Package: FLasher (via r-universe)

November 23, 2026

Title Projection and Forecasting of Fish Populations, Stocks and Fleets

Description Projection of future population and fishery dynamics is carried out for a given set of management targets. A system of equations is solved, using Automatic Differentation (AD), for the levels of effort by fishery (fleet) that will result in the required abundances, catches or fishing mortalities.

X-schema.org-keywords forecast, fisheries, flr, R

Version 0.7.3

Copyright European Union

Depends R(>= 3.5.0), FLCore(> 2.6.5), FLFishery(>= 0.1), ggplotFL

Imports methods, Rcpp (>= 0.12.0), rlang, ggplot2

LinkingTo Rcpp

Byarch false

Additional_repositories https://flr.r-universe.dev

Suggests testthat, knitr, rmarkdown, covr

Collate classes.R generics.R accessors.R methods.R utilities.R fwd.R ffwd.R constructors.R fcb_draw.R RcppExports.R test_helper_functions.R rcpp_plugin.R data.R coerce.R window.R FLasherLib.R plot.R zzz.R

License EUPL

VignetteBuilder knitr

LazyData No

NeedsCompilation Yes

Biarch FALSE

BugReports https://github.com/flr/FLasher/issues

RoxygenNote 7.3.2

Encoding UTF-8

Repository https://ices-tools-dev.r-universe.dev

Date/Publication 2026-05-05 08:35:44 UTC

RemoteUrl https://github.com/flr/FLasher

RemoteRef HEAD

RemoteSha 13761894283b40261f4bd85b638908c11f78db85

Contents

[,fwdControl,ANY,ANY,ANY-method . . . 3
add_target_order_fls . . . 4
basicBlock . . . 4
biolBlock . . . 5
biols . . . 5
calc_F . . . 6
catchBlock . . . 6
coerce . . . 7
compare,FLStock,fwdControl-method . . . 8
draw . . . 9
FCB . . . 10
FCBDrawing . . . 11
fillchar . . . 12
fisheryBlock . . . 12
fwd . . . 13
fwdControl . . . 15
get_FLQuant_element . . . 18
get_FLQuant_elements . . . 19
inlineCxxPlugin . . . 19
iters<- . . . 20
make_test_operatingModel . . . 20
match_posns_names . . . 21
operatingModelRun . . . 22
parsefwdList . . . 22
partialF . . . 23
plot,FLStock,fwdControl-method . . . 23
propagate,fwdControl-method . . . 24
random_FLBiolcpp_generator . . . 25
random_FLCatch_generator . . . 25
random_FLCatches_generator . . . 26
random_FLFisheries_generator . . . 27
random_FLFishery_generator . . . 27
random_FLQuant_generator . . . 28
random_FLQuant_list_generator . . . 29
random_fwdBiols_list_generator . . . 30
random_fwdControl_generator . . . 30
show,fwdControl-method . . . 31
stf . . . 31
summary . . . 33
target . . . 33

targetOrder 34

Index **35**

`[,fwdControl,ANY,ANY,ANY-method`
Set and replacement accessors for fwdControl

Description

We're Pastie to be Grill You

Usage

```
## S4 method for signature 'fwdControl,ANY,ANY,ANY'
x[i, j]

## S4 replacement method for signature 'fwdControl,ANY,ANY,vector'
x[i, j, k, ...] <- value

## S4 replacement method for signature 'fwdControl,ANY,ANY,ANY'
x[i, j, k, ...] <- value

## S4 method for signature 'fwdControl'
x$name

## S4 replacement method for signature 'fwdControl,vector'
x$name <- value

## S4 replacement method for signature 'fwdControl,AsIs'
x$name <- value
```

Arguments

`x`	A fwdControl object
`i`	Row of both target and iters
`j`	Third dimenions of iters
`k`	The replacement.
`...`	Some things.
`value`	Replacement value
`name`	Column name of target or value column of iters.

add_target_order_fls *Add the order column to the control target*

Description

Add the order column to the control target data.frame so that targets are processed in the correct order.

Usage

```
add_target_order_fls(control)
```

Arguments

control	A fwdControl object

Details

It is important that the targets in the control object are processed in the correct order. Targets can happen simultaneously. For example, if there are multiple FLFishery objects in operating model each will need to have a target to solve for at the same time as the others. The targets are processed in a time ordered sequence (year / season). However, within the same year and season it is necessary for the min and max targets to be processed separatley and after the other targets.

Value

A fwdControl object with an order column.

basicBlock *A base class for drawing FCB bits*

Description

Look up and see the flying saucers cruising in the sky I saw one myself it ain't no lie Look down and see the road you're on as if you are on a marathon That's the spirit, victory or die

Slots

height height of box

width width of box

x_centre x coordinate of centre of box

y_centre y coordinate of centre of box

name text to go in the box

name_cex size of text to go in the box

Author(s)

Finlay Scott - EC JRC.

`biolBlock` *A class for drawing a biological stock*

Description

A class for drawing a biological stock

Slots

height height of box
width width of box
x_centre x coordinate of centre of box
y_centre y coordinate of centre of box
name text to go in the box
name_cex size of text to go in the box
neck_length neck length
circle_cex size of the circle bit

Author(s)

Finlay Scott - EC JRC.

`biols` *An FLBiols object used in mixed fishery vignette*

Description

An FLBiols object with two FLBiol objects, one for plaice and one for sole. These are used in the mixed fishery example vignette.

An FLFisheries object with two FLFishery objects, a beam trawl and a gill netter. These are used in the mixed fishery example vignette.

Usage

```
biols

flfs
```

Format

An FLBiols object with two FLBiol objects.

An FLFisheries object with two FLFishery objects.

calc_F *Calculate fishing mortality*

Description

Calculate F in the same way as the internal C++ does

Usage

```
calc_F(catch, biol, effort)
```

Arguments

catch	The FLCatch
biol	The FLBiol
effort	The fishing effort

catchBlock *A class for drawing a catch*

Description

A class for drawing a catch

Slots

height height of box

width width of box

x_centre x coordinate of centre of box

y_centre y coordinate of centre of box

name text to go in the box

name_cex size of text to go in the box

tail_length tail length

Author(s)

Finlay Scott - EC JRC.

coerce *Methods for coercing objects between classes*

Description

A call to *as(from, 'to')* will coerce the object *from*, of a certain class, to one of class *to*, as specified in the method.

Arguments

from	Object to be coerced into one of another class.
to	Name of the output class, *character*.

Details

An object of class *FLQuants* can be coerced into a *fwdControl*, through a call to *as.data.frame*. The name of the element, or elements, in the object specifies the 'quant' in *fwdControl*. The 'quant' in the *FLQuant* object, the name of the first dimension, is ignored unles is one of 'min', 'value' or 'max'. See the examples below on how to pass one or more *FLQuant* objects to *fwd*.

Value

An object of the requested class.

Author(s)

Iago Mosqueira. EC JRC.

See Also

coerce

Examples

```
# Single *catch* target
as(FLQuants(catch=FLQuant(4500, dimnames=list(year=2000))), "fwdControl")
# Single single *f* range
as(FLQuants(f=FLQuant(c(0.1, 0.9),
  dimnames=list(quant=c("min", "max"), year=2000))), 'fwdControl')
# Single *f* target, *value* specified
as(FLQuants(f=FLQuant(0.5, dimnames=list(quant=c("value"), year=2000))),
  'fwdControl')
# *catch* and *ssb* targets
as(FLQuants(catch=FLQuant(4500, dimnames=list(year=2000)),
   ssb_end=FLQuant(12000, dimnames=list(year=2000))), "fwdControl")
# *f* target and *catch* limits
as(FLQuants(f=FLQuant(0.5, dimnames=list(year=2000)),
  catch=FLQuant(c(100, 4000), dimnames=list(quant=c("min", "max"), year=2000))),
  'fwdControl')
```

```
# *f* target and *catch* minimum
as(FLQuants(f=FLQuant(0.5, dimnames=list(year=2000)),
  catch=FLQuant(c(100), dimnames=list(quant=c("min"), year=2000))), 'fwdControl')
# targets with iters
as(FLQuants(fbar=propagate(FLQuant(seq(0.1, 0.5, by=0.1), dim=c(1,5)), 10)),
  "fwdControl")
# targets with different iters
as(FLQuants(fbar=FLQuant(rep(seq(0.1, 0.5, by=0.1), each=10),
  dim=c(1,5,1,1,1,10))), "fwdControl")
```

`compare,FLStock,fwdControl-method`

Compare the result of a `fwd()` *run with the defined targets.*

Description

A comparison between the objects or objects returned by `fwd()` and the targets and limits set in the `fwdControl` object used to run, is returned by this method.

Usage

```
## S4 method for signature 'FLStock,fwdControl'
compare(result, target, simplify = FALSE)

## S4 method for signature 'fwdControl,FLStock'
compare(result, target)

## S4 method for signature 'FLBiol,fwdControl'
compare(result, target, fishery, simplify = FALSE)
```

Arguments

`result`	Object returned by the call to fwd()
`target`	fwdControl object with required targets
`simplify`	Return whole table or logical vector only, logical
`fishery`	FLFishery oir FKFisheries object

Details

A comparison is carried out for each row in a `fwdControl` object, that is, for every target or limit. A `data.frame` is returned with columns 'year', 'quant', 'season' and 'unit' if relevant, and 'achieved'. The last is of class `logical` and will have value `TRUE` if the target or limits have been achieved for every iteration, and `FALSE` otherwise. Values are compared using `all.equal`.

Value

A table of comparisons, one for each target, of class data.frame.

Author(s)

Iago Mosqueira (WMR)

See Also

all.equal.

Examples

```
data(ple4)
control <- fwdControl(
 list(quant="fbar", value=0.5, year=1990),
 list(quant="catch", value=1, year=1991, relYear=1990),
 list(quant="catch", min=10000, year=1993, max=100000))
run <- fwd(ple4, sr=predictModel(model=rec~a*ssb*exp(-b*ssb),
  params=FLPar(a=9.16, b=3.55e-6)), control=control)

# Returns the full comparison table
compare(run, control)
# Returns a logical vector
compare(run, control, simplify=TRUE)
```

---

draw *Generic method for drawing components the FCB matrix*

Description

Used to draw individual components or the whole FCB matrix

Usage

```
draw(object, ...)

## S4 method for signature 'basicBlock'
draw(object)

## S4 method for signature 'biolBlock'
draw(object)

## S4 method for signature 'fisheryBlock'
draw(object)

## S4 method for signature 'catchBlock'
draw(object)

## S4 method for signature 'FCBDrawing'
draw(object)
```

```
## S4 method for signature 'matrix'
draw(object, fisheryNames = NULL, catchNames = NULL, biolNames = NULL)

## S4 method for signature 'fwdControl'
draw(object, fisheryNames = NULL, catchNames = NULL, biolNames = NULL)
```

Arguments

object	The object - linkbasicBlock, linkbiolBlock, linkfisheryBlock, linkcatchBlock, linkmatrix, linkFCBDrawing
...	Other arguments
fisheryNames	A vector of names of the FLFishery blocks
catchNames	A vector of names of the FLCatch blocks
biolNames	A vector of names of the FLBiol blocks

Value

Nothing. Just draws something

Examples

```
FCB <- matrix(c(1,1,1,1,2,2,2,1,2,2,2,3,2,2,4), nrow=5, byrow=TRUE)
draw(FCB)
```

FCB *Create and extract the FCB slot in fwdControl*

Description

The FCB slot in fwdControl specifies the relationships between the catches ('C') taken by each fishery ('F') from different biological units ('B').

Usage

```
FCB(object, ...) <- value

## S4 method for signature 'fwdControl'
FCB(object)

## S4 replacement method for signature 'fwdControl,matrix'
FCB(object) <- value
```

Arguments

object	Input object to construct or extract from.
...	Extract input arguments
value	Input matrix

Details

This slot is of class *matrix* and has thre columns, named 'F', 'C' and 'B', and as many rows as relationships between the FLBiol(s) and FLFishery(ies) objects the fwdControl refers to.

Examples

```
control <- fwdControl()
# Access FCB slot
FCB(control)
# Assign to existing fwdControl
FCB(control) <- FCB(c(f=1, c=1, b=2), c(f=1, c=2, b=2))
```

FCBDrawing *A class for drawing the FCB matrix*

Description

Has to figure out the height, width and positions of the FCB components

Usage

```
FCBDrawing(FCB, ...)

## S4 method for signature 'matrix'
FCBDrawing(FCB, fisheryNames = NULL, catchNames = NULL, biolNames = NULL)
```

Arguments

FCB	The (FCB matrix)
...	Other arguments
fisheryNames	A vector of names of the FLFishery blocks
catchNames	A vector of names of the FLCatch blocks
biolNames	A vector of names of the FLBiol blocks

Value

An FCBDrawing object

Slots

biolBlocks A list of biolBlocks

fisheryBlocks A list of fisheryBlocks

catchBlocks A list of catchBlocks

CBConnectors A list of coordinates connecting catches and biols

FCB The FCB matrix

Author(s)

Finlay Scott - EC JRC.

`fillchar` *Fill up character slots*

Description

Fill the character slots (name etc) with something.

Usage

```
fillchar(object)

## S4 method for signature 'FLFisheries'
fillchar(object)

## S4 method for signature 'FLBiols'
fillchar(object)
```

Arguments

`object`	The object (FLFisheries or FLBiols)

`fisheryBlock` *A class for drawing a fishery*

Description

A class for drawing a fishery

Slots

height height of box

width width of box

x_centre x coordinate of centre of box

y_centre y coordinate of centre of box

name text to go in the box

name_cex size of text to go in the box

neck_length neck length

no_tails number of catches

tail_gap space between catches

tail_length tail length

Author(s)

Finlay Scott - EC JRC.

`fwd` *Method for running fishery projections*

Description

fwd() projects the fishery through time and attempts to hit the specified targets by finding the appropriate fishing effort.

Usage

```
## S4 method for signature 'FLBiols,FLFisheries,fwdControl'
fwd(
  object,
  fishery,
  control,
  effort_max = rep(100, length(fishery)),
  deviances = residuals,
  residuals = lapply(lapply(object, spwn), "[<-", value = 1),
  verbose = FALSE
)

## S4 method for signature 'FLBiols,FLFishery,fwdControl'
fwd(object, fishery, control, ...)

## S4 method for signature 'FLBiol,FLFisheries,fwdControl'
fwd(object, fishery, control, deviances = "missing", ...)

## S4 method for signature 'FLBiol,FLFishery,fwdControl'
fwd(
  object,
  fishery,
  control,
  deviances = residuals,
  residuals = FLQuant(1, dimnames = dimnames(rec(object))),
  ...
)

## S4 method for signature 'FLBiol,FLFishery,missing'
fwd(
  object,
  fishery,
  ...,
  effort_max = 10,
```

```
  deviances = residuals,
  residuals = FLQuant(1, dimnames = dimnames(m(object)))
)

## S4 method for signature 'FLStock,missing,fwdControl'
fwd(
  object,
  control,
  sr,
  maxF = 4,
  deviances = residuals,
  residuals = FLQuant(1, dimnames = dimnames(rec(object))),
  ...
)

## S4 method for signature 'FLStock,ANY,missing'
fwd(
  object,
  fishery = NULL,
  sr,
  maxF = 4,
  deviances = residuals,
  residuals = FLQuant(1, dimnames = dimnames(rec(object))),
  ...
)
```

Arguments

object	An FLStock, an FLBiol or an FLBiols object.
fishery	If object is an FLBiol(s), a FLFishery(ies). Else this argument is ignored.
control	A fwdControl object.
effort_max	Sets a maximum effort limit by fishery as a multiplier over the maximum observed effort.
deviances	An FLQuant of deviances for the stock recruitment relationship (if object is an FLStock).
residuals	Old argument name for deviances, to be deleted
...	Stormbending.
sr	a predictModel, FLSR or list that describes the stock recruitment relationship (if object is an FLStock). Also an FLQuant with actual recruitment values.
maxF	Maximum yearly fishing mortality, when called on an FLStock object.

Details

A projection is run on either an FLStock object (for a single species, single fishery projection), or a pair of FLBiol(s) and FLFishery(ies) objects (for more advanced mixed fisheries projections).

The projection is controlled by the fwdControl object (although it is also possible to control the projection of an FLStock using a different interface). In each timestep of the projection, the fishing

effort of each FLFishery (or the equivalent F multiplier if object is an FLStock) is found so that the targets specified in the fwdControl object are hit.

For more details and examples, see the vignettes in the package and also the tutorial at: http://www.flrproject.org/doc/Forecast

Value

Either an FLStock, or a list of FLFishery and FLBiol objects.

Author(s)

The FLR Team

See Also

FLComp

Examples

```
data(ple4)
# Hindcast with past rec and fbar
hinf <- fwd(ple4, sr=rec(ple4)[, ac(1980:2017)],
  control=fwdControl(year=1980:2017, value=fbar(ple4)[, ac(1980:2017)],
  quant="fbar"))
plot(FLStocks(PLE=ple4, FWD=hinf))
#' Hindcast with past rec and catch
hind <- fwd(ple4, sr=rec(ple4)[, ac(1980:2017)],
  catch=catch(ple4)[, ac(1980:2017)])
```

---

fwdControl *A class for the targets and limits of a fishery and stock projection.*

Description

The desired targets, limits and time steps used in fishery projections can be specified by creating an object of class `fwdControl`.

Constructor for fwdControl objects. Bare bones man pages. Better to look at the vignettes and tutorials.

Usage

```
fwdControl(target, iters, ...)

G(...)

## S4 method for signature 'data.frame,array'
fwdControl(target, iters, ...)

## S4 method for signature 'data.frame,numeric'
```

```
fwdControl(target, iters, ...)

## S4 method for signature 'data.frame,missing'
fwdControl(target, iters, ...)

## S4 method for signature 'list,missing'
fwdControl(target, iters, ...)

## S4 method for signature 'list,list'
fwdControl(target, iters, ...)

## S4 method for signature 'missing,missing'
fwdControl(target, iters, ...)

## S4 method for signature 'FLQuant,missing'
fwdControl(target, quant, ...)
```

Arguments

target	The target. Can be a data.frame, a list or missing.
iters	target The iters. Can be an array, a numeric or missing.
...	Something
quant	name of target to assign 'FLQuant' to, 'character'

Details

...

Slots

`target` The table of quantities and time steps used as target, *data.frame*.

`iters` The values and limits for each target quantity and time step, *array*.

`FCB` The matrix describing which FLCatch of which FLFishery catches which FLBiol. A *matrix* with 3 columns: F, C, and B.

Validity

VALIDITY Neque porro quisquam est qui doloremd ipsum.

Accessors

All slots in the class have accessor and replacement methods defined that allow retrieving and substituting individual slots.

The values passed for replacement need to be of the class of that slot. A numeric vector can also be used when replacing FLQuant slots, and the vector will be used to substitute the values in the slot, but not its other attributes.

Constructor

A construction method exists for this class that can take named arguments for any of its slots. All slots are then created to match the requirements of the class validity. If an unnamed `FLQuant` object is provided, this is used for sizing but not stored in any slot.

Targets

Forecast targets can be based on a variery of stock and fishery quantities. The following are currently available, and a corresponding function, listed at the end, returns the same value when applied on a valid class, for example an `FLStock` object. These targets are defined per time period, either annual or seasonal.

- ssb, ssb_spawn: The Spawning Stock Biomass at spawning time, `ssb`.
- ssb_end: The Spawning Stock Biomass at the end of the time period, `ssb_end`.
- ssb_flash: The Spawning Stock Biomass at the beginning of the following year, `ssb`.
- biomass_spawn: Total Stock Biomass at spawning time, 'tsb'.
- biomass_end: Total Stock Biomass (`tsb`) at the end of the time period.
- f, fbar: Fishing mortality over the time period. `fbar`.
- catch: The total catch over the time period, `catch`.
- landings: The total landings over the time period, 'landings'.
- discards: The total discards over the time period, 'discards'.
- effort: The total effort over the time period, 'effort'.
- revenue: The total revenue over the time period, 'revenue'.

Author(s)

Iago Mosqueira, Finlay Scott - EC JRC.

See Also

data.frame

Examples

```
# CREATE targets on fishing mortality ('f') by year

target <- data.frame(year=2000:2010, value=rlnorm(11), quant='f')

fwc <- fwdControl(target=target)

# INSPECT fwdControl object

show(fwc)

# Construct from data.frame and array
fcn <- fwdControl(data.frame(year=2000:2005, quant='f', value=0.5))
# Construct a fwdControl with some targets having multiple Biols, specified using the G() function
```

```
fwdControl(list(year=2000:2001, value=200, quant="catch", biol=G(1,2)),
  list(year=2002:2003, value=100, quant="catch", biol=c(1,2)))
# Vector of values by year
fwdControl(data.frame(year=2010:2015, quant="f", value=seq(1, 1.3, length=6)))
# Two targets, with ranges for one
fwdControl(data.frame(year=rep(2010:2015, each=2),
  quant=c("f", "catch"),
  min=c(rbind(NA, 20000)), max=c(rbind(NA, 30000)),
  value=c(rbind(seq(1, 1.3, length=6), NA))))
# Single target value
fwdControl(list(year=2010:2014, quant='catch', value=2900))
# One value per target (year)
fwdControl(list(year=2010:2014, quant='catch', value=seq(2900, 3500, length=5)))
# With 40 values (iters) in each target
fwdControl(list(year=2010:2014, quant='catch',
  value=rnorm(200, seq(2900, 3500, length=5))))
# lapply can be used to constructs a list
fwdControl(lapply(2005:2020, function(x) list(quant="catch",
  value=runif(1, 1e5, 1e6), year=x)))
fwdControl(lapply(2005, function(x) list(quant="catch",
  value=runif(1, 1e5, 1e6), year=x)))
# FLQuant, needs 'quant' name
fwdControl(FLQuant(0.2, dimnames=list(year=2000)), quant="fbar")
```

`get_FLQuant_element` *Return 1D element index of an FLQuant*

Description

Given an FLQuant and the indices, returns the 1D element accessor.

Usage

```
get_FLQuant_element(flq, indices)
```

Arguments

`flq`	The FLQuant
`indices`	The indices (integer vector, length 6)

get_FLQuant_elements *Return 1D vector of element indices of an FLQuant*

Description

Given an FLQuant and the indices range, returns the vector of indices

Usage

```
get_FLQuant_elements(flq, indices_min, indices_max)
```

Arguments

flq	The FLQuant
indices_min	The min indices (integer vector, length 6)
indices_max	The max indices (integer vector, length 6)

inlineCxxPlugin *Make the plug in for inline Cxx*

Description

Make the plug in for inline Cxx

Usage

```
inlineCxxPlugin(...)
```

Arguments

...	I have no idea.

iters<- *Access and replace the iters slot of the fwdControl*

Description

Access and replace the iters slot of the fwdControl

Access the iters slot of the fwdControl

Set the iters slot of the fwdControl

Usage

```
iters(object, ...) <- value

## S4 method for signature 'fwdControl'
iters(object)

## S4 replacement method for signature 'fwdControl,array'
iters(object) <- value
```

Arguments

object	The fwdControl.
...	Other things.
value	The iters array to replace the existing one with.

make_test_operatingModel *Make a test operating model from a single FLStock object*

Description

Number of fisheries, catches and biols are taken from the FCB argument.

Usage

```
make_test_operatingModel(
  fls,
  FCB,
  nseasons = 1,
  recruitment_seasons = 1,
  recruitment_age = 1,
  niters = 1000,
  sd = 0.1
)
```

Arguments

fls	The FLStock that the OM is based on
FCB	The FCB matrix
nseasons	The number of seasons
recruitment_seasons	A vector of seasons in which recruitment occurs (1 - 4)
recruitment_age	The age of recruitment to the fishery
niters	The number of iterations.
sd	The standard deviation when applying random lognormal noise to some of the slots.

Value

A list of objects for sending to C++

---

match_posns_names *Change names of biols, catches and fisheries in the control object into integer positions*

Description

Change names of biols, catches and fisheries in the control object into integer positions

Usage

```
match_posns_names(trg, biol_names, fishery_catch_names)
```

Arguments

trg	The target slot of a fwdControl object
biol_names	A vector of names in the FLBiols objects
fishery_catch_names	A named list - elements of list are vector of the catch names of each fishery

Details

Before calling the C++ code it is necessary for the catch, fishery and biol columns (and their Rel) equivalents to be integers. The user can specify by name. This function changes the name to integer position and throws an error if the name does not match.

Value

The updated target slot

operatingModelRun *Call the CPP operatingModel run method*

Description

Call the CPP operatingModel run method

Usage

```
operatingModelRun(
  flfs,
  biols,
  ctrl,
  effort_max,
  effort_mult_initial,
  indep_min,
  indep_max,
  nr_iters = 50L
)
```

Arguments

flfs	FLFisheries.
biols	List of the Biol bits.
ctrl	fwdControl.
effort_max	Maximum yearly rate of change in effort for each fishery.
effort_mult_initial	Initial effort multiplier.
indep_min	Minimum independent solver value.
indep_max	Maximum independent solver value.
nr_iters	Maximum number of iterations for solver.

parsefwdList *Parse the list argument to fwd to make a fwdControl object*

Description

Internal function

Usage

```
parsefwdList(...)
```

Arguments

...	Things

partialF *Calculation of fisheries partial fishing mortalities*

Description

Fishing mortalities at age for one of both stock ('FLBiol') are partitioned along the fisheries ('FLFisheries') exploiting them.

Usage

```
partialF(object, fisheries, ...)

## S4 method for signature 'FLBiols,FLFisheries'
partialF(object, fisheries, biol = seq(length(object)), fcb = "missing")

## S4 method for signature 'FLBiol,FLFisheries'
partialF(object, fisheries, fcb = "missing")
```

Arguments

object	The exploited population or populations, 'FLBiol' or 'FLBiols'.
fisheries	The fisheries exploiting the resource, 'FLFisheries'.
...	Any extra argument.
biol	Position of the biols or biols to do the calculation for.
fcb	FCB matrix of the fishery-catch-biol relationships.

plot,FLStock,fwdControl-method
plot methods that highlight years in fwdControl

Description

plot methods that highlight years in fwdControl

plot method for FLStocks, fwdControl

Usage

```
## S4 method for signature 'FLStock,fwdControl'
plot(x, y, fill = "#E69F00", ...)

## S4 method for signature 'FLQuant,fwdControl'
plot(x, y, fill = "#E69F00", ...)

## S4 method for signature 'FLQuants,fwdControl'
```

```
plot(x, y, fill = "#E69F00", ...)

## S4 method for signature 'FLStocks,fwdControl'
plot(x, y, fill = "#E69F00", ...)
```

Arguments

x	FlStock object to plot
y	fwdControl from which to extract year ranges
fill	Colour to fill projection years background
...	Any other argument to be passed to [ggplotFL::plot]

Generic function

plot(x, y)

Examples

```
data(ple4)
control <- fwdControl(year=2008:2017, quant="f", value=0.3)
# No fwd projection took place, simply passing year range
plot(ple4, control)
plot(ssb(ple4), control)
plot(FLQuants(SSB=ssb(ple4), F=fbar(ple4)), control)
data(ple4)
control <- fwdControl(year=2008:2017, quant="f", value=0.3)
# No fwd projection took place, simply passing year range
plot(FLStocks(PLE4=ple4), control)
```

---

propagate,fwdControl-method

Propagate the fwdControl

Description

Change the nuber of iterations in the iter slot of the fwdControl.

Usage

```
## S4 method for signature 'fwdControl'
propagate(object, iter, fill.iter = TRUE)
```

Arguments

object	A fwdControl object.
iter	The number of iterations.
fill.iter	Fill the new iters with original values (TRUE) or NA (FALSE)

random_FLBiolcpp_generator

Generate randomly sized and filled FLBiolcpp objects

Description

Generate an FLBiolcpp of random size and filled with normally distributed random numbers with a mean of 0. Used for automatic testing, particularly of the fwdBiol class in CPP.

Usage

```
random_FLBiolcpp_generator(sd = 100, ...)
```

Arguments

sd	The standard deviation of the random numbers. Passed to rnorm() Default is 100.
...	Other arguments to pass to random_FLQuant_generator().

Value

An FLBiolcpp

Examples

```
flb <- random_FLBiolcpp_generator()
summary(flb)
```

random_FLCatch_generator

Generate randomly sized and filled FLCatch objects

Description

Generate an FLCatch of random size and filled with normally distributed random numbers with a mean of 0. Used for automatic testing, particularly of the FLCatch class in CPP.

Usage

```
random_FLCatch_generator(sd = 100, ...)
```

Arguments

sd	The standard deviation of the random numbers. Passed to rnorm() Default is 100.
...	Other arguments passed to random_FLQuant_generator().

Value

An FLCatch

Examples

```
flc <- random_FLCatch_generator()
summary(flc)
```

`random_FLCatches_generator` *Generates an FLCatches object - a list of randomly sized and filled FLCatch objects*

Description

Generates a list of identically sized FLCatch objects filled with normally distributed random numbers with a mean of 0. Used for automatic testing, particularly of the FLCatches_base<T> class in CPP.

Usage

```
random_FLCatches_generator(min_catches = 2, max_catches = 5, ...)
```

Arguments

`min_catches`	The minimum number of catches. Default is 2.
`max_catches`	The maximum number of catches. Default is 5.
`...`	Other arguments passed to random_FLQuant_generator().

Value

An FLCatches objects

Examples

```
flcs <- random_FLCatches_generator()
length(flcs)
summary(flcs)
lapply(flcs, summary)
```

random_FLFisheries_generator

Generate a randomly filled and sized FLFisheries object

Description

Generate a randomly sized FLFisheries object filled with normally distributed random numbers with a mean of 0. Used for automatic testing, particularly of the FLFisheries_base<T> class in CPP.

Usage

```
random_FLFisheries_generator(min_fisheries = 2, max_fisheries = 5, ...)
```

Arguments

min_fisheries	The minimum number of FLFisheries in the fisheries list. Default is 2.
max_fisheries	The maximum number of FLFisheries in the fisheries list. Default is 5.
...	Other arguments to pass to random_FLFishery_generator().

Value

An FLFishery object

Examples

```
flf <- random_FLFisheries_generator(fixed_dims = c(NA,10,1,1,1,1))
```

random_FLFishery_generator

Generate a randomly filled and sized FLFishery object

Description

Generate a randomly sized FLFishery object filled with normally distributed random numbers with a mean of 0. Used for automatic testing, particularly of the FLFishery_base<T> class in CPP.

Usage

```
random_FLFishery_generator(min_catches = 2, max_catches = 5, sd = 1, ...)
```

Arguments

min_catches	The minimum number of catches. Default is 2.
max_catches	The maximum number of FLCatches in the catches list. Default is 5.
sd	Standard deviation of the randomly generated FLQuant slots.
...	Other arguments passed to random_FLCatches_generator().

Value

An FLFishery object

Examples

```
flf <- random_FLFishery_generator(fixed_dims = c(NA,10,1,1,1,1))
lapply(flf, summary)
flf <- random_FLFishery_generator(fixed_dims = c(NA,10,1,1,1,1), max_dims = c(100,NA,NA,NA,NA,NA))
```

---

random_FLQuant_generator

Generate randomly sized and filled FLQuant objects

Description

Generate a randomly or fixed sized FLQuant filled with normally distributed random numbers with a mean of 0. Used for automatic testing.

Usage

```
random_FLQuant_generator(
  fixed_dims = rep(NA, 6),
  min_dims = rep(1, 6),
  max_dims = pmax(min_dims, c(5, 10, 5, 4, 4, 5)),
  min_age_name = 1,
  sd = 100
)
```

Arguments

fixed_dims	A vector of length 6 with the fixed length of each of the FLQuant dimensions. If any value is NA it is randomly set using the max_dims argument. Default value is rep(NA,6).
min_dims	A vector of length 6 with minimum size of each of the FLQuant dimensions. Default value is c(1,1,1,1,1,1).
max_dims	A vector of length 6 with maximum size of each of the FLQuant dimensions. Default value is c(5,10,5,4,4,5).
min_age_name	The name of the first age group.
sd	The standard deviation of the random numbers. Passed to rnorm() Default is 100.

Value

An FLQuant

Examples

```
flq <- random_FLQuant_generator()
dim(flq)
summary(flq)
flq <- random_FLQuant_generator(fixed_dims = c(NA,10,1,4,1,NA))
dim(flq)
summary(flq)
```

`random_FLQuant_list_generator` *Generate lists of randomly sized and filled FLQuant objects*

Description

Generate a list of FLQuant objects filled with normally distributed random numbers with a mean of 0. FLQuant objects can be randomly sized, depening on arguments passed to random_FLQuant_generator(). Used for automatic testing, particularly of the FLQuant7_base<T> class in CPP.

Usage

```
random_FLQuant_list_generator(min_elements = 1, max_elements = 10, ...)
```

Arguments

`min_elements`	The minimum number of elements in the list. Default is 1.
`max_elements`	The maximum number of elements in the list. Default is 10.
`...`	Other arguments to pass to random_FLQuant_generator(), e.g. those that fix the size of the objects.

Value

A list of FLQuant objects

Examples

```
flq_list <- random_FLQuant_list_generator()
length(flq_list)
summary(flq_list)
lapply(flq_list, summary)
```

random_fwdBiols_list_generator

Generates a list that can be passed to the CPP fwdBiols constructor

Description

The fwdBiols constructor takes a list (fwdbiols_list). Each element of fwdbiols_list is a list of: FLBiolcpp, SRR residuals and SRR residuals mult. This function generates randomly filled FLBiolcpp objects. Objects may be of different sizes unless appropriate arguments to random_FLBiolcpp_generator() are specified. Used for automatic testing, particularly of the fwdBiols<T> class in CPP.

Usage

```
random_fwdBiols_list_generator(min_biols = 1, max_biols = 5, ...)
```

Arguments

min_biols	The minimum number of fwdBiols in the list. Default is 1.
max_biols	The maximum number of fwdBiols in the list. Default is 5.
...	Other arguments passed to random_FLBiolcpp_generator().

Value

A list object

Examples

```
fwdBiols <- random_fwdBiols_list_generator()
```

random_fwdControl_generator

Random fwdControl object creator

Description

Creates a random fwdControl object for testing purposes

Usage

```
random_fwdControl_generator(
  years = 1:round(runif(1, min = 2, max = 3)),
  nseasons = 2,
  max_nsim_target = 3,
  niters = round(runif(1, min = 5, max = 10))
)
```

Arguments

years	numeric vector of years in the control object. Default value is 1:random interger (max = 10).
nseasons	number of seasons in the projection
max_nsim_target	maximum number of simultaneous targets in each timestep
niters	the number of iterations. Default number is random integer (max = 10).

Value

A fwdControl object

---

show,fwdControl-method *Show method for fwdControl*

Description

More Gills Less Fishcakes

Usage

```
## S4 method for signature 'fwdControl'
show(object)
```

Arguments

object	A fwdControl

---

stf *Prepare object for future projection*

Description

Similar to the old STF method in FLAssess. Extends the object by a number of years and fill in the life history characteristics by taking a mean of the last few years.

Usage

```
stf(object, ...)

## S4 method for signature 'FLStock'
stf(
  object,
  nyears = 3,
  wts.nyears = 3,
  fbar.nyears = wts.nyears,
  f.rescale = FALSE,
  arith.mean = TRUE,
  na.rm = TRUE,
  end = dims(object)$maxyear + nyears,
  disc.nyears = wts.nyears
)

## S4 method for signature 'FLBiol'
stf(
  object,
  nyears = 3,
  wts.nyears = 3,
  arith.mean = TRUE,
  na.rm = TRUE,
  end = dims(object)$maxyear + nyears
)

## S4 method for signature 'FLStocks'
stf(object, ...)
```

Arguments

object	The object (FLStock or FLBiol)
...	Other things.
nyears	Number of years to extend the object
wts.nyears	Number of years to average over to get the future mean weights at age.
fbar.nyears	Number of years to average the F over (only used if object is an FLStock)
f.rescale	Rescale F (TRUE or FALSE - default is FALSE)
arith.mean	If TRUE the arithmetic mean is used. If FALSE the geometric mean is used. Default is TRUE.
na.rm	For the mean function.
end	My beautiful friend
disc.nyears	Number of years to average over to get the future mean proportion of discards at age.

Examples

```
data(ple4)
proj <- stf(ple4, 3)
```

---

summary *summary method for fwdControl*

Description

summary method for fwdControl

Usage

```
## S4 method for signature 'fwdControl'
summary(object)
```

Arguments

object	fwdControl object to show summary of

Generic function

summary(object)

Examples

```
control <- fwdControl(data.frame(year=rep(2010:2015, each=2),
  quant=c("f", "catch"), min=c(rbind(NA, 20000)), max=c(rbind(NA, 30000)),
  value=c(rbind(seq(1, 1.3, length=6), NA))))

summary(control)
```

---

target *Access and replace the target*

Description

Access and replace the target slot of a fwdControl object

Replace the target slot of a fwdControl object

Usage

```
target(object, ...)

target(object, ...) <- value

## S4 method for signature 'fwdControl'
target(object)

## S4 replacement method for signature 'fwdControl,data.frame'
target(object) <- value
```

Arguments

object	The fwdControl object
...	Other things.
value	The target object to replace the existing one with.

---

targetOrder *Get the order of targets in a fwdControl*

Description

Targets must be processed by FLasher in the correct order. Internal function. Ignore.

Usage

```
targetOrder(target, iters)
```

Arguments

target	The target.
iters	The iters.

Index

∗ **classes**
basicBlock, 4
biolBlock, 5
catchBlock, 6
FCBDrawing, 11
fisheryBlock, 12
fwd, 13
fwdControl, 15
∗ **datasets**
biols, 5
∗ **methods**
coerce, 7
compare,FLStock,fwdControl-method, 8
..., *16*
[,fwdControl,ANY,ANY,ANY-method, 3
[<-,fwdControl,ANY,ANY,ANY-method *([,fwdControl,ANY,ANY,ANY-method)*, 3
[<-,fwdControl,ANY,ANY,vector-method *([,fwdControl,ANY,ANY,ANY-method)*, 3
$,fwdControl-method *([,fwdControl,ANY,ANY,ANY-method)*, 3
$<-,fwdControl,AsIs-method *([,fwdControl,ANY,ANY,ANY-method)*, 3
$<-,fwdControl,vector-method *([,fwdControl,ANY,ANY,ANY-method)*, 3

add_target_order_fls, 4
all.equal, *8, 9*

basicBlock, 4
basicBlock-class *(basicBlock)*, 4
biolBlock, 5, *11*
biolBlock-class *(biolBlock)*, 5
biols, 5

calc_F, 6
catchBlock, 6, *11*
catchBlock-class *(catchBlock)*, 6
coerce, 7, *7*
compare,FLBiol,fwdControl-method *(compare,FLStock,fwdControl-method)*, 8
compare,FLStock,fwdControl-method, 8
compare,fwdControl,FLStock-method *(compare,FLStock,fwdControl-method)*, 8

data.frame, *17*
draw, 9
draw,basicBlock-method *(draw)*, 9
draw,biolBlock-method *(draw)*, 9
draw,catchBlock-method *(draw)*, 9
draw,FCBDrawing-method *(draw)*, 9
draw,fisheryBlock-method *(draw)*, 9
draw,fwdControl-method *(draw)*, 9
draw,matrix-method *(draw)*, 9
draw-basicBlock *(draw)*, 9
draw-biolBlock *(draw)*, 9
draw-catchBlock *(draw)*, 9
draw-FCBDrawing *(draw)*, 9
draw-fisheryBlock *(draw)*, 9
draw-fwdControl *(draw)*, 9
draw-matrix *(draw)*, 9
draw-method *(draw)*, 9

FCB, 10
FCB,fwdControl-method *(FCB)*, 10
FCB<-,fwdControl,matrix-method *(FCB)*, 10
FCBDrawing, 11, *11*
FCBDrawing,matrix-method *(FCBDrawing)*, 11
FCBDrawing-class *(FCBDrawing)*, 11
FCBDrawing-matrix *(FCBDrawing)*, 11
FCBDrawing-method *(FCBDrawing)*, 11
fillchar, 12

fillchar,FLBiols-method *(fillchar)*, 12
fillchar,FLFisheries-method *(fillchar)*, 12
fisheryBlock, *11*, 12
fisheryBlock-class *(fisheryBlock)*, 12
FLComp, *15*
flfs *(biols)*, 5
fwd, 13
fwd,FLBiol,FLFisheries,fwdControl-method *(fwd)*, 13
fwd,FLBiol,FLFishery,fwdControl-method *(fwd)*, 13
fwd,FLBiol,FLFishery,missing-method *(fwd)*, 13
fwd,FLBiols,FLFisheries,fwdControl-method *(fwd)*, 13
fwd,FLBiols,FLFisheries,missing-method *(fwd)*, 13
fwd,FLBiols,FLFishery,fwdControl-method *(fwd)*, 13
fwd,FLStock,ANY,missing-method *(fwd)*, 13
fwd,FLStock,missing,fwdControl-method *(fwd)*, 13
fwd,FLStock,missing,missing-method *(fwd)*, 13
fwdControl, 15
fwdControl,data.frame,array-method *(fwdControl)*, 15
fwdControl,data.frame,missing-method *(fwdControl)*, 15
fwdControl,data.frame,numeric-method *(fwdControl)*, 15
fwdControl,FLQuant,missing-method *(fwdControl)*, 15
fwdControl,list,list-method *(fwdControl)*, 15
fwdControl,list,missing-method *(fwdControl)*, 15
fwdControl,missing,missing-method *(fwdControl)*, 15
fwdControl-class *(fwdControl)*, 15
fwdControl-methods *(fwdControl)*, 15

G *(fwdControl)*, 15
get_FLQuant_element, 18
get_FLQuant_elements, 19

inlineCxxPlugin, 19
iters,fwdControl-method *(iters<-)*, 20
iters<-, 20
iters<-,fwdControl,array-method *(iters<-)*, 20

make_test_operatingModel, 20
match_posns_names, 21
matrix, *11*

operatingModelRun, 22

parsefwdList, 22
partialF, 23
partialF,FLBiol,FLFisheries-method *(partialF)*, 23
partialF,FLBiols,FLFisheries-method *(partialF)*, 23
plot,FLQuant,fwdControl-method *(plot,FLStock,fwdControl-method)*, 23
plot,FLQuants,fwdControl-method *(plot,FLStock,fwdControl-method)*, 23
plot,FLStock,fwdControl-method, 23
plot,FLStocks,fwdControl-method *(plot,FLStock,fwdControl-method)*, 23
propagate,fwdControl-method, 24

random_FLBiolcpp_generator, 25
random_FLCatch_generator, 25
random_FLCatches_generator, 26
random_FLFisheries_generator, 27
random_FLFishery_generator, 27
random_FLQuant_generator, 28
random_FLQuant_list_generator, 29
random_fwdBiols_list_generator, 30
random_fwdControl_generator, 30

show,fwdControl-method, 31
stf, 31
stf,FLBiol-method *(stf)*, 31
stf,FLStock-method *(stf)*, 31
stf,FLStocks-method *(stf)*, 31
summary, 33
summary,fwdControl-method *(summary)*, 33

target, 33
target,fwdControl-method *(target)*, 33
target<- *(target)*, 33

target<-,fwdControl,data.frame-method
(target), 33
targetOrder, 34