4), LLgrid=TRUE, col=grey(.7) )

##### add focal mechs
siz = 0.2

MapNonDouble(Glocs, moments,col=NULL, add=TRUE, LEG=TRUE)

up = par("usr")
ui = par("pin")
ratx = (up[2] - up[1])/ui[1]
raty = (up[4] - up[3])/ui[2]

```

```

        usizx = siz * ratx

        AXY = NoOverlap(Glocs$x,Glocs$y, usizx )

        MapNonDouble(AXY, moments,col=NULL,  add=TRUE, LEG=TRUE)

##### MapNonDouble(NXY, moments,col=NULL,  add=TRUE, LEG=TRUE)


## End(Not run)

```

mc2cart

Convert azimuth, dip to Cartesian Coordinates

Description

takes the pole information from a stereonet and returns the cartesian coordinates

Usage

```
mc2cart(az, dip)
```

Arguments

az	degrees, orientation angle, from North
dip	degrees, dip of pole

Value

list of x,y,z values

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

Examples

```

v1 = mc2cart(65,32)
v2 = mc2cart(135,74)

```

`mijedr`*Moment Tensor to Strike-Dip-Rake*

Description

Convert a normalized moment tensor from the CMT catalog to Strike-Dip-Rake.

Usage

```
mijedr(mxx, myy, mzz, mxy, mxz, myz)
```

Arguments

<code>mxx</code>	moment tensor 1,1
<code>myy</code>	moment tensor 2,2
<code>mzz</code>	moment tensor 3,3
<code>mxy</code>	moment tensor 1,2
<code>mxz</code>	moment tensor 1,3
<code>myz</code>	moment tensor 2,3

Details

the coordinate system is modified to represent a system centered on the source.

Value

Focal Mechanism list

Note

This code will convert the output of the website, <http://www.globalcmt.org/CMTsearch.html> when dumped in the psmea (GMT v>3.3) format.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

<http://www.globalcmt.org/CMTsearch.html>

See Also

`getCMT`

Examples

```
mijcdr(-1.96, 1.07, 0.89, 0.51, 0.08, -0.68)
```

MomentDist

Distance Between Moment Tensors

Description

Calculate the distance between moment tensors based on quaternions.

Usage

```
MomentDist(E1, E2)
```

Arguments

E1	Moment tensor
E2	Moment tensor

Details

Moment tensors should be right handed.

Value

angle in degrees

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

Tape and Tape, 2012

See Also

forcerighthand, testrightHAND

Examples

```
Mtens = c(-0.412, 0.084, 0.328, 0.398, -1.239, 1.058)
M1 = matrix(c(Mtens[1], Mtens[4], Mtens[5], Mtens[4], Mtens[2],
Mtens[6], Mtens[5], Mtens[6], Mtens[3]), ncol=3, nrow=3, byrow=TRUE)
```

```
Mtens = c(5.054, -2.235, -2.819, -0.476, 5.420, 5.594)
M2 = matrix(c(Mtens[1], Mtens[4], Mtens[5], Mtens[4], Mtens[2],
Mtens[6], Mtens[5], Mtens[6], Mtens[3]), ncol=3, nrow=3, byrow=TRUE)
```

```
E1 = eigen(M1)
```

```
### make sure these are a right handed system,
### ie x1 cross x2 = x3
```

```
E2 = eigen(M2)
```

```
### make sure these are a right handed system,
### ie x1 cross x2 = x3
testrightHAND(E1$vectors)
testrightHAND(E2$vectors)
```

```
E1$vectors = forcerighthand(E1$vectors)
```

```
E2$vectors = forcerighthand(E2$vectors)
```

```
testrightHAND(E1$vectors)
testrightHAND(E2$vectors)
```

```
MomentDist(E1, E2)
```

MRake

Rake Calculation

Description

Calculate various parameters associated with the Rake or Slip of an earthquake

Usage

```
MRake(M)
```

Arguments

```
M                list(uaz, ud, vaz, vd, paz, pd, taz, td)
```

Details

This routine takes the four poles U, V, P, T, and returns a MEC structure. (uaz, ud) = U pole azimuth and dip (vaz, vd)= V pole azimuth and dip (paz, pd)= P pole azimuth and dip (taz, td)= T pole azimuth and dip

Value

returns a MEC structure

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

CONVERTSDR, GetRakeSense, GetRake

Examples

```
mc = CONVERTSDR(329, 8, 110 )
MEC = MRake(mc$M)
```

net

EqualArea Stereonet

Description

Plot Equal Area Stereo-Net. Lambert azimuthal Equal-Area (Schmidt) from Snyder p. 185-186

Usage

```
net(add = FALSE, col = gray(0.7), border = "black", lwd = 1, LIM = c(-1, -1, +1, +1))
```

Arguments

add	logical, TRUE=add to existing plot
col	color of lines
border	color of outer rim of stereonet
lwd	linewidth of lines
LIM	bounding area for a new plot

Value

Used for graphical side effects

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

References

Snyder, John P., 1987, Map Projections-a working manual, USGS-Professional Paper, 383p. pages 185-186

See Also

pcirc

Examples

```
net(FALSE, col=rgb(.8,.7,.7) ,border='blue' )
```

nipXY	<i>Fault-Slip vector plot</i>
-------	-------------------------------

Description

Plots a fault plane and the slip vector. Used for geographic representation of numerous focal spheres.

Usage

```
nipXY(MEC, x = x, y = y, focsiz=1, fcol = gray(0.9), nipcol = "black", cex = 0.4)
```

Arguments

MEC	MEC structure
x	coordinate on plot
y	coordinate on plot
focsiz	size in inches
fcol	color for plotting
nipcol	color of slip point
cex	character expansion for slip point

Details

Slip vector is the cross product of the poles to the fault plane and auxilliary planes.

Value

LIST

Q output of qpoint

N slip vector

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

qpoint, CROSSL, lowplane, TOCART

Examples

```

set.seed(2015)
N = 20
lon=runif(20, 268.1563 , 305)
lat=runif(20, 7.593004, 25.926045)
str1=runif(20,50,100)
dip1=runif(20,10, 80)
rake1=runif(20,5, 180)

dep=runif(20,1,15)
name=seq(from=1, to=length(lon), by=1)
Elat=NULL
Elon=NULL
yr = rep(2017, times=N)
jd = runif(N, min=1, max=365)

MEKS = list(lon=lon, lat=lat, str1=str1, dip1=dip1,
rake1=rake1, dep=dep, name=name, yr=yr, jd = jd)

PROJ = GEOMap::setPROJ(type=2, LAT0=mean(lat) , LON0=mean(lon) ) ## utm

XY = GEOMap::GLOB.XY(lat, lon, PROJ)

plot(range(XY$x), range(XY$y), type='n', asp=1, xlab='km', ylab='km' )
for(i in 1:length(XY$x))
{
  Msdr = CONVERTSDR(MEKS$str1[i], MEKS$dip1[i],MEKS$rake1[i])
  MEC = MRake(Msdr$M)
  MEC$UP = FALSE

  jcol = foc.color(foc.icolor(MEC$rake1), pal=1)

  nipXY(MEC, x = XY$x[i], y = XY$y[i], focsiz=0.5, fcol = jcol, nipcol = 'black' , cex = 1)
}

```

nodalLines

Nodal Lines

Description

Add nodal planes to focal mechanism

Usage

```
nodalLines(strike, dip, rake, PLOT=TRUE)
```

Arguments

strike	numeric, strike of fault
dip	numeric, dip of fault
rake	numeric, rake of fault
PLOT	logical, add lines to plot, default=TRUE

Details

Lower Hemisphere focal plane.

Value

Side effects

Note

Lower Hemisphere based on FOCangles.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

doNonDouble, MapNonDouble, FOCangles

Examples

```
mo <- list(n=1, m1=1.035675e+017,
           m2=-1.985852e+016, m3=-6.198052e+014,
           m4=1.177936e+017, m5=-7.600627e+016, m6=-3.461405e+017)
moments <- cbind(mo$n, mo$m1, mo$m2, mo$m3, mo$m4, mo$m5, mo$m6)
doNonDouble(moments)
```

`normal.fault`*Normal Fault Cartoon*

Description

Illustrate a normal fault using animation

Usage

```
normal.fault(ANG = (45), anim = seq(from = 0, to = 1, by = 0.1),  
             KAPPA = 4, Light = c(45, 45))
```

Arguments

ANG	Angle of dip
anim	animation vector
KAPPA	Phong parameter for lighting
Light	lighting point

Details

Program will animate a normal fault for educational purposes. Animation must be stopped by halting execution.

Value

Graphical Side effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

`strikeslip.fault`, `thrust.fault`

Examples

```
normal.fault(45, anim=0, KAPPA=4, Light=c(-20, 80))  
  
## Not run:  
#### execute a stop command to stop this animation  
anim= seq(from=0, to=1, by=.1)  
  
normal.fault(45, anim=anim, KAPPA=4, Light=c(-20, 80))  
  
## End(Not run)
```

pcirc	<i>Circle Plot</i>
-------	--------------------

Description

Add a circle to a plot, with cross-hairs

Usage

```
pcirc(gcol = "black", border = "black", ndiv = 36)
```

Arguments

gcol	color of crosshairs
border	border color
ndiv	number of divisions for the circle

Value

no return values, used for side effects

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

net

Examples

```
net()
pcirc(gcol = "green", border = "purple", ndiv = 36)
```

pglyph3D	<i>Plot a 3D body on an existing graphic</i>
----------	--

Description

rotates a body in 3D and plots projection on existing plot

Usage

```
pglyph3D(aglyph, M = diag(1, nrow = 4), M2 = diag(1, nrow = 4),
          anorms = list(), zee = c(0, 0, 1), col = "white", border = "black")
```

Arguments

aglyph	glyph structure describing the vertices and normal vectors of a 3D body
M	rotation matrix 1
M2	rotation matrix 2
anorms	up vector
zee	up vector
col	coor of body
border	color of border

Details

Hidden sides are removed and phong shading is introduced to create 3D effect.

The input consists of an object defined by a list structure, list(aglyph, anorm) where aglyph is list of 3D polygons (faces) and anorm are outward normals to these faces.

Value

Used for side effect on plots

Note

For unusual rotations or bizarre bodies, this routine may produce strange looking shapes.

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

References

Rogers and Adams, 1990, Mathematical Elements for Computer Graphics, McGraw-Hill, 611p.

See Also

Z3Darrow, ROTX, ROTY, ROTZ

Examples

```
### create the 3D object
len = .7
basethick=.05
headlip=.02
headlen=.3

#### create a 3D glyph structure
aglyph = Z3Darrow(len = len , basethick =basethick , headlen =headlen ,
headlip=headlip )

#### define the up vector
```

```

myzee = matrix(c(0,0,1, 1), nrow=1, ncol=4)

##### set rotation angles:
gamma =12
beta =39
alpha = 62

##### set up rotation matrix
R3 = ROTZ(gamma)

R2 = ROTY(beta)

R1 = ROTZ(alpha)

### create rotation matrix
M =      R1

M2 =      R1

plot(c(-1,1), c(-1,1))

pglyph3D(aglyph$aglyph, anorms=aglyph$anorm , M=M, M2=M2, zee=myzee ,
col=rgb(.7, 0,0) )

```

phong3D

Phong shading for a 3D body

Description

Create phong shading for faces showing on the 3D block

Usage

```

phong3D(aglyph, M = diag(1, nrow = 4), M2 = diag(1, nrow = 4),
        Light = c(45, 45), anorms = list(), zee = c(0, 0, 1),
        col = "white", border = "black")

```

Arguments

aglyph	3-D body list of faces and normals
M	Rotation Matrix
M2	Viewing Matrix
Light	light source direction
anorms	normals to faces

zee	Up vector for Body
col	color for faces
border	border color for sides

Details

Uses a standard phong shading model based on the dot product of the face normal vector and direction of incoming light.

Value

Graphical Side effect

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

Watt, Alan. Fundamentals of Three-dimensional Computer Graphics, Addison-Wesley, 1989, 430p.

See Also

makeblock3D, BOXarrows3D, PROJ3D, Z3Darrow, pglyph3D

Examples

```
##### create a block and rotation matrix, then color it
ANG=(45)
DEGRAD = pi/180

y1 = 1.5

y2 = y1 - 1/tan((ANG)*DEGRAD)

z1 = 1
x1 = 1

Ablock1 = matrix(c(0,0,0,
  1,0,0,
  1,y1,0,
  0,y1,0,
  0,0,-1,
  1,0,-1,
  1,y2,-1,
  0,y2,-1), byrow=TRUE, ncol=3)
```

```

Nblock1 = makeblock3D(Ablock1)
Light=c(45,45)
angz = -45
angx = -45

R1 = ROTZ(angz)
R2 = ROTX(angx)

M = R1

Z2 = PROJ3D(Nblock1$aglyph, M=M, anorms=Nblock1$anorm , zee=c(0,0,1))
RangesX = range(attr(Z2, "RangesX"))

RangesY = range(attr(Z2, "RangesY"))

plot( RangesX, RangesY, type='n', asp=1, ann=FALSE, axes=FALSE)

phong3D(Nblock1$aglyph, M=M, anorms=Nblock1$anorm , Light = Light,
zee=c(0,0,1), col=rgb(.7,.5, .5) , border="black")

```

PKAM

P and T-axes data from the Harvard CMT catalog

Description

P and T-axes and Locations of Harvard CMT catalog for the intersection of the Kamchataka and Aleutian arcs

Usage

data(PKAM)

Format

The format is: chr "PKAM"

Details

The data is selected from the CMT catalog. Parameters are extracted from the standard web distribution. Format of the list of data save in PKAM is:

itemPazP-axis azimuth angle itemPdipP-axis dip angle itemTazT-axis azimuth angle itemTdipT-axis dip angle itemhhorizontal point to plot on ternary plot itemvvertical point to plot on ternary plot itemfcolscolors, not used itemLATSLatitude itemLONSLongitude itemIFcolinteger pointer to internal color itemyear, not used itemJDHMJulian Day, hour, minute, not used itemJDHMSJulian Day, hour, minute, seconds

Source

<http://www.globalcmt.org/CMTsearch.html>

References

G. Ekstrom. Rapid earthquake analysis utilizes the internet. Computers in Physics, 8:632-638, 1994.

Examples

```
data(PKAM)
##

##### plot the locations:
plot( RPMG::fmod(PKAM$LONS, 360), PKAM$LATS)
#####

PlotTernfoc(PKAM$h,PKAM$v,x=0, y=0, siz=1, fcols='black', add=FALSE,
LAB=TRUE)

##### change the colors for the plot

acols = rainbow(7)
fcols = acols[PKAM$IFcol]

#####

PlotTernfoc(PKAM$h,PKAM$v,x=0, y=0, siz=1, fcols=fcols, add=FALSE,
LAB=TRUE)
```

plotfoc

Plot Focal Radiation Patterns

Description

Takes a MEC structure and plots all three radiation patterns.

Usage

```
plotfoc(MEC)
```

Arguments

MEC MEC list

Details

Plot makes three figures after calling `par(mfrow=c(3,1))`.

Value

Graphical Side Effects.

Note

Basic MEC List Structure

az1	azimuth angle plane 1, degrees
dip1	dip angle plane 1, degrees
az2	azimuth angle plane 2, degrees
dip2	dip angle plane 2, degrees
dir	0,1 to determine which section of focal sphere is shaded
rake1	rake angle plane 1, degrees
dipaz1	dip azimuth angle plane 1, degrees
rake2	rake angle plane 2, degrees
dipaz2	dip azimuth angle plane 2, degrees
P	pole list(az, dip) P-axis
T	pole list(az, dip) T-axis
U	pole list(az, dip) U-axis
V	pole list(az, dip) V-axis
F	pole list(az, dip) F-axis
G	pole list(az, dip) G-axis
sense	0,1 to determine which section of focal sphere is shaded
M	list of focal parameters used in some calculations
UP	logical, TRUE=upper hemisphere
icol	index to suite of colors for focal mechanism
ileg	Kind of fault
fcol	color of focal mechanism
CNVRG	Character, note on convergence of solution
LIM	vector plotting region (x1, y1, x2, y2)

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

SDRfoc, Mrake, Pradfoc, radiateSH, radP, radSV, SVradfoc, radiateP, radiateSV, radSH, SHradfoc, imageP, imageSH, imageSV

Examples

```
M = SDRfoc(-25, 34, 16,u = FALSE, ALIM = c(-1, -1, +1, +1), PLOT=FALSE)
plotfoc(M)
```

`plotmanyfoc`*Plot Many Focals*

Description

Plot a long list of focal mechanisms

Usage

```
plotmanyfoc(MEK, PROJ, focsiz = 0.5, foccol = NULL,  
UP=TRUE, focstyle=1, PMAT = NULL, LEG = FALSE, DOBAR = FALSE)
```

Arguments

MEK	List of Focal Mechanisms, see details
PROJ	Projection
focsiz	focal size, inches
foccol	focal color
UP	logical, UP=TRUE means plot upper hemisphere (DEFAULT=TRUE)
focstyle	integer, 1=beach ball, 2=nipplot, 3=strike-slip, 4=P-T, 5=P, 6=T
PMAT	Projection Matrix from persp
LEG	logical, TRUE= add focal legend for color codes
DOBAR	add strike dip bar at epicenter

Details

Input MEK list contains

```
MEKS = list(lon=0, lat=0, str1=0, dip1=0, rake1=0, dep=0, name="", Elat=0, Elon=0)
```

Value

Graphical Side Effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

Lees, J. M., Geotouch: Software for Three and Four Dimensional GIS in the Earth Sciences, Computers & Geosciences, 26, 7, 751-761, 2000.

See Also

justfocXY

Examples

```

set.seed(2015)
N = 20
lon=runif(20, 268.1563 , 305)
lat=runif(20, 7.593004, 25.926045)
str1=runif(20,50,100)
dip1=runif(20,10, 80)
rake1=runif(20,5, 180)

dep=runif(20,1,15)
name=seq(from=1, to=length(lon), by=1)
Elat=NULL
Elon=NULL
yr = rep(2017, times=N)
jd = runif(N, min=1, max=365)

MEKS = list(lon=lon, lat=lat, str1=str1, dip1=dip1,
rake1=rake1, dep=dep, name=name, yr=yr, jd = jd)

PROJ = GE0map::setPROJ(type=2, LAT0=mean(lat) , LON0=mean(lon) ) ## utm

XY = GE0map::GLOB.XY(lat, lon, PROJ)

plot(range(XY$x), range(XY$y), type='n', asp=1)

plotmanyfoc(MEKS, PROJ, focsiz=0.5)

```

plotMEC

Plot a Focal Mechanism

Description

Plot a Focal Mechanism

Usage

```
plotMEC(x, detail = 0, up = FALSE)
```

Arguments

x	Mechanism list
detail	level of detail
up	logical, Upper or lower hemisphere

Value

Side Effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

Examples

```
mc = CONVERTSDR(65, 32, -34 )
plotMEC(mc, detail=2, up=FALSE)
```

PlotPlanes

Plot Fault an Auxilliary Planes

Description

Plot both fault and auxilliary planes

Usage

```
PlotPlanes(MEC, col1 = 1, col2 = 3)
```

Arguments

MEC	MEC structure
col1	color for plane 1
col2	color for plane 2

Details

Given MEC structure and focal mechanism plot both planes. This code adds to existing plot, so net() should be called.

Value

Graphical Side Effects

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

net, lowplane

Examples

```
net()

MFOC1 = SDRfoc(65,25,13, u=FALSE, ALIM=c(-1,-1, +1, +1), PLOT=FALSE)
PlotPlanes(MFOC1, 'green', 'red' )
```

PlotPTsmooth	<i>Plot Smooth PT-axes</i>
--------------	----------------------------

Description

Project PT axes on the sphere and smooth the image. This function requires function kde2d, from the MASS library.

Usage

```
PlotPTsmooth(paz, pdip, x = 0, y = 0, siz = 1, bcol = "white", border ="black",
             IMAGE = TRUE, CONT = TRUE, cont.col = "black",
             pal = terrain.colors(100), LABS = FALSE, add = FALSE, NCP=50, NIP=200)
```

Arguments

paz	vector of Axis azimuths, degrees
pdip	vector of dip angles, degrees
x	x-location of plot center in user coordinates
y	y-location of plot center in user coordinates
siz	siz of plot in user coordinates
bcol	color
border	border color
IMAGE	logical, TRUE=create an image plot
CONT	logical, TRUE=add contour lines
cont.col	color of contour lines
pal	pallette for image plot
LABS	text Label for image
add	logical, TRUE=add to plot
NCP	integer, Number of points to use for calculating smoothed contours, default=50
NIP	integer, Number of points to use for calculating smoothed image, default=200

Details

Program requires MASS library for 2D smoothing routine kde2d.

For calculating contours the kde2d program creates a smoothed 2D image using NCP points per side. For the images, NIP points are used. To reduce the size of plots, or, if the subplots are very small, reduce NIP to a smaller value for faster plotting.

Value

Graphical Side Effect

Note

Points that fall on the opposite hemisphere are reflected through the origin.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

kde2d

Examples

```
plot(c(-1,1), c(-1,1), asp=1, type='n')

paz = rnorm(100, mean=297, sd=10)
pdip = rnorm(100, mean=52, sd=8)

PlotPTsmooth(paz, pdip, x=0.5, y=.5, siz=.3, border=NA, bcol='white' ,
LABS=FALSE, add=FALSE, IMAGE=TRUE, CONT=FALSE)

taz = rnorm(100, mean=138, sd=10)
tdip = rnorm(100, mean=12, sd=8)

PlotPTsmooth(taz, tdip, x=-.5, y=.4, siz=.3, border=NA, bcol='white' ,
LABS=FALSE, add=FALSE, IMAGE=TRUE, CONT=TRUE)

##### put them together
plot(c(-1,1), c(-1,1), asp=1, type='n')
PlotPTsmooth(paz, pdip, x=0, y=, siz=1, border=NA, bcol='white' ,
LABS=FALSE, add=FALSE, IMAGE=TRUE, CONT=FALSE)
PlotPTsmooth(taz, tdip, x=0, y=, siz=1, border=NA, bcol='white' ,
LABS=FALSE, add=TRUE, IMAGE=FALSE, CONT=TRUE)
```

PlotTernfoc

Ternary Distribution of focal mechanisms

Description

Create and plot a ternary diagram using rake angle to distribute focal mechanisms on a ternary diagram.

Usage

```
PlotTernfoc(h, v, x = 0, y = 0, siz = 1, fcols = "black", LABS = FALSE, add = FALSE)
```

Arguments

h	x-coordinate on ternary plot
v	y-coordinate of ternary plot
x	x Location of center of Ternary plot
y	y Location of center of Ternary plot
siz	size of plot in user coordinates
fcols	vector of colors associated with each focal mechanism
LABS	logical, TRUE=add labels at vertices of Ternary plot
add	logical, add to plot=TRUE

Value

Used for graphical side effect.

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

References

J. M. Lees. Geotouch: Software for three and four dimensional gis in the earth sciences. Computers & Geosciences, 26(7):751–761, 2000

See Also

ternfoc.point, Bfocvec

Examples

```

Z1 = c(159.33,51.6,206,18,78,
161.89,54.5,257,27,133,
170.03,53.57,-44,13,171,
154.99,50.16,-83,19,-40,
151.09,47.15,123,23,-170,
176.31,51.41,-81,22,122,
153.71,46.63,205,28,59,
178.39,51.21,-77,16,126,
178.27,51.1,-86,15,115,
177.95,51.14,-83,25,126,
178.25,51.18,215,16,27
)

MZ = matrix(Z1, ncol=5, byrow=TRUE)

h = vector()
v = vector()
Fcol = vector()
for(i in 1:length(MZ[,3]))
{
  Msdr = CONVERTSDR(MZ[i,3], MZ[i,4], MZ[i,5])
  MEC = MRake(Msdr$M)
  MEC$UP = FALSE

  az1 = Msdr$M$az1
  dip1 = Msdr$M$d1
  az2 = Msdr$M$az2
  dip2 = Msdr$M$d2
  BBB = Bfocvec(az1, dip1, az2, dip2)
  V = ternfoc.point(BBB$Bdip, Msdr$M$pd, Msdr$M$td )

  h[i] = V$h
  v[i] = V$v
  Fcol[i] = foc.color(foc.icolor(MEC$rake1), pal=1)
}

PlotTernfoc(h,v,x=0, y=0, siz=1, fcols=Fcol, add=FALSE, LAB=TRUE)

MFOC1 = SDRfoc(65,90,1, u=FALSE, ALIM=c(-1,-1, +1, +1), PLOT=FALSE)
  Fcol1 = foc.color(foc.icolor(MFOC1$rake1), pal=1)
MFOC2 = SDRfoc(135,45,-90, u=FALSE, ALIM=c(-1,-1, +1, +1), PLOT=FALSE)
  Fcol2 = foc.color(foc.icolor(MFOC2$rake1), pal=1)
MFOC3 = SDRfoc(135,45,90, u=FALSE, ALIM=c(-1,-1, +1, +1), PLOT=FALSE)
  Fcol3 = foc.color(foc.icolor(MFOC3$rake1), pal=1)

justfocXY( MFOC3, fcol = Fcol3, 1.2, -0.9, focsiz = 0.4 )
justfocXY( MFOC2, fcol = Fcol2, -1.2, -0.9, focsiz = 0.4 )
justfocXY( MFOC1, fcol = Fcol1, 0, 1.414443+.2, focsiz = 0.4 )

```

PLTcirc

Circle Plot with Cross Hairs

Description

Plot an arc of a circle with cross-hairs.

Usage

```
PLTcirc(gcol = "black", border = "black", ndiv = 36,
        ang = c(-pi, pi), PLOT = TRUE, add = FALSE)
```

Arguments

gcol	cross hairs color
border	border color
ndiv	number of divisions
ang	vector from ang[1] to ang[2] in radians
PLOT	logical, if TRUE plot
add	logical, if TRUE add to existing plot

Value

list used for plotting:

x	x coordinates
y	y coordinates
phi	angles, radians

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

Examples

```
PLTcirc(gcol = "purple", border = "black", ndiv = 36, ang = c(-pi, pi), PLOT = TRUE, add = FALSE)
```

```
PLTcirc(gcol = NULL, border = "green", ndiv = 36, ang = c(-pi/4, pi/4), PLOT = TRUE, add = TRUE)
```

pnet	<i>plot stereonet</i>
------	-----------------------

Description

Plots stereonet created by makenet

Usage

```
pnet(MN, add = FALSE, col = gray(0.7), border = "black", lwd = 1)
```

Arguments

MN	Net structure created by makenet
add	TRUE= add to existing plot
col	color of lines
border	color for outside border
lwd	line width

Value

Used Graphical Side Effects.

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

References

Snyder, John P., 1987, Map Projections-a working manual, USGS-Professional Paper, 383p. pages 185-186

See Also

net, pnet

Examples

```
MN = makenet()

pnet(MN)
```

polyfoc

Plot the focal mechanism polygon

Description

Calculate the projection of the focal mechanism polygon

Usage

```
polyfoc(strike1, dip1, strike2, dip2, PLOT = FALSE, UP = TRUE)
```

Arguments

strike1	strike of plane 1, degrees
dip1	dip of plane 1, degrees
strike2	strike of plane 1, degrees
dip2	dip of plane 2, degrees
PLOT	logical, TRUE = add to plot
UP	upper/lower hemisphere

Value

List of coordinates of polygon

Px	x-coordinates of polygon
Py	y-coordinates of polygon

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

faultplane

Examples

```
MEC = SDRfoc(13,59,125, PLOT=FALSE)

net()
ply = polyfoc(MEC$az1, MEC$dip1, MEC$az2, MEC$dip2, PLOT = TRUE, UP = TRUE)
```

Pradfoc

Plot P-wave radiation

Description

Plot P-wave radiation with information from the pickfile and waveform data

Usage

```
Pradfoc(A, MEC, GU, pscale, col)
```

Arguments

A	Pickfile structure
MEC	MEC structure
GU	Waveform Event Structure
pscale	logical (not used)
col	color palette

Details

Image plot of the P radiation pattern

Value

Graphical Side effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

imageP

Examples

```
MEC = SDRfoc(65, 32, -34, u=TRUE, ALIM=c(-1,-1, +1, +1), PLOT=FALSE)
```

```
Pradfoc(NULL, MEC , NULL, TRUE, rainbow(100) )
```

Preflect*Reflect a pole through to the lower hemisphere*

Description

Takes a vector to a pole and reflects it to the lower hemisphere

Usage

```
Preflect(az, dip)
```

Arguments

az	azimuth angle, degrees
dip	dip in degrees

Value

list	
az	azimuth angle, degrees
dip	dip in degrees
...	

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

REFLECT

Examples

```
z = Preflect(65, -23)
z = Preflect(265, -23)
```

 prepFOCS

Prepare Focals

Description

Prepare Focals for plotting. Program cycles through data and prepares a relevant data for further plotting and analysis.

Usage

```
prepFOCS(CMTSOL)
```

Arguments

CMTSOL see getCMT for the format for the input here.

Details

Used internally in spherefocgeo and ternfocgeo.

Value

List:

Paz	P-axis azimuth
Pdip	P-axis dip
Taz	T-axis azimuth
Tdip	T-axis dip
h	horizontal distance on ternary plot
v	vertical distance on ternary plot
fcols	focal color
LATS	latitudes
LONS	longitudes
IFcol	index of color
yr	year
JDHM	character identification
JDHMS	character identification

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

getCMT, spherefocgeo, ternfocgeo

printMEC	<i>Print focal mechanism</i>
----------	------------------------------

Description

Print focal mechanism

Usage

```
printMEC(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

x	Mechanism list
digits	digits for numeric information
...	standard printing parameters

Value

Side Effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

Examples

```
mc = CONVERTSDR(65, 32, -34 )
printMEC(mc)
```

PROJ3D	<i>Project 3D</i>
--------	-------------------

Description

Project a 3D body after rotation and translation

Usage

```
PROJ3D(aglyph, M = diag(1, nrow = 4), M2 = diag(1, nrow = 4),
       anorms = list(), zee = c(0, 0, 1))
```

Arguments

aglyph	glyph structure
M	rotation matrix
M2	rotation matrix
anorms	normals to structure
zee	Up direction of body

Details

This function takes a 3D body, rotates it and projects it for plotting. An example glyph is found in Z3Darrow.

Value

Glyph structure	
x, y, z	coordinates of rotated body faces
xp	rotated normal vectors
zd	depth mean value of each face

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

makeblock3D, ROTZ, ROTY, ROTX, BOXarrows3D, Z3Darrow, TRANmat

Examples

```

block1 = matrix(c(0,0,0,
  1,0,0,
  1,0.5,0,
  0,0.5,0,
  0,0,-2,
  1,0,-2,
  1,0.5,-2,
  0,0.5,-2), byrow=TRUE, ncol=3)

Bblock1 = makeblock3D(block1)

R3 = ROTX(-40)
R2 = ROTY(0)
R1 = ROTZ(20)
T = TRANmat(.1, 0, 0 )
M =      R1  %*% R2  %*% R3  %*% T

T2 = TRANmat(1, 0.5, 0 )
MT =      T2 %*%      R1  %*% R2  %*%      R3  %*% T

```

```
Z1 = PROJ3D(Bblock1$aglyph, M=MT, anorms=Bblock1$anorm , zee=c(0,0,1))
```

PTaxes	<i>Plot P-T axis on CLVD</i>
--------	------------------------------

Description

Plot P-T axis on CLVD

Usage

PTaxes(strike, dip, rake)

Arguments

strike	strike
dip	dip
rake	rake

Details

Lower Hemisphere. Add PT axes on a moment tensor plot

Value

Side effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

doNonDouble, MapNonDouble

Examples

```
mo = list(n=1, m1=1.035675e+017, m2=-1.985852e+016,
m3=-6.198052e+014, m4=1.177936e+017, m5=-7.600627e+016, m6=-3.461405e+017)
moments = cbind(mo$n, mo$m1, mo$m2, mo$m3, mo$m4, mo$m5, mo$m6)
doNonDouble(moments)
```

PTXY2*Plot P-T Axes*

Description

given a focal mechanism, add P-T lines to a plot

Usage

```
PTXY2(x = x, y = y, MEC, focsiz, pt = 0, ...)
```

Arguments

x	x-location on plot
y	y-location on plot
MEC	Focal Mechanism list from SDRFOC
focsiz	size of mechanism, inches
pt	pt = 0(plot both), 1=only P axes, 2=only T axes, default=0
...	graphical parameters

Details

This is a summary plot to be used instead of Beach Balls.

Value

Graphical Side Effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

Lees, J. M., Geotouch: Software for Three and Four Dimensional GIS in the Earth Sciences, Computers & Geosciences, 26, 7, 751-761, 2000.

See Also

nipXY, justfocXY

Examples

```

#### Haiti Earthquake Jan, 2010
MEC <- SDRfoc(71, 64, 25 , u=FALSE, ALIM=c(-1,-1, +1, +1), PLOT=FALSE)
plot(c(0, 1), c(0,1), type='n', asp=1)
u <- par("usr")

justfocXY(MEC, x=.5, y= .5, focsiz=0.5,
fcol ='brown' , fcolback = "white", xpd = TRUE)

PTXY2(1.0, .5 , MEC ,0.5, col="purple", lwd=3 )

nipXY(MEC, x = 0.25, y = .5, focsiz=0.5,
fcol ='purple', nipcol = "black", cex = 0.4)
##### or
set.seed(2015)
N = 20
lon=runif(20, 268.1563 , 305)
lat=runif(20, 7.593004, 25.926045)
str1=runif(20,50,100)
dip1=runif(20,10, 80)
rake1=runif(20,5, 180)

dep=runif(20,1,15)
name=seq(from=1, to=length(lon), by=1)
Elat=NULL
Elon=NULL
yr = rep(2017, times=N)
jd = runif(N, min=1, max=365)

MEKS = list(lon=lon, lat=lat, str1=str1, dip1=dip1,
rake1=rake1, dep=dep, name=name, yr=yr, jd = jd)

PROJ = GEOMap::setPROJ(type=2, LAT0=mean(lat) , LON0=mean(lon) ) ## utm

XY = GEOMap::GLOB.XY(lat, lon, PROJ)

plot(range(XY$x), range(XY$y), type='n', asp=1)

for(i in 1:length(XY$x))
{
  Msdr = CONVERTSDR(MEKS$str1[i], MEKS$dip1[i],MEKS$rake1[i])
  MEC = MRake(Msdr$M)
  MEC$UP = FALSE

  jcol = foc.color(foc.icolor(MEC$rake1), pal=1)

  PTXY2(XY$x[i], XY$y[i] , MEC ,focsiz=0.5, col=jcol, lwd=3)
}

```

qpoint	<i>Point on Stereonet</i>
--------	---------------------------

Description

Plot a set of (azimuths, takeoff) angles on a stereonet.

Usage

```
qpoint(az, iang, col = 2, pch = 5, lab = "", POS = 4, UP = FALSE, PLOT = FALSE, cex = 1)
```

Arguments

az	vector of azimuths, degrees
iang	vector of incident angles, degrees
col	color
pch	plotting character
lab	text labels
POS	position for labels
UP	logical, TRUE=upper
PLOT	logical, add to existing plot
cex	character expansion of labels

Details

The iang argument represents the takeoff angle, and is measured from the nadir (z-axis pointing down).

Value

List	
x	coordinate on plot
y	coordinate on plot

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

FixDip, focpoint

Examples

```
d = runif(10, 0, 90)
a = runif(10, 0, 360)
net()
qpoint(a, d)
```

radiateP

Plot radiation pattern for P-waves

Description

Plots focal mechanism and makes radiation plot with mark up

Usage

```
radiateP(MEC, SCALE = FALSE, col = col, TIT = FALSE)
```

Arguments

MEC	focal mechanism structure
SCALE	logical, TRUE=add scale
col	color palette
TIT	title for plot

Value

Used for side graphical effect

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

radP, SDRfoc

Examples

```
MEC =SDRfoc(65,25,13, u=FALSE, ALIM=c(-1,-1, +1, +1), PLOT=FALSE)
radiateP(MEC, SCALE = FALSE, col = rainbow(100) , TIT = FALSE)
```

radiateSH	<i>Plot radiation pattern for SH-waves</i>
-----------	--

Description

Plots focal mechanism and makes radiation plot with mark up

Usage

```
radiateSH(MEC, SCALE = FALSE, col = col, TIT = FALSE)
```

Arguments

MEC	focal mechanism structure
SCALE	logical, TRUE=add scale
col	color palette
TIT	title for plot

Value

Used for side graphical effect

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

radSH, SDRfoc

Examples

```
MEC =SDRfoc(65,25,13, u=FALSE, ALIM=c(-1,-1, +1, +1), PLOT=FALSE)
radiateSH(MEC, SCALE = FALSE, col = rainbow(100) , TIT = FALSE)
```

`radiateSV`*Plot radiation pattern for SV-waves*

Description

Plots focal mechanism and makes radiation plot with mark up

Usage

```
radiateSV(MEC, SCALE = FALSE, col = col, TIT = FALSE)
```

Arguments

MEC	focal mechanism structure
SCALE	logical, TRUE=add scale
col	color palette
TIT	title for plot

Value

Used for side graphical effect

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

radSV, SDRfoc

Examples

```
MEC =SDRfoc(65,25,13, u=FALSE, ALIM=c(-1,-1, +1, +1), PLOT=FALSE)
radiateSV(MEC, SCALE = FALSE, col = rainbow(100) , TIT = FALSE)
```

radP	<i>Radiation pattern for P waves</i>
------	--------------------------------------

Description

calculate the radiation patterns for P waves

Usage

```
radP(del, phiS, lam, ichi, phi)
```

Arguments

del	degrees, angle
phiS	degrees,angle
lam	degrees, angle
ichi	degrees, take off angle
phi	degrees, take off azimuth

Details

Given a focal mechanism strike-dip-rake and a given incident angle (take-off angle) and azimuth, return the P amplitude

Value

Amplitude of the P wave

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

References

K.~Aki and P.~G. Richards.*Quantitative seismology*. University Science Books, Sausalito, Calif., 2nd edition, 2002.

See Also

radP, radSV, imageP

Examples

```
phiS=65
del=25
lam=13
x = seq(-1, 1, 0.01)
y = x

X = matrix(rep(x, length(y)), nrow= length(x))
Y = t(X)
RAD2DEG = 180/pi
p = RAD2DEG*(pi/2 -atan2(Y, X))
p[p<0] = p[p<0] + 360

R = sqrt(X^2+Y^2)
R[R>1] = NaN
dip =RAD2DEG*2*asin(R/sqrt(2))

### Calculate the radiation pattern
G = radP(del, phiS, lam, dip, p)

### plot values
image(x,y,G, asp=1)
```

radSH	<i>Radiation pattern for SH waves</i>
-------	---------------------------------------

Description

calculate the radiation patterns for SH waves

Usage

```
radSH(del, phiS, lam, ichi, phi)
```

Arguments

del	degrees, angle
phiS	degrees,angle
lam	degrees, angle
ichi	degrees, take off angle
phi	degrees, take off azimuth

Details

Given a focal mechanism strike-dip-rake and a given incident angle (take-off angle) and azimuth, return the SH amplitude

Value

Amplitude of the SH wave

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

References

K.~Aki and P.~G. Richards. *Quantitative seismology*. University Science Books, Sausalito, Calif., 2nd edition, 2002.

See Also

radP, radSV, imageSH

Examples

```
phiS=65
del=25
lam=13
x = seq(-1, 1, 0.01)
y = x

X = matrix(rep(x, length(y)), nrow= length(x))
Y = t(X)
RAD2DEG = 180/pi
p = RAD2DEG*(pi/2 -atan2(Y, X))
p[p<0] = p[p<0] + 360

R = sqrt(X^2+Y^2)
R[R>1] = NaN
dip =RAD2DEG*2*asin(R/sqrt(2))

### Calculate the radiation pattern
G = radSH(del, phiS, lam, dip, p)

### plot values
image(x,y,G, asp=1)
```

radSV

Radiation pattern for SV waves

Description

calculate the radiation patterns for SV waves

Usage

```
radSV(del, phiS, lam, ichi, phi)
```

Arguments

del	degrees, angle
phiS	degrees, angle
lam	degrees, angle
ichi	degrees, take off angle
phi	degrees, take off azimuth

Details

Given a focal mechanism strike-dip-rake and a given incident angle (take-off angle) and azimuth, return the SV amplitude

Value

Amplitude of the SV wave

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

References

K.~Aki and P.~G. Richards. *Quantitative seismology*. University Science Books, Sausalito, Calif., 2nd edition, 2002.

See Also

radP, radSH, imageSV

Examples

```
phiS=65
del=25
lam=13
x = seq(-1, 1, 0.01)
y = x

X = matrix(rep(x, length(y)), nrow= length(x))
Y = t(X)
RAD2DEG = 180/pi
p = RAD2DEG*(pi/2 -atan2(Y, X))
p[p<0] = p[p<0] + 360

R = sqrt(X^2+Y^2)
R[R>1] = NaN
```

```
dip =RAD2DEG*2*asin(R/sqrt(2))

### Calculate the radiation pattern
G = radSV(del, phiS, lam, dip, p)

### plot values
image(x,y,G, asp=1)
```

rakelegend	<i>Focal Legend based on rake</i>
------------	-----------------------------------

Description

Focal Legend based on rake

Usage

```
rakelegend(corn="topright", pal=1)
```

Arguments

corn	position of legend, default="topright"
pal	palette number, default=1

Details

Colors are based on earlier publication of Geotouch program.

For pal = 1, colors are , DarkSeaGreen, cyan1, SkyBlue1, RoyalBlue, GreenYellow, orange, red.

Value

Graphical Side Effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

Lees, J. M., (1999) Geotouch: Software for Three and Four-Dimensional GIS in the Earth Sciences, Computers and Geosciences, 26(7) 751-761.

See Also

foc.color,focleg

Examples

```
plot(c(0,1), c(0,1), type='n')  
  
rakelegend(corn="topleft", pal=1)
```

readCMT	<i>Read Harvard CMT moment</i>
---------	--------------------------------

Description

Read and plot a CMT solution copied from the Harvard CMT website.

Usage

```
readCMT(filename, PLOT=TRUE)
```

Arguments

filename	character, file name
PLOT	Logical, TRUE=plot mechanisms sequentially

Details

Uses the standard output format.

Value

List of mechanisms and graphical Side effects. Each element in the list consists of a list including: FIRST,yr,mo,dom,hr,mi,sec,name,tshift,half,lat,lon,z,Mrr,Mtt,Mpp,Mrt,Mrp,Mtp. The FIRST element is simply a duplicate of the PDE solution card.

Note

Other formats are available.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

Ekstrom, G.; Nettles, M. and DziewoDski, A. The Global CMT Project 2004-2010: centroid-moment tensors for 13,017 earthquakes Physics of the Earth and Planetary Interiors, 2012.

See Also

doNonDouble, MapNonDouble

Examples

```

## Not run:
Hcmt = readCMT("CMT_FULL_FORMAT.txt")

##### or,

Hcmt = readCMT("CMT_FULL_FORMAT.txt", PLOT=FALSE)

moments = matrix(ncol=7, nrow=length(Hcmt))
Locs = list(y=vector(length=length(Hcmt)) ,x=vector(length=length(Hcmt)))

for(i in 1:length(Hcmt))
{
P1 = Hcmt[[i]]
##### Note the change of sign for cartesian coordinates
moments[i,] = cbind(i, P1$Mtt, P1$Mpp, P1$Mrr,
                    -P1$Mrp, P1$Mrt , -P1$Mtp)
Locs$y[i] = P1$lat
Locs$x[i] = P1$lon
}

mlon = mean(Locs$x, na.rm=TRUE)
mlat = mean(Locs$y, na.rm=TRUE)

PROJ = GEOMap::setPROJ(type = 1, LAT0 = mlat , LON0 = mlon)
Glocs = GEOMap::GLOB.XY(Locs$y, Locs$x, PROJ      )

LIMlat = expandbound(Locs$y)
LIMlon = expandbound(Locs$x)

PLAT = pretty(LIMlat)
PLON = pretty(LIMlon)

data(worldmap)
par(xpd=FALSE)

plotGEOMapXY(worldmap, LIM = c(LIMlon[1],LIMlat[1] ,LIMlon[2],LIMlat[2]) ,
              PROJ=PROJ, axes=FALSE, xlab="", ylab="" )

### add tic marks
kbox = GEOMap::GLOB.XY(PLAT,PLON, PROJ)

      sqrTICXY(kbox , PROJ, side=c(1,2,3,4), LLgrid=TRUE, col=grey(.7) )

##### add focal mechs

MapNonDouble(Glocs, moments, col=NULL, add=TRUE)

```

```
## End(Not run)
```

RectDense

Divide a region into rectangles based on density

Description

Given a set of (x,y) points, partition the field into rectangles each containing a minimum number of points

Usage

```
RectDense(INx, INy, icut = 1, u = par("usr"), ndivs = 10)
```

Arguments

INx	x-coordinates
INy	y-coordinates
icut	cut off for number of points
u	user coordinates
ndivs	number of divisions in x-coordinate

Details

Based on the user coordinates as returned from par('usr'). Each rectangular region is tested for the number of points that fall within icut or greater.

Value

List:

icorns	matrix of corners that passed test
ilens	vector,number of points in each icorns box
ipass	vector, index of the corners that passed icut
corners	matrix of all corners
lens	vector,number of points for each box

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

Examples

```

x = rnorm(100)
y = rnorm(100)

plot(x,y)
u = par('usr')
RI = RectDense(x, y, icut=3, u=u, ndivs=10)

rect(RI$icorns[,1],RI$icorns[,2],RI$icorns[,3],RI$icorns[,4], col=NA, border='blue')

```

REFLECT

*reflect pole***Description**

Reflect pole to lower hemisphere

Usage

```
REFLECT(A)
```

Arguments

A structure of azimuth and Dips in degrees

Value

list of:cartesian coordinates of reflected pole

x	x-coordinate
y	y-coordinate
z	z-coordinate
az	azimuth, degrees
dip	dip, degrees

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

Preflect

Examples

```
A = list(az=231, dip = -65)
REFLECT(A)
```

rotateFoc

Rotate Focal Mechanism

Description

Rotate mechanism to vertical plan at specified angle

Usage

```
rotateFoc(MEX, phi)
```

Arguments

MEX	Focal Mechanism list
phi	angle in degrees

Details

Assumed vertical plane, outer hemisphere

Value

Focal Mechanism

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

plotfoc, SDRfoc, Beachfoc, TEACHFOC, plotmanyfoc, getUWfocs

Examples

```
a1 = SDRfoc(90, 90, 90, u = TRUE , PLOT = TRUE)

par(mfrow=c(2,2))

SDRfoc(a1$az1, a1$dip1, a1$rake1, u = TRUE, PLOT = TRUE)
ra1 = rotateFoc(a1, -90)
```

```
SDRfoc(ra1$az1, ra1$dip1, ra1$rake1, u = TRUE , PLOT = TRUE)
```

```
ra1 = rotateFoc(a1, 0)
```

```
SDRfoc(a1$az1, a1$dip1, a1$rake1, u = TRUE, PLOT = TRUE)
```

```
SDRfoc(ra1$az1, ra1$dip1, ra1$rake1, u = TRUE , PLOT = TRUE)
```

Rotfocphi	<i>Rotate Focal Mechanism</i>
-----------	-------------------------------

Description

Rotate Focal Mechanism into the vertical plane by a certain number of degrees

Usage

```
Rotfocphi(phi, urot, udip, vrot, vdip, az1, d1, az2, d2, prot, pdip, trot, tdip)
```

Arguments

phi	degrees in plane to rotate
urot	U-vector azimuth
udip	U-vector dip
vrot	V-vector azimuth
vdip	V-vector dip
az1	First plane - azimuth
d1	First plane - dip
az2	Second plane - azimuth
d2	Second plane - dip
prot	P-axis azimuth
pdip	P-axis dip
trot	T-axis azimuth
tdip	T-axis dip

Details

Rotate the focal mech by phi degrees

Value

list:

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

xsecmanyfoc, rotateFoc

 RotTP

Rotate T-P axes

Description

Rotate T-P axes

Usage

RotTP(rotmat, strk1, dip1)

Arguments

rotmat	rotation matrix, 3 by 3
strk1	strike angle
dip1	dip angle

Details

These are used as functions auxially to rotateFoc.

Value

list:

strk	strike angle
dip	dip angle

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

Rotfocphi, TP2XYZ

Examples

```
phi = 18

MAT = JMAT(phi)

RotTP(MAT, 30, 40)
```

ROTX

X-axis Rotation Matrix

Description

Matrix rotation about the X-axis

Usage

ROTX(deg)

Arguments

deg Angle in degrees

Value

A 4 by 4 matrix for rotation and translation for 3-D transformation

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

References

Rogers and Adams, 1990, Mathematical Elements for Computer Graphics, McGraw-Hill, 611p.

See Also

ROTY, ROTZ

Examples

```
v = c(1,4,5)
A = ROTX(23)
vp = c(v, 1)
```

rotx3	<i>Rotate about the x axis</i>
-------	--------------------------------

Description

3x3 Rotation about the x axis

Usage

rotx3(deg)

Arguments

deg angle, degrees

Details

returns a 3 by 3 rotation matrix

Value

matrix, 3 by 3

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

roty3, rotz3, ROTX, ROTZ, ROTY

Examples

```
a = 45
rotx3(a)
```

ROTY	<i>Y-axis Rotation Matrix</i>
------	-------------------------------

Description

Matrix rotation about the Y-axis

Usage

ROTY(deg)

Arguments

deg	Angle in degrees
-----	------------------

Value

A 4 by 4 matrix for rotation and translation for 3-D transformation

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

References

Rogers and Adams, 1990, Mathematical Elements for Computer Graphics, McGraw-Hill, 611p.

See Also

ROTX, ROTZ

Examples

```
v = c(1,4,5)
A = ROTY(23)
vp = c(v, 1)
```

roty3	<i>Rotate about the y axis</i>
-------	--------------------------------

Description

3x3 Rotation about the y axis

Usage

roty3(deg)

Arguments

deg angle, degrees

Details

returns a 3 by 3 rotation matrix

Value

matrix, 3 by 3

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

rotz3, rotx3, ROTX, ROTZ, ROTY

Examples

```
a = 45
roty3(a)
```

ROTZ*Z-axis Rotation Matrix*

Description

Matrix rotation about the Z-axis

Usage

ROTZ(deg)

Arguments

deg Angle in degrees

Value

A 4 by 4 matrix for rotation and translation for 3-D transformation

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

References

Rogers and Adams, 1990, Mathematical Elements for Computer Graphics, McGraw-Hill, 611p.

See Also

ROTX, ROTY

Examples

```
v = c(1,4,5)
A = ROTZ(23)
vp = c(v, 1)
```

rotz3	<i>Rotate about the z axis</i>
-------	--------------------------------

Description

3x3 Rotation about the z axis

Usage

rotz3(deg)

Arguments

deg angle, degrees

Details

returns a 3 by 3 rotation matrix

Value

matrix, 3 by 3

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

roty3, rotx3, ROTX, ROTZ, ROTY

Examples

```
a = 45
rotz3(a)
```

SDRfoc

*Plot a Focal Mechanism from SDR***Description**

Given Strike-Dip-Rake plot a focal mechanism

Usage

```
SDRfoc(s, d, r, u = FALSE, ALIM = c(-1, -1, +1, +1), PLOT = TRUE)
```

Arguments

s	strike, degrees
d	dip, degrees
r	rake, degrees
u	logical, TRUE=upper hemisphere
ALIM	bounding box on plot
PLOT	logical, TRUE=add to plot

Details

The ALIM vector allows one to zoom into portions of the focal mechanism for details when points are tightly clustered.

Value

MEC structure

Note

Basic MEC List Structure

az1	azimuth angle plane 1, degrees
dip1	dip angle plane 1, degrees
az2	azimuth angle plane 2, degrees
dip2	dip angle plane 2, degrees
dir	0,1 to determine which section of focal sphere is shaded
rake1	rake angle plane 1, degrees
dipaz1	dip azimuth angle plane 1, degrees
rake2	rake angle plane 2, degrees
dipaz2	dip azimuth angle plane 2, degrees
P	pole list(az, dip) P-axis
T	pole list(az, dip) T-axis
U	pole list(az, dip) U-axis
V	pole list(az, dip) V-axis

F	pole list(az, dip) F-axis
G	pole list(az, dip) G-axis
sense	0,1 to determine which section of focal sphere is shaded
M	list of focal parameters used in some calculations
UP	logical, TRUE=upper hemisphere
icol	index to suite of colors for focal mechanism
ileg	Kind of fault
fcol	color of focal mechanism
CNVRG	Character, note on convergence of solution
LIM	vector plotting region (x1, y1, x2, y2)

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

CONVERTSDR

Examples

```
M = SDRfoc(-25, 34, 16,u = FALSE, ALIM = c(-1, -1, +1, +1), PLOT=TRUE)
```

ShadowCLVD

Plot CLVD focal mechanism

Description

Plot non-double couple part of the focal mechanism provided in the moment tensor.

Usage

```
ShadowCLVD(m, PLOT = TRUE, col=rgb(1, .75, .75))
```

Arguments

m	moment tensor
PLOT	logical, TRUE means plot
col	color, either a single color, rgb, or a color palette

Details

This code is meant to be used with `doNonDouble` or `MapNonDouble` functions for plotting the non-double couple components of the moment tensor. A color palette can be provided for some details of the radiation patterns, e.g. `col=rainbow(12)`.

Value

Side effects and image list

Note

Lower Hemisphere.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

doNonDouble, MapNonDouble

Examples

```
##### moment tensor from Harvard CMT catalog
sponent = 26
ef = 1*10^(sponent)
Mrr = 2.375*ef
Mtt = -2.777*ef
Mpp = 0.403*ef
Mrt = 2.800*ef
Mrp = 1.190*ef
Mtp = -0.539*ef

##### convert to cartesian coordinates
Mzz=Mrr
Mxx= Mtt
Myy= Mpp
Mxz= Mrt
Myz= -Mrp
Mxy= -Mtp

m=matrix( c(Mxx,Mxy,Mxz,
            Mxy,Myy,Myz,
            Mxz,Myz,Mzz), ncol=3, byrow=TRUE)

Fi=seq(from=0, by=0.1, to=361)
### dev.new()
plot(cos(Fi*pi/180.0),sin(Fi*pi/180.0),type='l', asp=1 , ann=FALSE, axes=FALSE)

ShadowCLVD(m, col='red')
```

SHradfoc

*Plot SH-wave radiation***Description**

Plot SH-wave radiation with information from the pickfile and waveform data

Usage

```
SHradfoc(A, MEC, GU, pscale, col)
```

Arguments

A	Pickfile structure
MEC	MEC structure
GU	Waveform Event Structure
pscale	logical (not used)
col	color palette

Details

Image plot of the SH radiation pattern

Value

Graphical Side effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

imageSH

Examples

```
MEC = SDRfoc(65, 32, -34, u=TRUE, ALIM=c(-1,-1, +1, +1), PLOT=FALSE)

SHradfoc(NULL, MEC , NULL, TRUE, rainbow(100) )
```

Source Type	<i>Moment Tensor Source Type</i>
-------------	----------------------------------

Description

Given a vector of EigenValues, extract the source type.

Usage

```
Source Type(v)
```

Arguments

v vector of decreasing eigenvalues

Details

plotting for -30 to 30 degree quadrant.

Value

list:

phi latitude angle in degrees
lam longitude angle in degrees

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

Tape, W., and C. Tape (2012), A geometric comparison of source-type plots for moment tensors, Geophys. J. Int., 190, 499-510.

See Also

HAMMERprojXY, TapeBase, TapePlot

Examples

```
Source Type(c(1,-1,1) )

T1 = TapeBase()

m1 = list(Mxx=1.543, Mxy=0.786, Myy=0.336, Mxz=-2.441, Myz=0.353, Mzz=0.961)

i = 1
M1=matrix( c(m1$Mxx[i],m1$Mxy[i],m1$Mxz[i],
```

```

      m1$Mxy[i],m1$Myy[i],m1$Myz[i],
      m1$Mxz[i],m1$Myz[i],m1$Mzz[i]), ncol=3, byrow=TRUE)

E1 = eigen(M1)
  h = SourceType( sort(E1$values, decreasing=TRUE) )
  h$dip = 90-h$phi
  ## cat(paste(h$dip, h$lam, sep=" "), sep="\n")
  h1 = HAMMERprojXY(h$dip*pi/180, h$lam*pi/180)

TapePlot(T1)
  points(h1$x, h1$y, pch=21, bg="red" )

```

spherefocgeo

SphereFocGeo

Description

Spherical Projections of PT axes distributed geographically.

Usage

```

spherefocgeo(CMTSOL, PROJ = NULL, icut = 5,
ndivs = 10, bbox=c(0,1, 0, 1), PLOT = TRUE,
add = FALSE, RECT = FALSE, pal = terrain.colors(100))

```

Arguments

CMTSOL	see output of getCMT for list input
PROJ	Map projection
icut	cut off for number of points in box, default=5
ndivs	divisions of map area, default=10
bbox	bounding box for dividing the area, given as minX, maxX, minY, maxY; default=usr coordinates from par()
PLOT	logical, default=TRUE
add	logical, add to existing plot
RECT	logical, TRUE=plot rectangles
pal	palette fo rimages in each box

Details

Program divides the area into blocks, tests each one for minimum number per block and projects the P and T axes onto an equal area stereonet.

Value

Graphical Side Effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

PlotPTsmooth, ternfocgeo, prepFOCS, RectDense

Examples

```

N = 100
LATS = c(7.593004, 25.926045)
LONS = c(268.1563 , 305)
lon=rnorm(N, mean=mean(LONS), sd=diff(LONS)/2 )
lat=rnorm(N, mean=mean(LATS), sd=diff(LATS)/2)

str1=runif(N,50,100)
dip1=runif(N,10, 80)
rake1=runif(N,5, 180)

dep=runif(N,1,15)
name=seq(from=1, to=length(lon), by=1)
Elat=NULL
Elon=NULL
yr = rep(2017, times=N)
jd = runif(N, min=1, max=365)

MEKS = list(lon=lon, lat=lat, str1=str1, dip1=dip1,
rake1=rake1, dep=dep, name=name, yr=yr, jd = jd)

PROJ = GEOMap::setPROJ(type=2, LAT0=mean(lat) , LON0=mean(lon) ) ## utm
XY = GEOMap::GLOB.XY(lat, lon, PROJ)
plot(range(XY$x), range(XY$y), type='n', asp=1)

points(XY$x, XY$y)
spherefocgeo(MEKS, PROJ, PLOT=TRUE, icut = 3, ndivs = 4,
add=TRUE, pal=terrain.colors(100), RECT=TRUE )

```

```
## Not run:

plot(x=range(IZ$x), y=range(IZ$y), type='n', asp=1, axes=FALSE, ann=FALSE)

image(x=IZ$x, y=IZ$y, z=(UZ), col=blues, add=TRUE)

image(x=IZ$x, y=IZ$y, z=(AZ), col=terrain.colors(100) , add=TRUE)

plotGEOMapXY(haiti.map,
              LIM = c(Lon.range[1],Lat.range[1] ,
Lon.range[2] ,Lat.range[2]),
              PROJ =PROJ, MAPstyle = 2,
              MAPcol = 'black' , add=TRUE )

H = rectPERIM(JMAT$x0, JMAT$y0)

antipolygon(H$x ,H$y,   col=grey(.85) , corner=1, pct=.4)

sqrTICXY(H , PROJ, side=c(1,2,3,4),   LLgrid=TRUE, col=grey(.7) )

spherefocgeo(OLDCMT, PROJ, PLOT=TRUE, add=TRUE, pal=topo.colors(100) )

## End(Not run)
```

spline.arrow

Spline Arrow

Description

Given a set of points, draw a spline and affix an arrow at the end.

Usage

```
spline.arrow(x, y = 0, kdiv = 20, arrow = 1,
             length = 0.2, col = "black", thick = 0.01,
             headlength = 0.2, headthick = 0.1, code = 2, ...)
```

Arguments

x	vector, x-coordinates
y	vector, y-coordinates
kdiv	Number of divisions
arrow	style of arrow, 1=simple arrow, 2=fancy arrow
length	length of head
col	color of arrow
thick	thickness of arrow stem
headlength	length of arrow head
headthick	thickness of arrow head
code	code, 1=arrow on end of spline, 3=arrow on beginning.
...	graphical parameters for the line

Details

Can use either simple arrows or fancy arrows.

Value

list of x,y coordinates of the spline and Graphical Side effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

fancyarrows

Examples

```
plot(c(0,1), c(0,1), type='n')

G=list()
G$x=c(0.1644,0.1227,0.0659,0.0893,0.2346,
0.3514,0.5518,0.7104,0.6887,0.6903,0.8422)
G$y=c(0.8816,0.8305,0.7209,0.6086,0.5372,
0.6061,0.6545,0.6367,0.4352,0.3025,0.0475)

spline.arrow(G$x, G$y)
```

StrikeDip*Plot Strike Dip Lines*

Description

Given a focal mechanism, add Strike Dip lines to a plot.

Usage

```
StrikeDip(x = x, y = y, MEC, focsiz, addDIP = TRUE, ...)
```

Arguments

x	x-location on plot
y	y-location on plot
MEC	Focal Mechanism list from SDRFOC
focsiz	size of mechanism, inches
addDIP	Logical, TRUE = add dip line perpendicular to strike
...	graphical parameters

Details

This is a summary plot to be used instead of Beach Balls.

Value

Graphical Side Effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

Lees, J. M., Geotouch: Software for Three and Four Dimensional GIS in the Earth Sciences, Computers & Geosciences, 26, 7, 751-761, 2000.

See Also

nipXY, justfocXY, plotmanyfoc

Examples

```

### Haiti Earthquake Jan, 2010
MEC <- SDRfoc(71, 64, 25 , u=FALSE, ALIM=c(-1,-1, +1, +1), PLOT=FALSE)
plot(c(0, 1), c(0,1), type='n', asp=1)
u <- par("usr")
focsiz <- 0.5
justfocXY(MEC, x=.5, y= .5, focsiz=0.5,
fcol ='brown' , fcolback = "white", xpd = TRUE)
StrikeDip(1.0, .5 , MEC ,focsiz, col="purple", lwd=3 )
nipXY(MEC, x = 0.25, y = .5, focsiz=0.5,
fcol ='purple', nipcol = "black", cex = 1)

##### or
set.seed(2015)
N = 20
lon=runif(20, 268.1563 , 305)
lat=runif(20, 7.593004, 25.926045)
str1=runif(20,50,100)
dip1=runif(20,10, 80)
rake1=runif(20,5, 180)

dep=runif(20,1,15)
name=seq(from=1, to=length(lon), by=1)
Elat=NULL
Elon=NULL
yr = rep(2017, times=N)
jd = runif(N, min=1, max=365)

MEKS = list(lon=lon, lat=lat, str1=str1, dip1=dip1,
rake1=rake1, dep=dep, name=name, yr=yr, jd = jd)

PROJ = GEOMap::setPROJ(type=2, LAT0=mean(lat) , LON0=mean(lon) ) ## utm

XY = GEOMap::GLOB.XY(lat, lon, PROJ)

plot(range(XY$x), range(XY$y), type='n', asp=1)

for(i in 1:length(XY$x))
{
  Msdr = CONVERTSDR(MEKS$str1[i], MEKS$dip1[i],MEKS$rake1[i])
  MEC = MRake(Msdr$M)
  MEC$UP = FALSE

  jcol = foc.color(foc.icolor(MEC$rake1), pal=1)

StrikeDip(XY$x[i], XY$y[i] , MEC ,focsiz, col=jcol, lwd=3 )

}

```

strikeslip.fault	<i>Strikeslip Fault Cartoon</i>
------------------	---------------------------------

Description

Illustrate a strikeslip fault using animation

Usage

```
strikeslip.fault(anim = seq(from = 0, to = 1, by = 0.1), KAPPA = 2,
                 Light = c(45, 45))
```

Arguments

anim	animation vector
KAPPA	Phong parameter for lighting
Light	lighting point

Details

Program will animate a strikeslip fault for educational purposes. Animation must be stopped by halting execution.

Value

Graphical Side effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

normal.fault, thrust.fault

Examples

```
strikeslip.fault(anim=0, Light=c(45,90) )

## Not run:
#### execute a stop command to stop this animation
anim= seq(from=0, to=1, by=.1)
strikeslip.fault(anim=anim, Light=c(45,90) )

## End(Not run)
```

SVradfoc*Plot SV-wave radiation*

Description

Plot SV-wave radiation with information from the pickfile and waveform data

Usage

```
SVradfoc(A, MEC, GU, pscale, col)
```

Arguments

A	Pickfile structure
MEC	MEC structure
GU	Waveform Event Structure
pscale	logical (not used)
col	color palette

Details

Image plot of the SV radiation pattern

Value

Graphical Side effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

imageSV

Examples

```
MEC = SDRfoc(65, 32, -34, u=TRUE, ALIM=c(-1,-1, +1, +1), PLOT=FALSE)

SVradfoc(NULL, MEC , NULL, TRUE, rainbow(100) )
```

TapeBase

Tape Base Lines

Description

Create a structure of Tape Base lines

Usage

TapeBase()

Details

Program returns the lines and points for plotting a Tape plot. Based on the Hammer projection.

Value

List

Note

The list includes points and other information

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

Tape, W., and C. Tape (2012), A geometric comparison of source-type plots for moment tensors, *Geophys. J. Int.*, 190, 499-510.

See Also

TapePlot, HAMMERprojXY

Examples

```
T1 =TapeBase()  
TapePlot(T1)
```

TapePlot	<i>Tape style Lune Plot</i>
----------	-----------------------------

Description

Tape style Lune Plot using Hammer projection

Usage

```
TapePlot(TapeList = list(), add = FALSE, ann = TRUE,  
pcol = c(grey(0), grey(0.85), grey(0.95)))
```

Arguments

TapeList	List of strokes from TapeBase
add	logical, TRUE=add to existing plot
ann	logical, TRUE=annotape
pcol	3-vector of colors: inner lines, upper polygon, lower polygon

Details

Plot an Tape net from the TapeBase function.

Value

Side effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

Tape, W., and C. Tape (2012), A geometric comparison of source-type plots for moment tensors, Geophys. J. Int., 190, 499-510. <https://doi.org/10.1111/j.1365-246X.2012.05490.x>

See Also

TapeBase, HAMMERprojXY

Examples

```

T1 = TapeBase()
TapePlot(T1)

data(widdenMoments)
WM = widdenMoments

par(mfrow=c(1,1), mai=c(0,0,0,0))
T1 = TapeBase()
TapePlot(T1)

for(i in 1:length(WM$Mxx))
{
  M1=matrix( c(WM$Mxx[i],WM$Mxy[i],WM$Mxz[i],
WM$Mxy[i],WM$Myy[i],WM$Myz[i],
WM$Mxz[i],WM$Myz[i],WM$Mzz[i]), ncol=3, byrow=TRUE)

  E1 = eigen(M1)
  h = SourceType( sort(E1$values, decreasing=TRUE) )
  h$dip = 90-h$phi
  ## cat(paste(h$dip, h$lam, sep=" "), sep="\n")
  h1 = HAMMERprojXY(h$dip*pi/180, h$lam*pi/180)

  points(h1$x, h1$y, pch=21, bg="orange" )
}

```

TEACHFOC

Graphical Plot of Focal Mechanism

Description

Plots Beachball figure with numerous vectors and points added and labeled. Useful for teaching about focal mechanisms.

Usage

```
TEACHFOC(s, d, r, up = FALSE)
```

Arguments

s	strike
d	dip
r	rake
up	logical, TRUE = upper

Value

Graphical side effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

CONVERTSDR, MRake,foc.icolor,focleg, foc.color, focpoint, PlotPlanes, nipXY , fancyarrows

Examples

```
TEACHFOC(65, 32, -34, up=TRUE)
```

ternfoc.point

Plot Ternary Point

Description

Add a point to a ternary plot

Usage

```
ternfoc.point(deltaB, deltaP, deltaT)
```

Arguments

deltaB	angle, degrees
deltaP	angle, degrees
deltaT	angle, degrees

Details

Plot point on a Ternary diagram using Froelich's algorithm.

Value

List

h	vector of x coordinates
v	vector of y coordinates

Note

Use Bfocvec(az1, dip1, az2, dip2) to get the deltaB angle.

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

References

C. Frohlich. Triangle diagrams: ternary graphs to display similarity and diversity of earthquake focal mechanisms. *Physics of the Earth and Planetary Interiors*, 75:193-198, 1992.

See Also

Bfocvec

Examples

```
Msdr = CONVERTSDR(55.01, 165.65, 29.2 )
MEC = MRake(Msdr$M)
MEC$UP = FALSE
az1 = Msdr$M$az1
dip1 = Msdr$M$d1
az2 = Msdr$M$az2
dip2 = Msdr$M$d2
BBB = Bfocvec(az1, dip1, az2, dip2)
V = ternfoc.point(BBB$Bdip, Msdr$M$pd, Msdr$M$td )
```

ternfocgeo	<i>Ternary Focals</i>
------------	-----------------------

Description

Ternary plots of rake categories (strike-slip, normal, thrust) distributed geographically.

Usage

```
ternfocgeo(CMTSOL, PROJ = NULL, icut = 5, ndivs = 10,
  bbox=c(0,1, 0, 1), PLOT = TRUE, add = FALSE, RECT = FALSE)
```

Arguments

CMTSOL	see output of getCMT for list input
PROJ	Map projection
icut	cut off for number of points in box, default=5
ndivs	divisions of map area, default=10
bbox	bounding box for dividing the area, given as minX, maxX, minY, maxY; default=usr coordinates from par()
PLOT	logical, default=TRUE
add	logical, add to existing plot
RECT	logical, TRUE=plot rectangles

Details

Program divides the area into blocks, tests each one for minimum number per block and plots a ternary plot for each block.

Value

Graphical Side Effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

PlotTernfoc, spherefocgeo, prepFOCS, RectDense

Examples

```

N = 100
LATS = c(7.593004, 25.926045)
LONS = c(268.1563, 305)
lon=rnorm(N, mean=mean(LONS), sd=diff(LONS)/2 )
lat=rnorm(N, mean=mean(LATS), sd=diff(LATS)/2)

str1=runif(N,50,100)
dip1=runif(N,10, 80)
rake1=runif(N,5, 180)

dep=runif(N,1,15)
name=seq(from=1, to=length(lon), by=1)
Elat=NULL
Elon=NULL
yr = rep(2017, times=N)
jd = runif(N, min=1, max=365)

MEKS = list(lon=lon, lat=lat, str1=str1, dip1=dip1,
rake1=rake1, dep=dep, name=name, yr=yr, jd = jd)
PROJ = GEOMap::setPROJ(type=2, LAT0=mean(lat), LON0=mean(lon)) ## utm
XY = GEOMap::GLOB.XY(lat, lon, PROJ)
plot(range(XY$x), range(XY$y), type='n', asp=1)

## points(XY$x, XY$y)

ternfocgeo(MEKS, PROJ, PLOT=TRUE, icut = 3,
ndivs = 4, add=TRUE, RECT=TRUE)

points(XY$x, XY$y, pch=8, col="purple" )

##### next restrict the boxes to a specific region
plot(range(XY$x), range(XY$y), type='n', asp=1)

```

```

points(XY$x, XY$y)

ternfocgeo(MEKS , PROJ, PLOT=TRUE, icut = 3, ndivs = 5,
  bbox=c(-2000,2000,-2000,2000) , add=TRUE, RECT=TRUE)

## Not run:

#####  this example shows a real application with a map
plot(x=range(IZ$x), y=range(IZ$y), type='n', asp=1, axes=FALSE, ann=FALSE)

image(x=IZ$x, y=IZ$y, z=(UZ), col=blues, add=TRUE)

image(x=IZ$x, y=IZ$y, z=(AZ), col=terrain.colors(100) , add=TRUE)

plotGEOMapXY(haiti.map,
  LIM = c(Lon.range[1],Lat.range[1] ,
  Lon.range[2] ,Lat.range[2]),
  PROJ =PROJ, MAPstyle = 2,
  MAPcol = 'black' , add=TRUE )

H = rectPERIM(JMAT$x0, JMAT$y0)

antipolygon(H$x ,H$y,  col=grey(.85) , corner=1, pct=.4)

sqrTICXY(H , PROJ, side=c(1,2,3,4),  LLgrid=TRUE, col=grey(.7) )

ternfocgeo(OLDCMT, PROJ, PLOT=TRUE, add=TRUE)

## End(Not run)

```

testrightHAND

Test Right Hand of tensor

Description

Test Right Hand of tensor

Usage

```
testrightHAND(U)
```

Arguments

U 3 by 3 matrix

Details

The fuction eigen does not always produce a right-handed eigenvector matrix. The code tests each cross product to see if it creates a right-hand system.

Value

logical vector

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

forcerighthand

Examples

```
Mtens <- c(-0.412, 0.084, 0.328 ,0.398, -1.239, 1.058)

M1 <- matrix(c(Mtens[1], Mtens[4], Mtens[5], Mtens[4],
Mtens[2], Mtens[6], Mtens[5],Mtens[6],
Mtens[3]), ncol=3, nrow=3, byrow=TRUE)

E1 <- eigen(M1)
testrightHAND(E1$vectors)
```

thrust.fault

Thrust Fault Cartoon

Description

Illustrate a thrust fault using animation

Usage

```
thrust.fault(anim = seq(from = 0, to = 1, by = 0.1), KAPPA = 2,
             Light = c(45, 45))
```

Arguments

anim animation vector
KAPPA Phong parameter for lighting
Light lighting point

Details

Program will animate a thrust fault for educational purposes. Animation must be stopped by halting execution.

Value

Graphical Side effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

strikeslip.fault, thrust.fault

Examples

```
thrust.fault(anim=0, KAPPA=4, Light=c(-20, 80))

## Not run:
#### execute a stop command to stop this animation
anim= seq(from=0, to=1, by=.1)
thrust.fault(anim=anim, KAPPA=4, Light=c(-20, 80))

## End(Not run)
```

tk2uv

Tk2uv

Description

Tk plot to u-v coordinate transformation

Usage

```
tk2uv(T, k)
```

Arguments

T	T-value
k	k-value

Details

T and k come from moment tensor analysis.

Value

List: u and v

Author(s)

Keehoon Kim<keehoon@live.unc.edu> Jonathan M. Lees<jonathan.lees@unc.edu>

References

Hudson

See Also

m2tk, hudson.net, hudson.plot

Examples

```
v = c(2,-1,-1)
m = m2tk(v)
tk2uv(m$T, m$k)
```

to.spherical

Convert Cartesian to Spherical

Description

Convert cartesian coordinates to strike and dip

Usage

```
to.spherical(x, y, z)
```

Arguments

x	x-coordinate
y	y-coordinate
z	z-coordinate

Value

LIST

az	angle, degrees
dip	angle, degrees
x	x-coordinate
y	y-coordinate
z	z-coordinate

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

SDRfoc

Examples

```
to.spherical(3, 4, 5)
```

TOCART.DIP

Convert to Cartesian

Description

Convert azimuth and dip to cartesian coordinates

Usage

```
TOCART.DIP(az, dip)
```

Arguments

az	azimuth, degrees
dip	dip, degrees

Value

LIST	
x	x-coordinate
y	y-coordinate
z	z-coordinate
az	azimuth, degrees
dip	dip, degrees

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

to.spherical

Examples

```
TOCART.DIP(134, 32)
```

tocartL	<i>Convert to cartesian coordinate</i>
---------	--

Description

Convert azimuth-dip to cartesian coordinates with list as argument

Usage

tocartL(A)

Arguments

A	az degrees, azimuth dip degrees, dip
---	---

Value

List	
x	x-coordinate
y	y-coordinate
z	z-coordinate

Note

x positive north, y positive east, z positive downward

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

TOCART.DIP, RSEIS::TOCART, tosphereL, to.spherical

Examples

```
A = list(az=23, dip=84)
tocartL(A)
```

TOSPHERE	<i>Convert to Spherical Coordinates</i>
----------	---

Description

Get Azimuth and Dip from Cartesian vector on a sphere.

Usage

TOSPHERE(x, y, z)

Arguments

x	x-coordinate
y	y-coordinate
z	z-coordinate

Value

az	azimuth angle, degrees
dip	dip, degrees

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

TOSPHERE.DIP, tosphereL, to.spherical

Examples

TOSPHERE(3, 4, 5)

TOSPHERE.DIP	<i>convert to spherical coordinates</i>
--------------	---

Description

convert to spherical coordinates

Usage

TOSPHERE.DIP(x, y, z)

Arguments

x	x-coordinate
y	y-coordinate
z	z-coordinate

Details

takes three components and returns azimuth and dip

Value

List

az	azimuth, degrees
dip	Dip, degrees
x	x-coordinate
y	y-coordinate
z	z-coordinate

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

to.spherical

Examples

TOSPHERE.DIP(3, 4, 5)

tosphereL	<i>convert to spherical coordinates</i>
-----------	---

Description

convert to spherical coordinates

Usage

```
tosphereL(A)
```

Arguments

A	list (x,y,z)
---	--------------

Details

takes list of three components and returns azimuth and dip

Value

List	
az	azimuth, degrees
dip	Dip, degrees
x	x-coordinate
y	y-coordinate
z	z-coordinate

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

TOSPHERE

Examples

```
A = list(x=12 ,y=2, z=-3 )
tosphereL(A)
```

TP2XYZ*Trend - Dip to XYZ*

Description

Convert trend and dip to cartesian coordinates.

Usage

TP2XYZ(trend, dip)

Arguments

trend	trend angle, degrees
dip	dip angle, degrees

Details

These are used as functions auxially to rotateFoc.

Value

vector: x, y, z

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

RotTP

Examples

TP2XYZ(34, 40)

TRANmat

Translation Matrix

Description

Create a 4 by 4 translation matrix

Usage

TRANmat(x, y, z)

Arguments

x	x-translation
y	y-translation
z	z-translation

Value

Matrix suitable for translating a 3D body.

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

References

Rogers and Adams, 1990, Mathematical Elements for Computer Graphics, McGraw-Hill, 611p.

See Also

ROTX, ROTZ, ROTY

Examples

zT = TRANmat(5, 4, 2)

*Vmoments**Cartesian Moment Tensors*

Description

Cartesian Moment Tensors from Varvryuk

Usage

`data(Vmoments)`

Format

A list of 9 moment tensors from Vaclav Varvryuk

Source

<http://www.ig.cas.cz/en/research-&-teaching/software-download/>

References

<http://www.ig.cas.cz/en/research-&-teaching/software-download/>

*widdenMoments**Cartesian Moment Tensors*

Description

Cartesian Moment Tensors from Widden Paper in Utah

Usage

`data(widdenMoments)`

Format

A list of 48 moment tensors from Utah

Source

SRL paper

References

Seismological Research Letters

Wnet	<i>Wulff Stereonet</i>
------	------------------------

Description

plot a Wulff Stereonet (Equal-Angle)

Usage

```
Wnet(add = FALSE, col = gray(0.7), border = "black", lwd = 1)
```

Arguments

add	Logical, TRUE=add to existing plot
col	color
border	border color
lwd	line width

Details

Plots equal-angle stereonet as opposed to equal-area.

Value

graphical side effects

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

net, pnet

Examples

```
Wnet()
```

Wpoint*Plot points on Wulff Stereonet*

Description

Adds points to Wulff Equal-Angle Stereonet

Usage

```
Wpoint(az1, dip1, col = 2, pch = 5, lab = "", UP = FALSE)
```

Arguments

az1	azimuth angle, degrees
dip1	dip angle, degrees
col	color
pch	plotting character
lab	label for point
UP	logical, TRUE=Upperhemisphere

Details

Wulff net point is added to existing plot.

Value

graphical side effects

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

Wnet

Examples

```
Wnet()  
Wpoint(23, 34)
```

xsecmanyfoc

Plot Focal Mechs at X-Y position on cross sections

Description

Plot Focal Mechs at X-Y positions on cross sections or other plots that do not have geographic coordinates and projection.

Usage

```
xsecmanyfoc(MEK, theta=NULL, focsiz = 0.5,
foccol = NULL, UP=TRUE, focstyle=1, LEG = FALSE, DOBAR = FALSE)
```

Arguments

MEK	List of Focal Mechanisms, see details
focsiz	focal size, inches
theta	degrees, angle from north for projecting the focal mechs
foccol	focal color, default is to calculate based on rake
UP	logical, UP=TRUE means plot upper hemisphere (DEFAULT=TRUE)
focstyle	integer, 1=beach ball, 2=nipplot
LEG	logical, TRUE= add focal legend for color codes
DOBAR	add strike dip bar at epicenter

Details

Input MEK list contains

```
MEKS = list(lon=0, lat=0, str1=0, dip1=0, rake1=0, dep=0, name="", Elat=0, Elon=0, x=0, y=0)
```

The x, y coordinates of the input list are location where the focals will be plotted. For cross sections x=distance along the section and y would be depth. The focal mechs are added to the current plot.

Value

Graphical Side Effects

Note

If theta is NULL focals are plotted as if they were on a plan view. If theta is provided, however, the mechs are plotted with view from the vertical cross section. The cross section is taken at two points. Theta should be determined by viewing the cross section with the first point on the left and the second on the right. The view angle is through the section measured in degrees from north.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

Lees, J. M., Geotouch: Software for Three and Four Dimensional GIS in the Earth Sciences, Computers & Geosciences, 26, 7, 751-761, 2000.

See Also

justfocXY, plotmanyfoc

Examples

```
##### create and plot the mechs in plan view:
N = 20
lon=runif(20, 235, 243)
  lat=runif(20, 45.4, 49)
  str1=runif(20,50,100)
  dip1=runif(20,10, 80)
  rake1=runif(20,5, 180)

  dep=runif(20,1,15)
  name=seq(from=1, to=length(lon), by=1)
  Elat=NULL
  Elon=NULL
yr = rep(2017, times=N)
jd = runif(N, min=1, max=365)

  MEKS = list(lon=lon, lat=lat, str1=str1, dip1=dip1,
    rake1=rake1, dep=dep, name=name, yr=yr, jd = jd)

  PROJ = GEOMap::setPROJ(type=2, LAT0=mean(lat) , LON0=mean(lon) ) ## utm

  XY = GEOMap::GLOB.XY(lat, lon, PROJ)

  plot(range(XY$x), range(XY$y), type='n', asp=1)

  plotmanyfoc(MEKS, PROJ, focsiz=0.5)

ex = range(XY$x)
why = range(XY$y)

JJ = list(x=ex, y=why)

SWA = GEOMap::eqswath(XY$x, XY$y, MEKS$dep, JJ, width = diff(why) , PROJ = PROJ)

MEKS$x = rep(NA, length(XY$x))
MEKS$y = rep(NA, length(XY$y))

MEKS$x[SWA$flag] = SWA$r
MEKS$y[SWA$flag] = -SWA$depth
```

```
bigR = sqrt( (JJ$x[2]-JJ$x[1])^2 + (JJ$y[2]-JJ$y[1])^2)

plot(c(0,bigR) , c(0, min(-SWA$depth)) , type='n',
     xlab="Distance, KM", ylab="Depth")
points(SWA$r, -SWA$depth)

xsecmanyfoc(MEKS, focsiz=0.5, LEG = TRUE, DOBAR=FALSE)
title("cross section: focals are plotted as if in plan view")

ang1 = atan2( JJ$y[2]-JJ$y[1] , JJ$x[2]-JJ$x[1])

degang = ang1*180/pi

xsecmanyfoc(MEKS, focsiz=0.5, theta=degang, LEG = TRUE, DOBAR=FALSE)
title("cross section: focals are view from the side projection (outer hemisphere)")
```

Z3Darrow	<i>Make a 3D arrow</i>
----------	------------------------

Description

Create the list structure for a 3D arrow.

Usage

```
Z3Darrow(len = 1, basethick = 0.1, headlen = 0.6, headlip = 0.1)
```

Arguments

len	Length in user coordinates
basethick	Thickness of the base
headlen	Length of the head
headlip	Width of the overhang lip

Details

Creates a strucutre suitable for plotting rotated and translated 3D arrows.

Value

List	
aglyph	List of vertices of the faces
anorm	Outward facing normal vectors to faces

Author(s)

Jonathan M. Lees <jonathan.lees@unc.edu>

See Also

PROJ3D, pglyph3D, phong3D

Examples

```
ZA = Z3Darrow(len = 1, basethick = 0.1, headlen = 0.6, headlip = 0.1)
```

Index

* **aplot**

- addmecpoints, [6](#)
- addPT, [7](#)
- addPTarrows, [7](#)
- addsmallcirc, [8](#)
- Beachfoc, [14](#)
- BOXarrows3D, [17](#)
- cirtics, [19](#)
- fancyarrows, [27](#)
- faultplane, [28](#)
- focpoint, [35](#)
- imageSCALE, [48](#)
- justfocXY, [53](#)
- lowplane, [56](#)
- nipXY, [68](#)
- pcirc, [72](#)
- pglyph3D, [72](#)
- phong3D, [74](#)
- PlotPlanes, [81](#)
- PlotPTsmooth, [82](#)
- PlotTernfoc, [84](#)
- PLTcirc, [86](#)
- pnet, [87](#)
- polyfoc, [88](#)
- qpoint, [97](#)
- SDRfoc, [119](#)
- Wpoint, [151](#)

* **array**

- bang, [13](#)
- ROTX, [113](#)
- ROTY, [115](#)
- ROTZ, [117](#)

* **datasets**

- egl, [25](#)
- jimbo, [52](#)
- PKAM, [76](#)
- Vmoments, [149](#)
- widdenMoments, [149](#)

* **dplot**

- ROTX, [113](#)

- ROTY, [115](#)

- ROTZ, [117](#)

* **hplot**

- Beachfoc, [14](#)
- imageP, [47](#)
- imageSH, [49](#)
- imageSV, [50](#)
- justfocXY, [53](#)
- net, [67](#)
- nipXY, [68](#)
- normal.fault, [71](#)
- plotfoc, [77](#)
- PlotPTsmooth, [82](#)
- PlotTernfoc, [84](#)
- Pradfoc, [89](#)
- radiateP, [98](#)
- radiateSH, [99](#)
- radiateSV, [100](#)
- SDRfoc, [119](#)
- SHradfoc, [122](#)
- strikeslip.fault, [130](#)
- SVradfoc, [131](#)
- TEACHFOC, [134](#)
- thrust.fault, [139](#)
- Wnet, [150](#)

* **iplot**

- AXpoint, [12](#)
- EApoin, [24](#)

* **list**

- CONVERTSDR, [20](#)

* **misc**

- AlongGreat, [9](#)
- alpha95, [10](#)
- Bfocvec, [15](#)
- cross.prod, [21](#)
- CROSSL, [22](#)
- doNonDouble, [23](#)
- FixDip, [29](#)

- flipnodal, [30](#)
- foc.color, [31](#)
- foc.icolor, [32](#)
- FOCangles, [33](#)
- focleg, [34](#)
- forcerighthand, [36](#)
- getCMT, [37](#)
- GetRake, [39](#)
- GetRakeSense, [40](#)
- getUWfocs, [42](#)
- HAMMERprojXY, [43](#)
- hudson.net, [44](#)
- hudson.plot, [45](#)
- inverseTAPE, [51](#)
- JMAT, [52](#)
- KAMCORN, [55](#)
- m2tk, [57](#)
- makeblock3D, [58](#)
- makenet, [59](#)
- MapNonDouble, [60](#)
- mc2cart, [63](#)
- mijedr, [64](#)
- MomentDist, [65](#)
- MRake, [66](#)
- nodallines, [70](#)
- plotmanyfoc, [79](#)
- plotMEC, [80](#)
- Preflect, [90](#)
- prepFOCS, [91](#)
- printMEC, [92](#)
- PROJ3D, [92](#)
- PTaxes, [94](#)
- PTXY2, [95](#)
- radP, [101](#)
- radSH, [102](#)
- radSV, [103](#)
- rakelegend, [105](#)
- readCMT, [106](#)
- RectDense, [108](#)
- REFLECT, [109](#)
- rotateFoc, [110](#)
- Rotfocphi, [111](#)
- RotTP, [112](#)
- rotx3, [114](#)
- roty3, [116](#)
- rotz3, [118](#)
- ShadowCLVD, [120](#)
- SourceType, [123](#)
- spherefocgeo, [124](#)
- spline.arrow, [126](#)
- StrikeDip, [128](#)
- TapeBase, [132](#)
- TapePlot, [133](#)
- ternfoc.point, [135](#)
- ternfocgeo, [136](#)
- testrighthAND, [138](#)
- tk2uv, [140](#)
- to.spherical, [141](#)
- TOCART.DIP, [142](#)
- tocartL, [143](#)
- TOSPHERE, [144](#)
- TOSPHERE.DIP, [145](#)
- tosphereL, [146](#)
- TP2XYZ, [147](#)
- TRANmat, [148](#)
- xsecmanyfoc, [152](#)
- Z3Darrow, [154](#)
- * package**
 - RF0C-package, [4](#)
- * spatial**
 - RectDense, [108](#)
- * univar**
 - alpha95, [10](#)
- addmecpoints, [6](#)
- addPT, [7](#)
- addPTarrows, [7](#)
- addsmallcirc, [8](#)
- AlongGreat, [9](#)
- alpha95, [10](#)
- AXpoint, [12](#)
- bang, [13](#)
- Beachfoc, [14](#)
- Bfocvec, [15](#)
- BOXarrows3D, [17](#)
- cirtics, [19](#)
- CONVERTSDR, [20](#)
- cross.prod, [21](#)
- CROSSL, [22](#)
- doNonDouble, [23](#)
- EApoint, [24](#)
- egl, [25](#)
- fancyarrows, [27](#)

faultplane, 28
FixDip, 29
flipnodal, 30
foc.color, 31
foc.icolor, 32
FOCangles, 33
focleg, 34
focpoint, 35
forcerighthand, 36

getCMT, 37
GetRake, 39
GetRakeSense, 40
getUWfocs, 42

HAMMERprojXY, 43
hudson.net, 44
hudson.plot, 45

imageP, 47
imageSCALE, 48
imageSH, 49
imageSV, 50
inverseTAPE, 51

jimbo, 52
JMAT, 52
justfocXY, 53

KAMCORN, 55

lowplane, 56

m2tk, 57
makeblock3D, 58
makenet, 59
MapNonDouble, 60
mc2cart, 63
mijedr, 64
MomentDist, 65
MRake, 66

net, 67
nipXY, 68
nodallines, 70
normal.fault, 71

pcirc, 72
pglyph3D, 72
phong3D, 74

PKAM, 76
plotfoc, 77
plotmanyfoc, 79
plotMEC, 80
PlotPlanes, 81
PlotPTsmooth, 82
PlotTernfoc, 84
PLTcirc, 86
pnet, 87
polyfoc, 88
Pradfoc, 89
Preflect, 90
prepFOCS, 91
printMEC, 92
PROJ3D, 92
PTaxes, 94
PTXY2, 95

qpoint, 97

radiateP, 98
radiateSH, 99
radiateSV, 100
radP, 101
radSH, 102
radSV, 103
rakelegend, 105
readCMT, 106
RectDense, 108
REFLECT, 109
RFOC (RFOC-package), 4
RFOC-package, 4
rotateFoc, 110
Rotfocphi, 111
RotTP, 112
ROTX, 113
rotx3, 114
ROTY, 115
roty3, 116
ROTZ, 117
rotz3, 118

SDRfoc, 119
ShadowCLVD, 120
SHradfoc, 122
SourceType, 123
spherefocgeo, 124
spline.arrow, 126
StrikeDip, 128

strikeslip.fault, [130](#)
SVradfoc, [131](#)

TapeBase, [132](#)
TapePlot, [133](#)
TEACHFOC, [134](#)
ternfoc.point, [135](#)
ternfocgeo, [136](#)
testrightHAND, [138](#)
thrust.fault, [139](#)
tk2uv, [140](#)
to.spherical, [141](#)
TOCART.DIP, [142](#)
tocartL, [143](#)
TOSPHERE, [144](#)
TOSPHERE.DIP, [145](#)
tosphereL, [146](#)
TP2XYZ, [147](#)
TRANmat, [148](#)

Vmoments, [149](#)

widdenMoments, [149](#)
Wnet, [150](#)
Wpoint, [151](#)

xsecmanyfoc, [152](#)

Z3Darrow, [154](#)