Rfringe

November 5, 2003

R topics documented:

fillzm . . . 1
fraunhofer . . . 2
gray256 . . . 3
interferogram . . . 4
listInterferograms . . . 6
makezlist . . . 6
project . . . 7
pupil . . . 8
Rfringe-internal . . . 9
Rfringe . . . 11
rzernike . . . 12
Summarystats . . . 13
synth.interferogram . . . 14
wf.3dplot . . . 15
wf.persp . . . 16
Zernike . . . 16
zlist.qf . . . 18
zmult . . . 18

Index **20**

`fillzm` *Create a matrix of Zernike polynomial values*

Description

Creates a matrix of Zernike polynomials values from vectors `rho` and `theta`

Usage

```
fillzm(rho, theta, phi = 0, zlist = zlist.qf)
```

Arguments

`rho`	normalized radius, 0 <= *rho* <= 1
`theta`	angular coordinate
`phi`	angular coordinate to rotate entire coordinate system
`zlist`	A list with named components `n, m, t`

Details

The arguments `rho` and `theta` must be vectors of the same length.

The optional argument `phi` is an angle specified in *degrees* for rotation of the entire coordinate system. All angles are measured increasing *counterclockwise*.

Value

A matrix of size `length(rho)` rows by `length(zlist$n)` columns.

Note

The main use for this function in Rfringe is creating a matrix of Zernikes to be used as the predictors in the least squares fit to fringe orders.

Author(s)

M.L. Peck ⟨mpeck1@ix.netcom.com⟩

See Also

`Zernike`, `makezlist`, `zlist.qf`.

`fraunhofer` *Star test simulator*

Description

A simple star test simulator, with optional MTF or wavefront plot.

Usage

```
fraunhofer(zcoef, zlist = zlist.qf, obstruct = 0, lambda = 1, defocus = 5,
    pupilsize = 255, npad=1024, gamma=2, psfmag = 2,
    displaymtf = TRUE, displaywf = FALSE, fileout = FALSE)
```

Arguments

`zcoef`	Vector of Zernike coefficients
`zlist`	A list with named components `n, m, t` describing the contents of `zcoef`
`obstruct`	central obstruction *fraction*
`lambda`	Wavelength. Defaults to 1 for a wavefront in waves.
`defocus`	Amount of defocus for side panes, in waves P-V.
`pupilsize`	Size of matrix representing pupil
`npad`	Size of the 0 padded matrix for fft calculations
`gamma`	gamma value for image display
`psfmag`	Magnification factor for in focus PSF display
`displaymtf`	Boolean: display MTF?
`displaywf`	Boolean: display Wavefront?
`fileout`	Boolean: Is the intended output a file?

Details

Computes the intensity distribution from a monochromatic point source in the approximation of Fraunhofer diffraction theory. See *Suiter (1994)* for a full discussion of the application of the theory to practical star testing of small telescopes.

Note

The ratio of npad to pupilsize determines the size of the Airy disk in the "in focus" image. The default choices will produce an Airy disk with a diameter of approximately 10 pixels, and the amount of zero padding is sufficient to prevent significant aliasing for defocus values up to about 20 waves.

If memory limitations cause errors satisfactory images can still be obtained with, say, `npad=512` and `pupilsize=127`.

Author(s)

M.L. Peck ⟨mpeck1@ix.netcom.com⟩

References

Born, M. and Wolf, E. 1999, *Principles of Optics, 7th Edition*, Cambridge University Press.

Suiter, H. R., 1994, *Star Testing Astronomical Telescopes*, Willman-Bell, Inc.

See Also

`Zernike`, `pupil`.

Examples

```
# a random, but probably almost diffraction limited, wavefront

fraunhofer(rnorm(length(zlist.qf$n), mean=0, sd=0.01), displaywf=TRUE)
```

gray256 *8 bit Grayscale*

Description

A vector of gray scale levels

Usage

```
gray256
```

Value

Defined as `gray256 <- grey(seq(0,1,length=256))`

Author(s)

M.L. Peck ⟨mpeck1@ix.netcom.com⟩.

`interferogram` *Interferogram object*

Description

Creates an instance of an interferogram object.

Usage

```
interferogram(filename)
```

Arguments

`filename`	The name of a graphics file in "Portable Anymap" format

Details

Provides an "object oriented" framework for analysis of interferograms. A call of the form `.thisint <- interferogram("myinterferogram.ppm")` creates an "instance" of an interferogram object. All functions for processing the interferometry data are returned with the function call.

The current version only supports graphics files in "Portable Anymap" format, and RGB files are converted to grayscale using the R channel only on the assumption that almost all interferograms will use a He-Ne source.

If `interferogram()` is called from the `Rfringe` GUI **ImageMagick's** (`http://www.imagemagick.org`) `convert` tool is used if necessary to convert from any graphics format supported by **ImageMagick** to pnm format. This means that `convert` must be present and on the search path if this feature is to be utilized. In Windows it also means that the path to the **ImageMagick** installation must be placed earlier in the path list than system files, since there is also a system `convert` command.

Value

A list with the following components:

`ev`	The environment of the `interferogram()` function call
`isInterferogram`	Identifies this object as an interferogram
`image.info`	Data entry function - Basic image information
`analysis.info`	Data entry function - Information required in wavefront analysis
`target.conic.info`	Data entry function - Target conic in single pass tests
`circle.pars`	Image analysis function - Outline aperture edge
`obstruct.pars`	Image analysis function - Outline obstruction (perforation)
`plot.fringes`	Plot function - Plots fringe trace
`autotrace`	Image analysis function - (Semi) automatic fringe tracing
`clearpoints`	Image analysis function - Fringe editing
`addpoints`	Image analysis function - Fringe editing

`clearfringe`	Image analysis function - Fringe editing
`retrace`	Image analysis function - Fringe editing
`addsegment`	Image analysis function - Fringe editing
`insertfringe`	Image analysis function - Fringe editing
`fitzernikes`	Image analysis function - Least squares fit to fringe centers
`plot.si`	Plot function - Synthetic interferogram
`plot.wf`	Plot function - Wavefront map
`plot.contour`	Plot function - Cross sections of wavefront along selected diameters
`plot.wf3d`	Plot function - Interactive 3d plot of wavefront (requires package `rgl`)
`plot.residuals`	Plot function - Some possibly useful diagnostic plots of residuals from least squares fit
`plot.startest`	Plot function - Star test simulation with optional MTF plot
`print.summary`	Output function - Prints basic summary statistics to the console window
`print.details`	Output function - Prints more details of estimated Zernike coefficients
`print.latex`	Output function - A printable detailed report, in pdf format (requires `pdflatex`)

Note

These functions will rarely be accessed directly at the command line if the GUI is used.

Author(s)

M.L. Peck ⟨mpeck1@ix.netcom.com⟩. Thanks to Steven Koehler for valuable programming ideas, especially his approach to object instantiation in R.

See Also

`Rfringe`, `project`, `pixmap`.

Examples

```
   ## Don't run:
     .thisint <- interferogram("myinterferogram.ppm")  # create an interferogram object
     .thisint$circle.pars()      #interactively outline the aperture edge
     .thisint$obstruct.pars()    #outline the obstruction
     .thisint$autotrace()        #trace the fringes
     .thisint$fitzernikes()      #do the least squares fit
     .thisint$plot.wf()          #plot the wavefront
     .thisint$print.summary()    #print some basic summary stats

## End Don't run
```

`listInterferograms` *List Interferogram and Project Objects*

Description

Lists the objects identified as Interferogram or Project objects in the user's workspace.

Usage

```
listInterferograms()
listProjects()
```

Value

Character vectors containing the names of objects identified as interferograms or interferogram projects.

Note

Interferogram objects are identified by having a named attribute `isInterferogram`.

Similarly, project objects have a named attribute `isIntProject`.

Author(s)

M.L. Peck ⟨mpeck1@ix.netcom.com⟩

See Also

`interferogram`, `project`.

`makezlist` *Make a Zernike polynomial list*

Description

Makes a complete list of Zernike polynomial indices `(n, m, t)` suitable for subsequent calls to `fillzm`, `pupil`, etc.

Usage

```
makezlist(minorder = 2, maxorder = 12)
```

Arguments

`minorder`	minimum radial polynomial order (must be even)
`maxorder`	maximum radial polynomial order (must be even)

Value

A list with the following components:

`n`	Radial polynomial order
`m`	Azimuthal order
`t`	character for trig function: one of c(”n”, ”c”, ”s”)

Note

This is a popular one dimensional ordering of the indices `(n,m)` of Zernike polynomials.

The returned list is used in the functions `fillzm`, `pupil`, `zmult`, etc.

Author(s)

M.L. Peck ⟨mpeck1@ix.netcom.com⟩

`project` *Interfometry project*

Description

Creates an instance of an interferometry project.

Usage

```
project(project.id, project.notes=NULL, project.tester=NULL, project.date=NULL)
```

Arguments

`project.id`	Character string - an identifier for the project
`project.notes`	Optional character string - additional project notes
`project.tester`	Optional character string - the tester (or any other useful information)
`project.date`	Optional character string - date of the test (or other information)

Details

Provides an ”object oriented” framework for the analysis of multiple interferograms as a group. A call of the form `.thisproject <- project("myproject")` creates an ”instance” of a project object. All functions for processing the grouped data are returned with the function call.

Value

A list with the following components:

`ev`	The environment of the `project()` function call
`isIntProject`	Identifies this object as an interferometry project
`project.addto`	Utility function - copies data from an interferogram into project object

`project.removefrom`
Utility function - removes interferogram data from project object

`plot.wf` Plot function - Wavefront map

`plot.contour` Plot function - Cross sections of wavefront along selected diameters

`plot.startest` Plot function - Star test simulation with optional MTF plot

`plot.wf3d` Plot function - Interactive 3d plot of wavefront (requires package `rgl`)

`plot.spm` Plot function - Coplots of RMS, P-V, and Strehl from individual interferograms

`print.summary` Output function - Prints basic summary statistics to the console window

`print.details` Output function - Prints more details of estimated Zernike coefficients

`print.latex` Output function - A printable detailed report, in pdf format (requires `pdflatex`)

Note

These functions will rarely be accessed directly at the command line if the GUI is used.

Author(s)

M.L. Peck ⟨mpeck1@ix.netcom.com⟩. Thanks to Steven Koehler for valuable programming ideas, especially his approach to object instantiation in R.

See Also

`Rfringe`, `interferogram`,

Examples

```
  ## Don't run:
    .thisproject <- project("myproject")  # create a project object
    .thisproject$project.addto(.thisint$ev)     #note you add an interferogram
                                                # by passing its environment
    .thisproject$plot.wf()      #Plot averaged wavefront
    .thisproject$plot.spm()     #Possibly interesting comparisons of interferograms
    .thisproject$print.summary()        #Some basic summary statistics

## End Don't run
```

pupil *Create a circular pupil and fill it with a wavefront*

Description

Creates a representation of a wavefront from a vector of Zernike polynomial values.

Usage

```
pupil(size = 255, obstruct = 0, zcoef = NULL, zlist = zlist.qf, phi = 0, piston = 0)
```

Arguments

`size`	size of the returned matrix
`obstruct`	central obstruction *fraction*
`zcoef`	Vector of Zernike coefficients
`zlist`	A list with named components `n, m, t` describing the contents of `zcoef`
`phi`	angular coordinate to rotate entire coordinate system
`piston`	Piston (constant) term to add to the wavefront

Details

The coordinate system is rotated *clockwise* by the angle `phi` specified in *degrees*. This is done to present consistent displays of rotated wavefronts in `Rfringe`.

Value

A `size` by `size` matrix of wavefront values. `NA`'s are used to fill out the matrix outside the circular pupil and inside the obstruction.

Note

Most high level R graphics functions will handle `NA`'s as intended.

This function can take a while if `zcoef` includes high order Zernikes. Decreasing the matrix size will help with speed, but may provide too low resolution for good graphical representations. No attempt is made to "anti-alias" the edges of the pupil.

Author(s)

M.L. Peck ⟨mpeck1@ix.netcom.com⟩

See Also

`Zernike`, `makezlist`, `zlist.qf`, `fillzm`, `pupilrms`, `pupilpv`.

Examples

```
# A fairly typical use of this function:

wf <- pupil(zcoef=rnorm(length(zlist.qf$n)))
image(wf, col=topo.colors(256), asp=1)
contour(wf, add=TRUE)
```

`Rfringe-internal` *Rfringe internal functions*

Description

Internal functions called by the `Rfringe` GUI.

Usage

```
aboutrfringe()
addtoproject()
analysisinfo()
autotrace()
autotrace.options()
batchadd()
changeint()
changeproject()
circlepars()
clearint()
clearproject()
closeRfringe()
closeRfringeandr()
editfringe()
fitzernikes()
helpbox(string)
helprfringe()
imageinfo()
maxorder()
newproject()
obstructpars()
openint()
opensavedint()
opensavedproject()
pdfreport()
pdfreport.project()
plotcontour()
plotcontour.project()
plotfringes()
plotresiduals()
plotspm.project()
plotstartest()
plotstartest.project()
plotwf()
plotwf.project()
printdetails()
printdetails.project()
printsummary()
printsummary.project()
printtoc.project()
prompts(string)
removefromproject()
saveint()
saveproject()
synthint()
targetconicinfo()
tclvar()
wf3d()
wf3d.project()
wf3d.vanilla()
wf3d.vanilla.project()
```

Details

These are called by `Rfringe()` and are not to be called directly by the user. From the command line there are always functions stored within `interferogram`, `project`, or R to perform the actions carried out by these functions.

Author(s)

M.L. Peck ⟨mpeck1@ix.netcom.com⟩

See Also

`Rfringe`, `interferogram`, `project`

`Rfringe` *Rfringe GUI*

Description

GUI front end for R fringe analysis.

Usage

```
Rfringe()
```

Arguments

none

Details

A Tcl/Tk wrapper for the functions provided in `interferogram` and `project`.

Note

The full functionality of `Rfringe` will be described in a separate user guide. This note briefly describes how objects are created and manipulated within the GUI interface.

On startup `Rfringe` creates `NULL` valued objects named `.thisint` and `.thisproject` in the user's global environment. When the user subsequently creates a new interferogram object with the menu selection `File|Load interferogram from image...` `make.names` is run on the value entered in the image id field to create a new (syntactically correct) variable in the user's workspace. This variable is assigned the value of the call to `interferogram()` and `.thisint` is in turn evaluated to the newly created interferogram object. Subsequent commands operate directly on `.thisint`.

Similarly when a new project object is created with the menu selection `File|Project|Create...` a new variable is created from the Project ID entry, it is assigned the value of the call to `project()`, and `.thisproject` is evaluated to the newly created project object.

The objects created with `make.names` are deliberately *not* assigned unique names. This allows the user to replace an object simply by recycling her image or project id entries. The Rfringe GUI will warn if this is attempted.

Author(s)

M.L. Peck ⟨mpeck1@ix.netcom.com⟩. GUI based in part on the package `Rcmdr` by John Fox.

See Also

`tcltk`, `interferogram`, `project`

Examples

```
# start the GUI

Rfringe()
```

---

rzernike *Radial Zernike Polynomial*

Description

Zernike's Radial Polynomials

Usage

```
rzernike(rho, n, m)
odd(n,m)
```

Arguments

`rho`	normalized radius, $0 <= rho <= 1$
`n`	radial polynomial order
`m`	azimuthal order

Details

The arguments `n` and `m` must be relatively even.

Value

The value of the radial Zernike polynomial of order `(n,m)` at normalized radius `rho`. If `rho` is a vector, matrix, or higher order array the returned value is a vector or array of the same dimension.

The utility function `odd(n,m)` returns `FALSE` iff `(n-m)%%2 == 0`.

Note

This routine implements the recurrence relation given in Equation 12 of `http://mathworld.wolfram.com/ZernikePolynomial.html`.

In general you should call the higher level function `Zernike` instead of this.

Author(s)

M.L. Peck ⟨mpeck1@ix.netcom.com⟩

References

http://mathworld.wolfram.com/ZernikePolynomial.html

See Also

Zernike.

`Summarystats` *Wavefront summaries*

Description

Estimate the RMS or P-V wavefront error over the pupil given in the argument.

Usage

```
pupilrms(pupil)
pupilpv(pupil)
strehlratio(rms)
```

Arguments

`pupil`	`pupil` is the matrix created by the call to `pupil`
`rms`	The rms wavefront error

Value

An estimate of the RMS or P-V error of the wavefront, or Strehl ratio.

Note

The function `pupilrms` simply returns the standard deviation of the defined values in `pupil`, which is a crude but usually good enough approximation to the properly defined integral over the aperture.

For a wavefront defined entirely in terms of a vector `zcoef` of Zernike coefficients `rms <- sqrt(crossprod(zcoef))` is faster and more accurate.

`pupilpv` does the obvious. There is no analytical solution in general for P-V.

`strehlratio` computes Mahajan's approximation to the Strehl ratio.

Author(s)

M.L. Peck ⟨mpeck1@ix.netcom.com⟩

See Also

`pupil`.

Examples

```
# A random vector of Zernike coefficients

zcoef <- rnorm(length(zlist.qf$n), mean=0, sd=0.01)
wf <- pupil(zcoef=zcoef)
image(wf, col=topo.colors(256), asp=1)
contour(wf, add=TRUE)
pupilrms(wf)
sqrt(crossprod(zcoef)) #should be the same to about 4 digits
pupilpv(wf)
strehlratio(sqrt(crossprod(zcoef))) #probably around 0.8
```

---

synth.interferogram *Synthetic Interferogram*

Description

Computes and displays a synthetic interferogram for a wavefront constructed from a vector of Zernike coefficients.

Usage

```
synth.interferogram(zcoef, zlist = zlist.qf, phi = 0,
  size = 255, obstruct = 0, iname = "")
```

Arguments

zcoef	Vector of Zernike coefficients, with piston term as the first element
zlist	A list with named components n, m, t describing the contents of zcoef
phi	angular coordinate to rotate entire coordinate system
size	Size of matrix representing pupil
obstruct	central obstruction *fraction*
iname	short string for identification

Details

It's important to note that zcoef is treated differently than in other functions that use the same variable name. The first element *must* be a piston (constant) term, which is stripped off and passed to pupil as the piston argument. The length of zcoef therefore should be one more than the length of zlist$n.

Value

A size by size matrix of intensity levels in the simulated interferogram.

Note

The relationship between wavefront phase and intensity is iwf <- cos(2 * pi * wf + pi), which is the value returned by synth.interferogram. The plot routine in this function plots the image on a 256 level grayscale.

Author(s)

M.L. Peck ⟨mpeck1@ix.netcom.com⟩

See Also

`Zernike`, `pupil`.

Examples

```
zcoef <- c(0, 3, 3, rnorm(length(zlist.qf$n)-2, mean=0, sd = .01))

temp <- synth.interferogram(zcoef, iname="Random wavefront")

# lets see what it looks like in a star test

zcoef <- zcoef[-1]
zcoef[1:2] <- 0
fraunhofer(zcoef, displaywf=TRUE)
```

`wf.3dplot` *Interactive 3D Wavefront plot*

Description

Uses the `rgl` package for OpenGL graphics to produce an interactive 3D wavefront map.

Usage

```
wf.3dplot(wf, zoom.wf = 1)
```

Arguments

`wf`	A wavefront matrix as returned by `pupil`
`zoom.wf`	Zoom factor to stretch wavefront heights

Details

The `rgl` package is available at `http://wsopuppenkiste.wiso.uni-goettingen.de/~dadler/rgl/`. The README file in the source distribution contains installation instructions.

Author(s)

M.L. Peck ⟨mpeck1@ix.netcom.com⟩

References

`http://wsopuppenkiste.wiso.uni-goettingen.de/~dadler/rgl/`

See Also

`pupil`

`wf.persp` *3D Wavefront plot*

Description

An alternative 3D Wavefront plot using the R base package plotting function `persp`.

Usage

```
wf.persp(wf, zoom.wf = 1, theta=0, phi=30, ...)
```

Arguments

`wf`	A wavefront matrix as returned by `pupil`
`zoom.wf`	Zoom factor to stretch wavefront heights
`theta`	Value of theta to pass to `persp`
`phi`	Value of phi to pass to `persp`
`...`	Additional parameters for call to `persp`

Author(s)

M.L. Peck ⟨mpeck1@ix.netcom.com⟩

See Also

`pupil`, `persp`, `wf.3dplot`.

`Zernike` *Zernike Polynomials*

Description

Routines for creating and manipulating Zernike polynomials.

Usage

```
Zernike(rho, theta, n, m, t)
```

Arguments

`rho`	normalized radius, $0 <= rho <= 1$
`theta`	angular coordinate
`n`	radial polynomial order
`m`	azimuthal order
`t`	character for trig function: one of c(”n”, ”c”, ”s”)

Details

The arguments `n` and `m` must be relatively even.

Value

The value of the Zernike polynomial of order `(n, m)` at polar coordinates `(rho, theta)`. The arguments `rho` and `theta` may be vectors, matrices, or higher order arrays, in which case the returned value is a vector or array of the same dimension.

Note

This function returns Zernikes scaled such that they form an orthonormal basis set for the space of functions defined on the unit circle. Note that this is not the most commonly used definition (as given e.g. in *Born and Wolf*). The definition I use is often associated with *Noll (1976)*.

The otherwise unused function `zmult` can be used to convert between normalized and conventionally defined vectors of Zernike coefficients.

Author(s)

M.L. Peck ⟨mpeck1@ix.netcom.com⟩

References

Born, M. and Wolf, E. 1999, *Principles of Optics, 7th Edition*, Cambridge University Press, chapter 9 and appendix VII.

Noll, R.J. 1976, **Zernike polynomials and atmospheric turbulence**, *J. Opt. Soc. Am.*, Vol. 66, No. 3, p. 207.

http://wyant.opt-sci.arizona.edu/zernikes/zernikes.htm

http://mathworld.wolfram.com/ZernikePolynomial.html

See Also

`rzernike`, `makezlist`, `zlist.qf`, `zmult`, `fillzm`, `pupil`, `pupilrms`, `pupilpv`, `strehlratio`.

Examples

```
Zernike(1, 0, 4, 0, "n")        # == sqrt(5)

# A slightly more complex example

rho <- seq(0, 1, length = 101)
theta <- rep(0, 101)

plot(rho, Zernike(rho, theta, 6, 0, "n"), type="l",
  ylim=c(-3.5,3.5), main="Some 6th order Zernike Polynomials")
lines(rho, Zernike(rho, theta, 5, 1, "c"), lty=2)
lines(rho, Zernike(rho, theta, 4, 2, "c"), lty=3)
lines(rho, Zernike(rho, theta, 3, 3, "c"), lty=4)
```

zlist.qf *Fringe set of Zernike Polynomials*

Description

List of the ‘Fringe’ set of Zernike polynomials

Usage

```
zlist.qf
```

Details

The “Fringe” aka “QuickFringe” set of Zernike polynomials.

Value

A list with the following components:

n	Radial polynomial order
m	Azimuthal order
t	character for trig function: one of c(”n”, ”c”, ”s”)

Note

This is the default list of Zernike polynomial indices used by all functions that work with lists of Zernikes. It includes all Zernikes from 2nd through 10th order, plus the 12th order spherical term.

Author(s)

M.L. Peck ⟨mpeck1@ix.netcom.com⟩

zmult *Zernike coefficient multipliers*

Description

Conversion factors between normalized and conventional Zernike polynomials.

Usage

```
zmult(zlist = zlist.qf)
```

Arguments

zlist	A list with named components `n, m, t`

Details

The list indicates the Zernike polynomial orders to use, in the form returned by `makezlist` and `zlist.qf`.

Value

A vector the same length as the components of `zlist`.

Note

This function is not actually used in the current version of Rfringe, but is included for user convenience.

Author(s)

M.L. Peck ⟨mpeck1@ix.netcom.com⟩

See Also

`Zernike`, `makezlist`, `zlist.qf`.

Examples

```
zcoef <- rnorm(length(zlist.qf$n))
zcoef           # a vector of normalized Zernike coefficients
zcoef*zmult()   # Coefficients in conventional representation
sqrt(crossprod(zcoef)) # This is the RMS error of the wavefront
                        # constructed from these Zernikes
```

Index

∗Topic **Basics**
  listInterferograms, 5
∗Topic **Graphics**
  fraunhofer, 2
  gray256, 3
  synth.interferogram, 14
  wf.3dplot, 15
  wf.persp, 16
∗Topic **Mathematics**
  fillzm, 1
  makezlist, 6
  pupil, 8
  rzernike, 12
  Summarystats, 13
  Zernike, 16
  zlist.qf, 18
  zmult, 18
∗Topic **Statistics**
  interferogram, 3
  project, 7
  Rfringe, 10
∗Topic **hplot**
  fraunhofer, 2
  synth.interferogram, 14
  wf.persp, 16
∗Topic **interactive**
  wf.3dplot, 15
∗Topic **internal**
  Rfringe-internal, 9
∗Topic **list**
  listInterferograms, 5
∗Topic **models**
  interferogram, 3
  project, 7
  Rfringe, 10

aboutrfringe *(Rfringe-internal)*, 9
addtoproject *(Rfringe-internal)*, 9
analysisinfo *(Rfringe-internal)*, 9
autotrace *(Rfringe-internal)*, 9

batchadd *(Rfringe-internal)*, 9

changeint *(Rfringe-internal)*, 9
changeproject *(Rfringe-internal)*, 9
circlepars *(Rfringe-internal)*, 9
clearint *(Rfringe-internal)*, 9
clearproject *(Rfringe-internal)*, 9
closeRfringe *(Rfringe-internal)*, 9
closeRfringeandr *(Rfringe-internal)*, 9

editfringe *(Rfringe-internal)*, 9

fillzm, 1, *6*, *9*, *17*
fitzernikes *(Rfringe-internal)*, 9
fraunhofer, 2

gray256, 3
grey256 *(gray256)*, 3

helpbox *(Rfringe-internal)*, 9
helprfringe *(Rfringe-internal)*, 9

imageinfo *(Rfringe-internal)*, 9
interferogram, 3, *6*, *8*, *10*, *11*

listInterferograms, 5
listProjects *(listInterferograms)*, 5

make.names, *11*
makezlist, *2*, 6, *9*, *17–19*
maxorder *(Rfringe-internal)*, 9

newproject *(Rfringe-internal)*, 9

obstructpars *(Rfringe-internal)*, 9
odd *(rzernike)*, 12
openint *(Rfringe-internal)*, 9
opensavedint *(Rfringe-internal)*, 9
opensavedproject *(Rfringe-internal)*, 9

pdfreport *(Rfringe-internal)*, 9
persp, *16*
plotcontour *(Rfringe-internal)*, 9
plotfringes *(Rfringe-internal)*, 9
plotresiduals *(Rfringe-internal)*, 9
plotspm.project *(Rfringe-internal)*, 9
plotstartest *(Rfringe-internal)*, 9

plotwf *(Rfringe-internal)*, 9
printdetails *(Rfringe-internal)*, 9
printsummary *(Rfringe-internal)*, 9
printtoc.project *(Rfringe-internal)*, 9
project, *5, 6*, 7, *10, 11*
prompts *(Rfringe-internal)*, 9
pupil, *3*, *6*, 8, *13–17*
pupilpv, *9*, *17*
pupilpv *(Summarystats)*, 13
pupilrms, *9*, *17*
pupilrms *(Summarystats)*, 13

removefromproject *(Rfringe-internal)*, 9
Rfringe, *4, 5*, *8*, *9*, 10, *10*
Rfringe-internal, 9
rgl, *15*
rzernike, 12, *17*

saveint *(Rfringe-internal)*, 9
saveproject *(Rfringe-internal)*, 9
strehlratio, *17*
strehlratio *(Summarystats)*, 13
Summarystats, 13
synth.interferogram, 14
synthint *(Rfringe-internal)*, 9

targetconicinfo *(Rfringe-internal)*, 9
tclvar *(Rfringe-internal)*, 9

wf.3dplot, 15, *16*
wf.persp, 16
wf3d *(Rfringe-internal)*, 9
wf3d.vanilla.project *(Rfringe-internal)*, 9

Zernike, *2, 3*, *9*, *12*, *14*, 16, *19*
zlist.qf, *2*, *9*, *17*, 18, *18, 19*
zmult, *6*, *17*, 18