Package ‘whalestrike’

March 6, 2026

Title Simulate Whale Ship Strikes

Version 0.6.2

Maintainer Dan Kelley <dan.kelley@dal.ca>

Description

Provides tools for simulating the biophysical effects of vessel-strikes on whales. The aim is to support the evaluation of marine policies limiting ship speeds through regions in which whales reside. This is important because ship strikes are a major source of lethality for several whale species, including the critically endangered North Atlantic right whale. In this analysis, whales are modelled with a four-layer system comprising skin, blubber, sub-layer (muscle or organ) and bone. Reasonable values for the material properties of these layers, along with other factors such as whale surface area and mass, are provided for a variety of whale species. Similarly, key values are provided for several ship types. The collision is modelled according to Newtonian dynamics, with stresses and strains within the whale layers being simulated over time. The simulation results are analyzed in the context of whale-strike data, to develop a Lethality Index for the whale in the modelled collision. For the underlying science, see Kelley and other ``Assessing the Lethality of Ship Strikes on Whales Using Simple Biophysical Models." (2021) <doi:10.1111/mms.12745>. For more on the R code, see Kelley ```whalestrike`: An R package for simulating ship strikes on whales" (2024) <doi:10.21105/joss.06473>.

Depends bslib, deSolve, R (>= 3.5.0), shiny

Suggests knitr, rmarkdown, testthat

License GPL (>= 3)

Encoding UTF-8

BugReports https://github.com/dankelley/whalestrike/issues

URL https://dankelley.github.io/whalestrike/

LazyData true

RoxygenNote 7.3.2

BuildVignettes true

VignetteBuilder knitr

NeedsCompilation no

Author Dan Kelley [aut, cre] (ORCID: <https://orcid.org/0000-0001-7808-5911>),
James Vlasic [rtm] (ORCID: <https://orcid.org/0000-0002-3846-4391>),
Sean Brilliant [rtm] (ORCID: <https://orcid.org/0000-0001-5494-3475>),
Alexandra Mayette [rtm] (ORCID:
<https://orcid.org/0000-0002-9766-9565>)

Repository CRAN

Date/Publication 2026-03-06 18:20:02 UTC

Contents

app *GUI application for whale simulation*

Description

Graphical-user-interface tool for exploring whale-strike simulations.

Usage

```
app(debug = FALSE)
```

Arguments

debug	logical value indicating whether to print output to the R console as the computation is done.

Details

Sliders, buttons, and choosers are grouped into panes that appear on the left of the view. When `app()` first opens, all of these panes are closed. To get acquainted with the app, try adjusting the controllers that *are* visible on the initial view. Then, open the "ship" pane and increase the ship mass. Do you find that the results make qualitative sense? Continue this process, exploring all the panes. A half-hour of such exploration should be enough to build enough confidence to start investigating practical applications. To learn more about how the simulations are carried out, and to read more about the underlying goals of this tool, please consult Kelley et al. (2021) and Kelley (2024). Extensive details on the calculations are provided in the help pages for the various functions of the whalestrike package, of which that for `whalestrike()` is a good starting point.

More information on `app()` in video form on youtube.

Note that an older version of a similar GUI application is still available as `app_2025()`, but it is not maintained and is slated for removal in the early months of 2026.

Value

A Shiny app object, created by a call to `shinyApp()`.

Author(s)

Dan Kelley

References

Kelley, Dan E., James P. Vlasic, and Sean W. Brillant. "Assessing the Lethality of Ship Strikes on Whales Using Simple Biophysical Models." Marine Mammal Science 37, no. 1 (2021): 251–67. doi:10.1111/mms.12745.

Kelley, Dan E.""Whalestrike: An R Package for Simulating Ship Strikes on Whales." Journal of Open Source Software 9, no. 97 (2024): 6473. doi:10.21105/joss.06473.

Mayette, Alexandra. "Whale Layer Thickness." December 15, 2025. (Personal communication of a 5-page document.)

Mayette, Alexandra, and Sean W. Brillant. "A Regression-Based Method to Estimate Vessel Mass for Use in Whale-Ship Strike Risk Models." PloS One 21, no. 1 (2026): e0339760. doi:10.1371/journal.pone.0339760.

See Also

Other interactive apps: `app_2025()`

`app_2025` *GUI app for interactive whale-strike simulations (OLD VERSION, no longer maintained)*

Description

The `app_2025()` function starts a GUI application that makes it easy to run simple simulations and see the results in graphical form. Sliders and buttons permit a fair degree of customization. The application has some build-in documentation, which supplements what can be found in the 'Details' section of the present documentation.

Usage

```
app_2025(mode = "simple", options = list(height = 500))
```

Arguments

`mode`	character value specifying the style to use. Only the value `"simple"` is permitted at present. This yields a 3-panel plot, constructed by `plot.strike`, called with default arguments.
`options`	list containing options that are provided to `shinyApp`, which creates the GUI app.

Details

When `app_2025()` is run, a window will appear within a few moments. At the top of that is a textual introduction to the system, with a button to hide that information. Below is a user-interaction area, with buttons and sliders that control the simulation and the plotted output. Below that is a plotting area, the contents of which depend on the configuration of the simulation as well as the user's selection of items to display.

The default setup, which is shown before the user alters any of the sliders, etc., is a simulation of a small fishing boat, of mass 45 tonnes, moving at speed 10 knots towards a whale of length 13.7m. (The whale length is used to compute its mass, using a formula that is described by the output of typing `help("whaleMassFromLength","whalestrike")` in an R console).

Sliders are provided for setting certain key properties of the ship and the whale, with italic labels for those properties that are deemed most likely to be adjusted during simulations. The details of these and the other parameters are revealed by typing `help("parameters","whalestrike")` and `help("strike","whalestrike")` in an R console.

To the right of the sliders is a column of checkboxes that control the plotted output. At startup, three of these boxes are ticked, yielding a display with three panels showing the time history of the simulation (`help("plot.strike","whalestrike")` provides details of the plots):

- The left-hand plot panel shows whale and boat location, the former with an indication of the interfaces between skin, blubber, sublayer, and bone.
- The middle panel shows the same information as the left one, but with a whale-centred coordinate system, and with labels for the components. This makes it easier to see the degree to which the layers are compressed during the impact.
- The right panel is an indication of the estimated threat to the four layers of the whale, with curves that are filled with grey for time intervals when the impact stress (force/area) is less than the strength of the material in the layer, and black for times when that threshold is exceeded.

Much can be learned by adjusting the sliders and examining the plotted output. As an exercise, try setting to a particular ship mass of interest, and then to slide the ship speed to higher and lower values, whilst monitoring the "threat" panel for black regions. This will reveal a critical speed for conditions that threaten the whale. Next, try altering the sublayer thickness, which is a surrogate for location along the whale body, because e.g. the sublayer is thinner near the mandible.

Advanced users are likely to want to alter the values of impact width and height. The default setting are intended to mimic a small fishing boat, such as a Cape Islander. Try lowering the width, to simulate a strike by a daggerboard or keel of a sailing boat.

Note that the pulldown menu for setting the whale species affects *only* whale mass. It does not affect the thicknesses of blubber or sublayer, the values of which have been set up to represent a midsection strike on a North Atlantic Right Whale.

Value

A Shiny app object, created by a call to `shinyApp()`.

Author(s)

Dan Kelley

See Also

Other interactive apps: `app()`

`derivative` *Calculate derivative using first difference*

Description

The derivative is estimated as the ratio of the first-difference of `var` divided by the first-difference of `time`. To make the results have the same length as `time`, the final result is appended at the end.

Usage

```
derivative(var, t)
```

Arguments

`var`	variable.
`t`	time in seconds.

Value

Derivative estimated by using `diff()` on both `var` and `time`.

Author(s)

Dan Kelley

`dynamics` *Dynamical law*

Description

This function handles Newton's second law, which is the dynamical law that relates the accelerations of whale and ship to the forces upon each. It is used by `strike()`, as the latter integrates the acceleration equations to step forward in time through the simulation of a whale-strike event. Thus, `dynamics()` is a core function of this package. The code is very simple, because the forces are determined by other functions, as described in the "Details" section.

Usage

```
dynamics(t, y, parms)
```

Arguments

`t`	time (s).
`y`	model state, a vector containing ship position `xs` (m), ship speed `vs` (m/s), whale position `xw` (m), and whale speed `vw` (m/s).
`parms`	A named list holding model parameters, created by `parameters()`.

Details

Given a present state (defined by the positions and velocities of ship and whale) at the present time, apply Newton's second law to find the time derivatives of that state. Forces are determined with `whaleCompressionForce()`, `whaleSkinForce()`, `shipWaterForce()`, `whaleWaterForce()`, while engine force (assumed constant over the course of a collision) is computed from initial `shipWaterForce()`. Whale and ship masses are set by `parameters()`, which also sets up areas, drag coefficients, etc.

Value

An List contain items named `dxsdt` (time derivative of ship location), `dvsdt` (time derivative of ship speed), `dxwdt` (time derivative of whale location) and `dvwdt` (time derivative of whale speed). These are computed by solving the dynamical system using Newton's second law, based on the known masses of ship and whale, and the forces involved in the collision.

Author(s)

Dan Kelley

References

See `whalestrike()` for a list of references.

`fillplot` *Draw polygon between two xy curves*

Description

This adds to an existing plot by filling the area between the lower=lower(x) and upper=upper(x) curves. In most cases, as shown in "Examples", it is helpful to use `xaxs="i"` in the preceding plot call, so that the polygon reaches to the edge of the plot area.

Usage

```
fillplot(x, lower, upper, ...)
```

Arguments

`x`	Coordinate along horizontal axis
`lower`	Coordinates of the lower curve, of same length as `x`, or a single value that gets repeated to the length of `x`.
`upper`	Coordinates of the upper curve, or a single value that gets repeated to the length of `x`.
`...`	passed to `polygon()`. In most cases, this will contain `col`, the fill colour, and possibly `border`, the border colour, although cross-hatching with `density` and `angle` is also a good choice.

Value

None. This function is called to add to a plot.

Author(s)

Dan Kelley

Examples

```
# 1. CO2 record
plot(co2, xaxs = "i", yaxs = "i")
fillplot(time(co2), min(co2), co2, col = "pink")

# 2. stack (summed y) plot
x <- seq(0, 1, 0.01)
lower <- x
upper <- 0.5 * (1 + sin(2 * pi * x / 0.2))
plot(range(x), range(lower, lower + upper),
    type = "n",
    xlab = "x", ylab = "y1, y1+y2",
    xaxs = "i", yaxs = "i"
)
fillplot(x, min(lower), lower, col = "darkgray")
fillplot(x, lower, lower + upper, col = "lightgray")
```

---

knot2mps *Convert a speed in knots to a speed in m/s*

Description

See also `mps2knot()`, which is the inverse of this function.

Usage

```
knot2mps(knot)
```

Arguments

`knot`	Speed in knots.

Value

Speed in m/s.

Author(s)

Dan Kelley

See Also

Other functions dealing with units: `mps2knot()`

Examples

```
library(whalestrike)
knots <- seq(0, 20)
plot(knots, knot2mps(knots), xlab = "Speed [knots]", ylab = "Speed [m/s]", type = "l")
```

`lethalityIndexFromStress` *Compute lethality index, based on compression stress*

Description

The model used for this is the logistic model, fitting observed injury/lethality statistics to the base-10 logarithm of the maximum compression stress during a simulated impact event.

Usage

```
lethalityIndexFromStress(stress)
```

Arguments

`stress`	numerical value or vector, giving whale compression stress in Pascals.

Value

threat of injury (in range 0 to 1)

Author(s)

Dan Kelley

See Also

Other functions dealing with Whale Lethality index: `maximumLethalityIndex()`, `stressFromLethalityIndex()`

Examples

```
lethalityIndexFromStress(parameters()$logistic$tau50) # approx. 0.5
```

maximumLethalityIndex *Find maximum Lethality Index during a strike*

Description

This works by finding the maximum Lethality Index encountered during a simulation created by calling strike(), and so it is important to use a detailed setting for the output times. In the example, the results are reported every 0.7/200 seconds (i.e. 3.5 milliseconds), which is likely sufficient (see the example, where a plot is used for this assessment).

Usage

```
maximumLethalityIndex(strike)
```

Arguments

strike	the value returned by a call to strike.

Value

The maximum value of the Lethality Index that is involved in the simulation of the ship-whale collision event. This is a unitless number; see Kelley et al. (2021).

Author(s)

Dan Kelley, wrapping code provided by Alexandra Mayette

References

Kelley, Dan E., James P. Vlasic, and Sean W. Brillant. "Assessing the Lethality of Ship Strikes on Whales Using Simple Biophysical Models." Marine Mammal Science 37, no. 1 (January 2021): 251–67.

See Also

Other functions dealing with Whale Lethality index: lethalityIndexFromStress(), stressFromLethalityIndex()

Examples

```
library(whalestrike)
t <- seq(0, 0.7, length.out = 200)
state <- list(xs = -2, vs = knot2mps(10), xw = 0, vw = 0)
parms <- parameters()
s <- strike(t, state, parms)
# Compute the desired value and (for context) show it on a plot
maximumLethalityIndex(s)
# For context, this is how this can be done "by hand"
max(lethalityIndexFromStress(s[["WCF"]][["stress"]]))
# Show the maximum on a plot (see also the plot title)
```

```
plot(s, which = "lethality index")
abline(h=maximumLethalityIndex(s), col=2)
```

---

mps2knot *Convert a speed in m/s to a speed in knots*

Description

This is done by dividing by the factor 1.852e3/3600, See also knot2mps(), which is the inverse of this function.

Usage

```
mps2knot(mps)
```

Arguments

mps	Speed in metres per second.

Value

Speed in knots.

Author(s)

Dan Kelley

See Also

Other functions dealing with units: knot2mps()

Examples

```
library(whalestrike)
mps <- seq(0, 10)
plot(mps, mps2knot(mps), xlab = "Speed [m/s]", ylab = "Speed [knots]", type = "l")
```

parameters *Set parameters for a whale strike simulation*

Description

Assembles control parameters into a list suitable for passing to `strike()` and the functions that it calls. If `file` is provided, then all the other arguments are read from that source. Note that `updateParameters()` may be used to modify the results of `parameters`, e.g. for use in sensitivity tests.

Usage

```
parameters(
  ms = 45000,
  Ss = NULL,
  Ly = 1.15,
  Lz = 1.15,
  species = "N. Atl. Right Whale",
  lw = 13.7,
  mw = NULL,
  Sw = NULL,
  l = NULL,
  a = NULL,
  b = NULL,
  s = NULL,
  theta = 55,
  Cs = 0.01,
  Cw = 0.0025,
 logistic = list(logStressCenter = 5.38, logStressWidth = 0.349, tau25 = 1e+05, tau50 =
    241000, tau75 = 581000),
  file = NULL
)
```

Arguments

ms	Ship mass (kg).
Ss	Ship wetted area (m^2). This, together with `Cs`, is used by `shipWaterForce()` to estimate ship drag force. If `Ss` is not given, then an estimate is made by calling `shipAreaFromMass()` with the provided value of `ms`.
Ly	Ship impact horizontal extent (m); defaults to 1.15m if not specified, based on an analysis of the shape of the bow of typical coastal fishing boats of the Cape Islander variety.
Lz	Ship impact vertical extent (m); defaults to 1.15m if not specified, based on the same analysis as for Ly.

species
: a string indicating the whale species. For the permitted values, see `whaleMassFromLength()`. (The `species` value can also set the `lw` and `l` values, as noted in their portions of this documentation.)

lw
: either (1) whale length in metres or (2) the string `"from_species"`. If the latter, then the length is determined from `whaleMeasurements()`. In either case, the length is used by `whaleAreaFromLength()` to calculate area, which is needed for the water drag calculation done by `whaleWaterForce()`.

mw
: either (1) the whale mass in kg or (2) NULL. In the latter case, the mass is calculated from whale length, using `whaleMassFromLength()` with `type="wetted"`.

Sw
: either (1) the whale surface area in m^2 or (2) NULL. If the latter case, the area is calculated from whale length using `whaleAreaFromLength()`.

l
: either (1) a numerical vector of length 4 that indicates the thicknesses in metres of skin, blubber, sublayer and bone; (2) NULL to set these four values to 0.025, 0.16, 1.12, and 0.1; or (3) the string `"from_species"`, in which case these four values are determined by calling `whaleMeasurements()`. The default skin thickness of 0.025 m represents the 0.9-1.0 inch value stated in Section 2.2.3 of Raymond (2007). The blubber default of 0.16 m is a rounded average of the values inferred by whale necropsy, reported in Appendix 2 of Daoust et al., 2018. The sublayer default of 1.12 m may be reasonable at some spots on the whale body. The bone default of 0.1 m may be reasonable at some spots on the whale body. The sum of these default values, 1.40 m, is a whale radius that is consistent with a half-circumference of 4.4 m, reported in Table 2.2 of Raymond (2007). Note, however, that these values are not identical to those found in `whaleMeasurements`.

a, b
: Numerical vectors of length 4, giving values to use in the stress-strain law `stress=a*(exp(b*strain)-1)`, where `a` is in Pa and `b` is unitless. By construction, `a*b` is the local modulus at low strain (i.e. at low `b*strain` values), and that `b` is the efolding scale for nonlinear increase in stress with strain. This exponential relationship has been mapped out for whale blubber, using a curve fit to Figure 2.13 of Raymond (2007), and these values are used for the second layer (blubber); see the documentation for the raymond2007 dataset, to see for how that fit was done. If not provided, `a` defaults to `c(17.8e6/0.1, 1.58e5, 1.58e5, 8.54e8/0.1)` and `b` defaults to `c(0.1, 2.54, 2.54, 0.1)`. The skin defaults are set up to give a linear shape (since `b` is small) with the `a*b` product being 17.8e6 Pa, which is the adult-seal value given in Table 3 of Grear et al. (2017). The blubber defaults are from a regression of the stress-strain relationship shown in Figure 2.13 of Raymond (2007). The sublayer defaults are set to match those of blubber, lacking any other information. The bone default for `b` is small, to set up a linear function, and `a*b` is set to equal 8.54e8 Pa, given in Table 2.3 of Raymond (2007) and Table 4.5 of Campbell-Malone (2007).

s
: Numerical vector of length 4, giving the ultimate strengths (Pa) of skin, blubber, sublayer, and bone, respectively. If not provided, the value is set to `1e6 * c(19.600,0.255,0.255,22.900)` with reasoning as follows. The skin default of 19.6 MPa is a rounded value from Table 3 of Grear et al. (2018) for adult seal skin strength at an orientation of 0 degrees. The blubber and sublayer values were chosen as the central point of a logistic fit of whale collision damage to maximal stress during a default impact simulation. (For comparison, a

strength of 0.437 MPa may be inferred by multiplying Raymond's (2007) Figure 2.13 elastic modulus of 0.636 MPa by the ratio 0.97/1.41 determined for adult seal strength/modulus, as reported in Table 3 of Grear et al. (2018).) The bone default o 22.9 MPa is from Table 2.3 of Raymond (2007) and Table 4.5 of Campbell-Malone (2007).

`theta` Whale skin deformation angle (deg); defaults to 55 degrees, if not supplied, because that angle produces a good match to Raymond's (2007) Figure 6.1 for the total force as a function of vessel speed, for large vessels. Note that the match works almost as well in the range 50 deg to 70 deg.

`Cs` Drag coefficient for ship (dimensionless), used by `shipWaterForce()` to estimate ship drag force. Defaults to 1e-2, which is 4 times the frictional coefficient of 2.5e-3 inferred from Figure 4 of Manen and van Oossanen (1988), assuming a Reynolds number of 5e7, computed from speed 5m/s, lengthscale 10m and viscosity 1e-6 m^2/s. The factor of 4 is under the assumption that frictional drag is about a quarter of total drag. The drag force is computed with `shipWaterForce()`.

`Cw` Drag coefficient for whale (dimensionless), used by `whaleWaterForce()` to estimate whale drag force. Defaults to 2.5e-3, for Reynolds number 2e7, computed from speed 2 m/s, lengthscale 5m which is chosen to be between radius and length, and viscosity 1e-6 m^2/s. The drag force is computed with `whaleWaterForce()`.

`logistic` a list containing `logStressCenter` and `logStressWidth`, which define an empirical logistic fit of an index of whale injury in observed strikes (ranging from 0 for no injury to 1 for fatal injury), as a function of the base-10 logarithm of compressive stress, as well as `tau25`, `tau50` and `tau75`, which are the stresses in that fit that yield index values of 0.25, 0.50 and 0.75, respectively; these values set colour boundaries in `plot.strike()` plots that have `which="threat"`.

`file` Optional name a comma-separated file that holds all of the previous values, except `Cs` and `Cw`. If provided, then other parameters except `Cs` and `Cw` are ignored, because values are sought from the file. The purpose of this is in shiny apps that want to save a simulation framework. The file should be saved `write.csv()` with `row.names=FALSE`.

Value

A named list holding the parameters, with defaults and alternatives reconciled according to the system described above, along with some items used internally, including `lsum`, which is the sum of the values in `l`, and `stressFromStrain()`, a function created by `stressFromStrainFunction()` that computes compression force from engineering strain.

Author(s)

Dan Kelley

References

Campbell-Malone, Regina. "Biomechanics of North Atlantic Right Whale Bone: Mandibular Fracture as a Fatal Endpoint for Blunt Vessel-Whale Collision Modeling." PhD Thesis, Massachusetts

Institute of Technology and Woods Hole Oceanographic Institution, 2007. doi:10.1575/1912/1817.

Daoust, Pierre-Yves, Emilie L. Couture, Tonya Wimmer, and Laura Bourque. "Incident Report. North Atlantic Right Whale Mortality Event in the Gulf of St. Lawrence, 2017." Canadian Wildlife Health Cooperative, Marine Animal Response Society, and Fisheries and Oceans Canada, 2018. https://publications.gc.ca/site/eng/9.850838/publication.html.

Grear, Molly E., Michael R. Motley, Stephanie B. Crofts, Amanda E. Witt, Adam P. Summers, and Petra Ditsche. "Mechanical Properties of Harbor Seal Skin and Blubber - a Test of Anisotropy." Zoology 126 (2018): 137-44. doi:10.1016/j.zool.2017.11.002.

Raymond, J. J. "Development of a Numerical Model to Predict Impact Forces on a North Atlantic Right Whale during Collision with a Vessel." University of New Hampshire, 2007. https://scholars.unh.edu/thesis/309/.

Examples

```
parms <- parameters()
epsilon <- seq(0, 1, length.out = 100) # strain
sigma <- parms$stressFromStrain(epsilon) # stress
plot(epsilon, log10(sigma), xlab = "Strain", ylab = "log10(Stress [MPa])", type = "l")
mtext("Note sudden increase in stress, when bone compression starts")
```

---

pin *Pin numerical values between stated limits*

Description

Pin numerical values between stated limits

Usage

```
pin(x, lower = NULL, upper = NULL)
```

Arguments

x	Vector or matrix of numerical values
lower	Numerical values of minimum value allowed; set to `NULL` to avoid trimming the lower limit.
upper	As for `lower`, but for the upper limit.

Value

Copy of `x`, with any value that exceeds `lim` having been replaced by `lim`.

Author(s)

Dan Kelley

`plot.strike` *Plot a strike object*

Description

Creates displays of various results of a simulation performed with `strike()`.

Usage

```
## S3 method for class 'strike'
plot(
  x,
  which = "default",
  drawEvents = TRUE,
  colwcenter = "black",
  colwinterface = "black",
  colwskin = "black",
  cols = "black",
  colThreat = c("white", "lightgray", "darkgray", "black"),
  lwd = 1,
  D = 3,
  debug = 0,
  ...
)
```

Arguments

`x` An object created by `strike()`.

`which` A character vector that indicates what to plot. This choices for its entries are listed below, in no particular order.

- `"location"` for a time-series plot of boat location `xw` in dashed black, whale centerline `xs` in solid gray, blubber-interior interface in red, and skin in blue. The maximum acceleration of ship and whale (in "g" units) are indicated in notes placed near the horizontal axes. Those acceleration indications report just a single value for each of ship and whale, but if the blubber and sublayer have been squeezed to their limits, yielding a short and intense force spike as the bone compresses, then the summaries will also report on the spike duration and intensity. The spike is computed based on using `runmed()` on the acceleration data, with a `k` value that is set to correspond to 5 ms, or to k=11, whichever is larger.
- `"section"` to plot skin thickness, blubber thickness and sublayer thickness in one panel, creating a cross-section diagram.
- `"threat"` a stacked plot showing time-series traces of an ad-hoc measure of the possible threat to skin, blubber and sublayer. The threat level is computed as the ratio of stress to ultimate strength, e.g. for blubber, it is `x$WCF$stress/x$parms$s[2]`. The same vertical scale is used in each of

the subpanels that make up the stack. Any values exceeding 10 are clipped to 10, and in such a case the overall label on the vertical axis will note this clipping, although it should be easy to see, because the way it most often occurs is if the soft layers "bottom out" onto the bone, which yields a short period of very high stress, owing to the very high compression modulus of bone. Each of the curves is filled in with a light gray colour for stress/strength values up to 1, and with black for higher values; this makes it easy to tell at a glance whether the threat level is high.

- `"whale acceleration"` for a time-series plot of whale acceleration.
- `"blubber thickness"` for a time-series plot of blubber thickness.
- `"sublayer thickness"` for a time-series plot of the thickness of the layer interior to the blubber.
- `"reactive forces"` for a time-series showing the reactive forces associated with skin stretching (solid) and the compression of the blubber and sublayer components (dashed).
- `"compression stress"` for a time-series plot of the compression stress on the blubber and the layer to its interior. (These stresses are equal, under an equilibrium assumption.)
- `"skin stress"` for a time-series of skin stress in the along-skin y and z directions.
- `"lethality index"` for a time-series of Lethality Index, computed from compression stress using `lethalityIndexFromStress()`. Values of Lethality Index that exceed 0.5 are highlighted, and a dotted line is drawn at that value.
- `"values"` for a listing of `param` values.
- `"all"` for all of the above.
- `"default"` for a three-element plot showing `"location"`, `"section"`, and `"threat"`.

`drawEvents` Logical, indicating whether to draw lines for some events, such as the moment of closest approach.

`colwcenter` Colour used to indicate the whale centre.

`colwinterface` Colour used to indicate the interface between whale blubber and sublayer.

`colwskin` Colour used to indicate the whale skin.

`cols` As `colw`, but the colour to be used for the ship bow location, which is drawn with a dashed line.

`colThreat` a 4-element colour specification used in `"threat"` plots. The colour transitions depend on the layer being plotted. For skin and bone (i.e. in the bottom and top subpanels of the plot), the first colour is used up to a stress:strength ratio of 1/4, the second up to 1/2, the third up to 3/4, and the fourth above 3/4. For blubber and sublayer (i.e. the second and third subpanels, counting from the bottom), the colours are defined by comparing compressive stress to the values of `tau25`, `tau50` and `tau75` (see the `parameters()` documentation), with the first colour for stress under `tau25`, the second for stress up to `tau50`, the third for stress up to `tau75`, and the fourth for higher stresses.

`lwd` Line width used in plots for time intervals in which damage criteria are not exceeded.

D	Factor by which to thicken lines during times during which damage criteria are exceeded.
debug	Integer indicating debugging level, 0 for quiet operation and higher values for more verbose monitoring of progress through the function.
...	Ignored.

Value

None. This function is called to add to a plot.

Author(s)

Dan Kelley

References

See `whalestrike()` for a list of references.

Examples

```
# Plot lethality index
t <- seq(0, 0.7, length.out = 200)
state <- c(xs = -2, vs = knot2mps(12), xw = 0, vw = 0) # 12 knot ship
parms <- parameters() # default values
sol <- strike(t, state, parms)
plot(sol, which="lethality index")
```

`raymond2007` *Whale blubber stress-strain relationship*

Description

This is a data frame with elements `strain` and `stress`, found by digitizing (accurate to perhaps 1 percent) the curve shown in Figure 2.13 of Raymond (2007). It is used to develop a stress-strain relationship used by `parameters()`, as shown in “Examples”.

References

Raymond, J. J. "Development of a Numerical Model to Predict Impact Forces on a North Atlantic Right Whale during Collision with a Vessel." University of New Hampshire, 2007. https://scholars.unh.edu/thesis/309/.

Examples

```
data(raymond2007)
attach(raymond2007)
# Next yields \code{a=1.64e5} Pa and \code{b=2.47}.
m <- nls(stress ~ a * (exp(b * strain) - 1), start = list(a = 1e5, b = 1))
plot(strain, stress, xaxs = "i", yaxs = "i")
x <- seq(0, max(strain), length.out = 100)
lines(x, predict(m, list(strain = x)))
```

---

shipAreaFromMass — *Compute ship wetted area from mass*

Description

Estimate the wetted area of a Cape Islander boat, given the vessel mass.

Usage

```
shipAreaFromMass(ms)
```

Arguments

`ms`	Ship mass (kg).

Details

The method is based on scaling up the results for a single Cape Islander ship, of displacement 20.46 tonnes, length 11.73m, beam 4.63m, and draft 1.58m, on the assumption that the wetted area is proportional to $length * (2 * draft + beam)$. This reference area is scaled to the specified mass, `ms`, by multiplying by the 2/3 power of the mass ratio.

Note that this is a crude calculation meant as a stop-gap measure, for estimates values of the `Ss` argument to `parameters()`. It should not be used in preference to inferences made from architectural drawings of a given ship under study.

Value

Estimated area (m^2).

Author(s)

Dan Kelley

See Also

Other functions relating to ship characteristics: `shipLength()`, `shipMassFromLength()`, `shipWaterForce()`

shipLength — *Nominal ship length in m*

Description

This is based on the "Average LOA in m" column in Table 1 of Mayette and Brillant (2026).

Usage

```
shipLength(type = NULL)
```

Arguments

`type`	either (1) a string identifying the ship type, in which case the average overall length of the named vessel is returned, or (2) NULL, in which case a data frame containing type and length is returned.

Value

`shipLength` returns ship length in m, as defined in Mayette and Brillant (2026).

Author(s)

Dan Kelley, with help from Alexandra Mayette

References

Mayette, Alexandra, and Sean W. Brillant. "A Regression-Based Method to Estimate Vessel Mass for Use in Whale-Ship Strike Risk Models." PloS One 21, no. 1 (2026): e0339760. doi:10.1371/journal.pone.0339760.

See Also

Other functions relating to ship characteristics: `shipAreaFromMass()`, `shipMassFromLength()`, `shipWaterForce()`

Examples

```
library(whalestrike)
# An individual length
shipLength("Fishing")
# A table of lengths
shipLength()
```

`shipMassFromLength` *Ship displacement in kg based on vessel type and length*

Description

This is done using formulae in Table 3 of Mayette and Brillant (2026).

Usage

```
shipMassFromLength(type = NULL, L)
```

Arguments

`type`	either (1) a string identifying the ship type, in which case the average overall length of the named vessel is returned, or (2) NULL, in which case a vector of permitted values of `type` is returned.
`L`	vessel length in metres.

Details

The formulae used are as follows.

`type`	Formula
"Bulk Carrier"	5.64 * L^3.06
"Container Ship"	86.40 * L^2.46
"Cruise"	97.51 * L^2.28
"Ferry"	25.15 * L^2.62
"Fishing"	0.71 * L^3.79
"Government/Research"	2.95 * L^3.22
"Other"	2.64 * L^3.35
"Passenger"	4.32 * L^3.08
"Pleasure Craft"	34.47 * L^2.68
"Sailing"	1.23 * L^3.53
"Tanker"	7.25 * L^3.03
"Tug"	104.48 * L^2.51

Value

`shipMassFromLength` returns ship displacement mass (in kg), according to Mayette and Brillant (2026) Table 3.

Author(s)

Dan Kelley, with help from Alexandra Mayette

References

Mayette, Alexandra, and Sean W. Brillant. "A Regression-Based Method to Estimate Vessel Mass for Use in Whale-Ship Strike Risk Models." PloS One 21, no. 1 (2026): e0339760. doi:10.1371/journal.pone.0339760.

See Also

Other functions relating to ship characteristics: `shipAreaFromMass()`, `shipLength()`, `shipWaterForce()`

Examples

```
library(whalestrike)
shipMassFromLength("Tug", 50) / 1e3 # 1920.648
```

---

shipWaterForce *Ship Drag Force*

Description

Compute the retarding force of water on the ship, based on a drag law $(1/2) * rho * Cs * A * vs^2$ where `rho` is water density taken to be 1024 (kg/m^3), `Cs` is drag coefficient stored in `parms` and `A` is area, also stored in `parms, and vs`' is the ship speed (m/s).

Usage

```
shipWaterForce(vs, parms)
```

Arguments

`vs`	Ship speed in m/s. (Consider using `knot2mps()` if you prefer to think of speeds in knots.)
`parms`	A named list holding model parameters, created by `parameters()`.

Value

Water drag force (N).

Author(s)

Dan Kelley

See Also

Other functions relating to ship characteristics: `shipAreaFromMass()`, `shipLength()`, `shipMassFromLength()`

Other functions relating to forces: `stressFromStrainFunction()`, `whaleCompressionForce()`, `whaleSkinForce()`, `whaleWaterForce()`

sol20200708 *Reference strike() solution*

Description

This was produced with the package as it existed on 2020-jul-8, prior to the publication of Kelley et al. (2021). It is used in testing, to ensure that the package does not inadvertently change in its predictions.

References

Kelley, Dan E., James P. Vlasic, and Sean W. Brillant. "Assessing the Lethality of Ship Strikes on Whales Using Simple Biophysical Models." Marine Mammal Science, 37(1), 2021, mms.12745. doi:10.1111/mms.12745.

Examples

```
# Three plots
data(sol20200708)
plot(sol20200708)
```

stressFromLethalityIndex *Compute stress, based on lethality index*

Description

The model used for this is the logistic model, fitting observed injury/lethality statistics to the base-10 logarithm of the maximum compression stress during a simulated impact event.

Usage

```
stressFromLethalityIndex(injury)
```

Arguments

injury	numerical value or vector, giving threat of injury (in range 0 to 1).

Value

whale compression stress, in Pascals.

Author(s)

Dan Kelley

See Also

Other functions dealing with Whale Lethality index: `lethalityIndexFromStress()`, `maximumLethalityIndex()`

Examples

```
stressFromLethalityIndex(0.5) # approx. 254000 Pa, i.e. parameters()$logistic$tau50
```

---

stressFromStrainFunction

Create a function for stress in laminated layers

Description

Denoting unforced layer thickness in the i layer as l_i and strain there as $\epsilon_i = \Delta l_i / l_i$, we may write the stress-strain relationship as

$$\sigma = a_i * (exp(b_i * \epsilon_i) - 1)$$

for each layer, where it is assumed that stress σ is equal across layers. Inverting this yields

$$\epsilon_i = ln(1 + \sigma / a_i) / b_i$$

where ln is the natural logarithm. Therefore, the change ΔL in the total thickness $L = \sum l_i$ may be written

$$0 = \Delta L - \sum ((l_i / b_i) ln(1 + \sigma / a_i))$$

. Note that zero-thickness layers are removed from the calculation, to avoid spurious forces.

Usage

```
stressFromStrainFunction(l, a, b, N = 1000)
```

Arguments

`l`	vector of layer thicknesses
`a`	vector of multipliers
`b`	vector of e-fold parameters
`N`	integer specifying how many segments to use in the spline

Details

This expression is not easily inverted to get σ in terms of ΔL but it may be solved easily for particular numerical values, using `uniroot()`.

This is done for a sequence of `N` values of strain ϵ that range from 0 to 1. Then `approxfun()` is used to create a piecewise-linear representation of the relationship between σ and ΔL, which becomes the return value of the present function. (The purpose of using a piecewise-linear representation to reduce computation time.)

Value

A piecewise-linear function, created with approxfun(), that returns stress as a function of total strain of the system of compressing layers. For the purposes of the whale-strike analysis, the strain should be between 0 and 1, i.e. there is no notion of compressing blubber, etc. to negative thickness.

Author(s)

Dan Kelley

See Also

Other functions relating to whale characteristics: whaleCompressionForce(), whaleLengthFromMass(), whaleMassFromLength(), whaleSkinForce(), whaleWaterForce()

Other functions relating to forces: shipWaterForce(), whaleCompressionForce(), whaleSkinForce(), whaleWaterForce()

Examples

```
library(whalestrike)
# Set blubber parameters for each layer, to see if
# we recover the raymond2007 data.
param <- parameters(a = rep(1.64e5, 4), b = rep(2.47, 4))
x <- seq(0, 0.5, length.out = 100)
y <- param$stressFromStrain(x)
plot(x, y, type = "l", lwd = 4, col = "gray")
data("raymond2007")
points(raymond2007$strain, raymond2007$stress, col = 2)
```

---

strike *Simulate the collision of a ship and a whale*

Description

Newtonian mechanics are used, taking the ship as non-deformable, and the whale as being cushioned by a skin layer and a blubber layer. The forces are calculated by shipWaterForce(), whaleSkinForce(), whaleCompressionForce(), and whaleWaterForce() and the integration is carried out with deSolve::lsoda().

Usage

```
strike(t, state, parms, debug = 0)
```

Arguments

- `t` — a suggested vector of times (s) at which the simulated state will be reported. This is only a suggestion, however, because `strike` is set up to detect high accelerations caused by bone compression, and may set a finer reporting interval, if such accelerations are detected. The detection is based on thickness of compressed blubber and sublayer; if either gets below 1 percent of the initial(uncompressed) value, then a trial time grid is computed, with 20 points during the timescale for bone compression, calculated as $0.5 * sqrt(Ly * Lz * a[4] * b[4]/(l[4] * mw)$, with terms as discussed in the documentation for `parameters()`. If this trial grid is finer than the grid in the `t` parameter, then the simulation is redone using the new grid. Note that this means that the output will be finer, so code should not rely on the output time grid being
- `state` — A list or named vector holding the initial state of the model: ship position `xs` (m), ship speed `vs` (m/s), whale position `xw` (m), and whale speed `vw` (m/s).
- `parms` — A named list holding model parameters, created by `parameters()`.
- `debug` — Integer indicating debugging level, 0 for quiet operation and higher values for more verbose monitoring of progress through the function.

Value

`strike` returns an object of class `"strike"`, which is a list holding 13 items. Nine of these items are time-series vectors, namely time `t`, ship position `xs`, ship speed `vs`, whale position `xw`, whale speed `vw`, ship acceleration `dvsdt`, whale acceleration `dvwdt`, drag force on the ship `SWF`, and drag force on the whale `WWF`. In addition to these time-series items, there are 3 items that are lists: `WSF` holds time-series of surface forces on the whale; `WCF` holds time-series of compressive forces on the whale; and `parameters` holds parameters of the simulation (see `parameters()` for more information). Finally, there is a logical value named `refinedGrid` that indicates whether the simulation required a restart because the initial timestep proved inadequate to track the high forces that may arise quickly if bone starts to be compressed appreciably. All quantities are in SI units, s for time, m/s for speed, m/s^2 for acceleration, N for force, etc.

Author(s)

Dan Kelley

References

See `whalestrike()` for a list of references.

Examples

```
library(whalestrike)
# Example 1: three plots, as in the default three panels of app()
t <- seq(0, 0.7, length.out = 200)
state <- list(xs = -2, vs = knot2mps(10), xw = 0, vw = 0) # ship speed 10 knots
parms <- parameters()
sol <- strike(t, state, parms)
plot(sol)
```

```
# Example 2: time-series plots of blubber stress and stress/strength,
# for a 200 tonne ship moving at 10 knots
t <- seq(0, 0.7, length.out = 1000)
state <- list(xs = -2, vs = knot2mps(10), xw = 0, vw = 0) # ship 10 knots
parms <- parameters(ms = 200 * 1000) # 1 metric tonne is 1000 kg
sol <- strike(t, state, parms)
plot(t, sol$WCF$stress / 1e6,
    type = "l",
    xlab = "Time [s]", ylab = "Blubber stress [MPa]"
)
plot(t, sol$WCF$stress / sol$parms$s[2],
    type = "l",
    xlab = "Time [s]", ylab = "Blubber stress / strength"
)

# Example 3: max stress and stress/strength, for a 200 tonne ship
# moving at various speeds.
knots <- seq(0, 10, 1)
maxStress <- NULL
maxStressOverStrength <- NULL
for (speed in knot2mps(knots)) {
    t <- seq(0, 10, length.out = 1000)
    state <- list(xs = -2, vs = speed, xw = 0, vw = 0)
    parms <- parameters(ms = 200 * 1000) # 1 metric tonne is 1000 kg
    sol <- strike(t, state, parms)
    maxStress <- c(maxStress, max(sol$WCF$stress))
  maxStressOverStrength <- c(maxStressOverStrength, max(sol$WCF$stress / sol$parms$s[2]))
}
nonzero <- maxStress > 0
plot(knots[nonzero], log10(maxStress[nonzero]),
    type = "o", pch = 20, xaxs = "i", yaxs = "i",
    xlab = "Ship Speed [knots]", ylab = "log10 peak blubber stress"
)
abline(h = log10(sol$parms$s[2]), lty = 2)
plot(knots[nonzero], log10(maxStressOverStrength[nonzero]),
    type = "o", pch = 20, xaxs = "i", yaxs = "i",
    xlab = "Ship Speed [knots]", ylab = "log10 peak blubber stress / strength"
)
abline(h = 0, lty = 2)
```

`summary.parameters` *Summarize a parameters object*

Description

This provides an overview of the contents of an object created with `parameters()`.

Usage

```
## S3 method for class 'parameters'
summary(object, ...)
```

Arguments

`object`	an object of class `"parameters"`, as created with `parameters()`.
`...`	ignored

Value

`summary.parameters` returns nothing. It is called for its side effect of printing information about the parameters in a ship-whale collision simulation.

Author(s)

Dan Kelley

Examples

```
summary(parameters())
```

---

summary.strike *Summarize the parameters of a simulation, and its results*

Description

Summarize the parameters of a simulation, and its results

Usage

```
## S3 method for class 'strike'
summary(object, ...)
```

Arguments

`object`	an object of class `"strike"`, as created by `strike()`.
`...`	ignored

Value

None. This function is called for its side-effect of printing information about the ship-whale collision simulation.

Author(s)

Dan Kelley

Examples

```
library(whalestrike)
# Example 1: graphs, as in the shiny app
t <- seq(0, 0.7, length.out = 200)
state <- list(xs = -2, vs = knot2mps(10), xw = 0, vw = 0) # ship speed 10 knots
parms <- parameters()
sol <- strike(t, state, parms)
summary(sol)
```

---

updateParameters   *Update parameters*

Description

`updateParameters()` is used to alter one or more components of an existing object of type "parameters" that was created by `parameters()`. This can be useful for e.g. sensitivity tests (see "Details").

Usage

```
updateParameters(
  original,
  ms,
  Ss,
  Ly,
  Lz,
  species,
  lw,
  mw,
  Sw,
  l,
  a,
  b,
  s,
  theta,
  Cs,
  Cw,
  logistic,
  debug = 0
)
```

Arguments

`original`	An object of class "parameters", as created by `parameters()` and perhaps later altered by previous calls to `updateParameters()`.
`ms`	Ship mass (kg).

`Ss` Ship wetted area (m^2). This, together with `Cs`, is used by `shipWaterForce()` to estimate ship drag force. If `Ss` is not given, then an estimate is made by calling `shipAreaFromMass()` with the provided value of `ms`.

`Ly` Ship impact horizontal extent (m); defaults to 1.15m if not specified, based on an analysis of the shape of the bow of typical coastal fishing boats of the Cape Islander variety.

`Lz` Ship impact vertical extent (m); defaults to 1.15m if not specified, based on the same analysis as for Ly.

`species` a string indicating the whale species. For the permitted values, see `whaleMassFromLength()`. (The `species` value can also set the `lw` and `l` values, as noted in their portions of this documentation.)

`lw` either (1) whale length in metres or (2) the string `"from_species"`. If the latter, then the length is determined from `whaleMeasurements()`. In either case, the length is used by `whaleAreaFromLength()` to calculate area, which is needed for the water drag calculation done by `whaleWaterForce()`.

`mw` either (1) the whale mass in kg or (2) NULL. In the latter case, the mass is calculated from whale length, using `whaleMassFromLength()` with `type="wetted"`.

`Sw` either (1) the whale surface area in m^2 or (2) NULL. If the latter case, the area is calculated from whale length using `whaleAreaFromLength()`.

`l` either (1) a numerical vector of length 4 that indicates the thicknesses in metres of skin, blubber, sublayer and bone; (2) NULL to set these four values to 0.025, 0.16, 1.12, and 0.1; or (3) the string `"from_species"`, in which case these four values are determined by calling `whaleMeasurements()`. The default skin thickness of 0.025 m represents the 0.9-1.0 inch value stated in Section 2.2.3 of Raymond (2007). The blubber default of 0.16 m is a rounded average of the values inferred by whale necropsy, reported in Appendix 2 of Daoust et al., 2018. The sublayer default of 1.12 m may be reasonable at some spots on the whale body. The bone default of 0.1 m may be reasonable at some spots on the whale body. The sum of these default values, 1.40 m, is a whale radius that is consistent with a half-circumference of 4.4 m, reported in Table 2.2 of Raymond (2007). Note, however, that these values are not identical to those found in `whaleMeasurements`.

`a, b` Numerical vectors of length 4, giving values to use in the stress-strain law `stress=a*(exp(b*strain)-1)`, where `a` is in Pa and `b` is unitless. By construction, `a*b` is the local modulus at low strain (i.e. at low `b*strain` values), and that `b` is the efolding scale for nonlinear increase in stress with strain. This exponential relationship has been mapped out for whale blubber, using a curve fit to Figure 2.13 of Raymond (2007), and these values are used for the second layer (blubber); see the documentation for the raymond2007 dataset, to see for how that fit was done. If not provided, `a` defaults to `c(17.8e6/0.1, 1.58e5, 1.58e5, 8.54e8/0.1)` and `b` defaults to `c(0.1, 2.54, 2.54, 0.1)`. The skin defaults are set up to give a linear shape (since `b` is small) with the `a*b` product being 17.8e6 Pa, which is the adult-seal value given in Table 3 of Grear et al. (2017). The blubber defaults are from a regression of the stress-strain relationship shown in Figure 2.13 of Raymond (2007). The sublayer defaults are set to match those of blubber, lacking any other information. The bone default for `b` is small, to set up a linear function, and a*b is set to equal 8.54e8 Pa, given in Table 2.3 of Raymond (2007) and Table 4.5 of Campbell-Malone (2007).

s	Numerical vector of length 4, giving the ultimate strengths (Pa) of skin, blubber, sublayer, and bone, respectively. If not provided, the value is set to 1e6 * c(19.600,0.255,0.255,22.900) with reasoning as follows. The skin default of 19.6 MPa is a rounded value from Table 3 of Grear et al. (2018) for adult seal skin strength at an orientation of 0 degrees. The blubber and sublayer values were chosen as the central point of a logistic fit of whale collision damage to maximal stress during a default impact simulation. (For comparison, a strength of 0.437 MPa may be inferred by multiplying Raymond's (2007) Figure 2.13 elastic modulus of 0.636 MPa by the ratio 0.97/1.41 determined for adult seal strength/modulus, as reported in Table 3 of Grear et al. (2018).) The bone default o 22.9 MPa is from Table 2.3 of Raymond (2007) and Table 4.5 of Campbell-Malone (2007).

theta	Whale skin deformation angle (deg); defaults to 55 degrees, if not supplied, because that angle produces a good match to Raymond's (2007) Figure 6.1 for the total force as a function of vessel speed, for large vessels. Note that the match works almost as well in the range 50 deg to 70 deg.

Cs	Drag coefficient for ship (dimensionless), used by shipWaterForce() to estimate ship drag force. Defaults to 1e-2, which is 4 times the frictional coefficient of 2.5e-3 inferred from Figure 4 of Manen and van Oossanen (1988), assuming a Reynolds number of 5e7, computed from speed 5m/s, lengthscale 10m and viscosity 1e-6 m^2/s. The factor of 4 is under the assumption that frictional drag is about a quarter of total drag. The drag force is computed with shipWaterForce().

Cw	Drag coefficient for whale (dimensionless), used by whaleWaterForce() to estimate whale drag force. Defaults to 2.5e-3, for Reynolds number 2e7, computed from speed 2 m/s, lengthscale 5m which is chosen to be between radius and length, and viscosity 1e-6 m^2/s. The drag force is computed with whaleWaterForce().

logistic	a list containing logStressCenter and logStressWidth, which define an empirical logistic fit of an index of whale injury in observed strikes (ranging from 0 for no injury to 1 for fatal injury), as a function of the base-10 logarithm of compressive stress, as well as tau25, tau50 and tau75, which are the stresses in that fit that yield index values of 0.25, 0.50 and 0.75, respectively; these values set colour boundaries in plot.strike() plots that have which="threat".

debug	Integer indicating debugging level, 0 for quiet operation and higher values for more verbose monitoring of progress through the function.

Details

Two important differences between argument handling in updateParameters() and parameters() should be kept in mind.

First, updateParameters() does not check its arguments for feasible values. This can lead to bad results when using strike(), which is e.g. expecting four layer thicknesses to be specified, and also that each thickness is positive.

Second, updateParameters() does not perform ancillary actions that parameters() performs, with regard to certain interlinking argument values. Such actions are set up for whale length and ship mass, which are easily-observed quantities from other quantities can be estimated using whalestrike functions. If lw (whale length) is supplied to parameters() without also supplying mw (whale mass), then parameters() uses whaleMassFromLength() to infer mw from lw. The same procedure is used to infer Sw if it is not given, using whaleAreaFromLength(). Similarly, parameters() uses shipAreaFromMass() to compute Ss (ship area) from ms (ship mass), if the Ss argument is not given. Importantly, these three inferences are *not* made by updateParameters(), which alters only those values that are supplied explicitly. It is easy to supply those values, however; for example,

```
parms <- updateParameters(PARMS, lw=1.01 * PARMS$lw))
parms <- updateParameters(parms, mw=whaleMassFromLength(parms$lw))
parms <- updateParameters(parms, Sw=whaleAreaFromLength(parms$lw))
```

modifies a base state stored in PARMS, increasing whale length by 1% and then increasing whale mass and area accordingly. This code block is excerpted from a sensitivity test of the model, in which

```
parms <- updateParameters(PARMS, ms=1.01 * PARMS$ms)
parms <- updateParameters(parms, Ss=shipAreaFromMass(parms$ms))
```

was also used to perturb ship mass (and inferred area).

Value

A named list holding the items of the same name as those in the list returned by parameters().

Author(s)

Dan Kelley

References

Daoust, Pierre-Yves, Emilie L. Couture, Tonya Wimmer, and Laura Bourque. "Incident Report. North Atlantic Right Whale Mortality Event in the Gulf of St. Lawrence, 2017." Canadian Wildlife Health Cooperative, Marine Animal Response Society, and Fisheries and Oceans Canada, 2018. https://publications.gc.ca/site/eng/9.850838/publication.html.

---

whaleAreaFromLength *Whale projected area, as function of length*

Description

This depends on calculations based on the digitized shape of a whale necropsy, which is provided by whaleShape(). The results are $0.143 * L^2$ for the projected area (see reference 1) and $0.448 * (0.877 * L)^2$ for the wetted area (see reference 2, but note that we use a correction related to whale mass).

Usage

```
whaleAreaFromLength(L, species = "N. Atl. Right Whale", type = "wetted")
```

Arguments

L	whale length in metres.
species	a string indicating the whale species. In the present version of the package, this parameter is ignored, and it is assumed that the formula developed for North Atlantic Right Whales will be applicable to other species. This is not a large concern, because the area only affects the water drag, which will not be large during the short interval of a ship impact.
type	character string indicating the type of area, with `"projected"` for a side-projected area, and `"wetted"` for the total wetted area. The wetted area was computed by mathematically spinning a spline fit to the side-view. In both cases, the original data source is the necropsy side-view presented in Daoust et al. (2018).

Details

Note that multiple digitizations were done, and that the coefficients used in the formulae agreed to under 0.7 percent percent between these digitizations.

Value

`whaleAreaFromLength` returns the surface area of the whale, in square metres.

Author(s)

Dan Kelley

References

1. Dan Kelley's internal document `dek/20180623_whale_area.Rmd`, available upon request.
2. Dan Kelley's internal document `dek/20180707_whale_mass.Rmd`, available upon request.
3. Daoust, Pierre-Yves, Emilie L. Couture, Tonya Wimmer, and Laura Bourque. "Incident Report. North Atlantic Right Whale Mortality Event in the Gulf of St. Lawrence, 2017." Canadian Wildlife Health Cooperative, Marine Animal Response Society, and Fisheries and Oceans Canada, 2018. https://publications.gc.ca/site/eng/9.850838/publication.html.

Examples

```
L <- 3:20
A <- whaleAreaFromLength(L)
plot(L, A, xlab = "Length [m]", ylab = "Area [m^2]", type = "l")
```

whaleCompressionForce *Whale compression force*

Description

Calculate the total compression stress and force, along with the thicknesses of skin, blubber, sublayer, and bone. The stress is computed with the `stressFromStrainFunction()` function that is created by `parameters()` and stored in `para`. the force is computed by multiplying stess by area computed as the product of `parms$Ly` and `parms$Lz`. Any negative layer thicknesses are set to zero, as a way to avoid problems with aphysical engineering compression strains that exceed 1.

Usage

```
whaleCompressionForce(xs, xw, parms)
```

Arguments

`xs`	Ship position (m).
`xw`	Whale position (m).
`parms`	A named list holding model parameters, created by `parameters()`.

Value

A list containing: `force` (N), the compression-resisting force; `stress` (Pa), the ratio of that force to the impact area; `strain`, the total strain, and `compressed`, a four-column matrix (m) with first column for skin compression, second for blubber compression, third for sublayer compression, and fourth for bone compression.

Author(s)

Dan Kelley

References

See `whalestrike()` for a list of references.

See Also

Other functions relating to whale characteristics: `stressFromStrainFunction()`, `whaleLengthFromMass()`, `whaleMassFromLength()`, `whaleSkinForce()`, `whaleWaterForce()`

Other functions relating to forces: `shipWaterForce()`, `stressFromStrainFunction()`, `whaleSkinForce()`, `whaleWaterForce()`

`whaleLengthFromMass` *Compute whale length from mass*

Description

This works by inverting `whaleMassFromLength()` using `uniroot()`.

Usage

```
whaleLengthFromMass(M, species = "N. Atl. Right Whale", model = "fortune2012")
```

Arguments

`M`	Whale mass (kg).
`species`	A string indicating the whale species (see `whaleMassFromLength()` for details).
`model`	Character string specifying the model (see `whaleMassFromLength()` for details).

Value

Whale length (m).

Author(s)

Dan Kelley

References

See `whalestrike()` for a list of references.

See Also

`whaleMassFromLength()` is the reverse of this.

Other functions relating to whale characteristics: `stressFromStrainFunction()`, `whaleCompressionForce()`, `whaleMassFromLength()`, `whaleSkinForce()`, `whaleWaterForce()`

whaleMassFromLength *Whale mass inferred from length*

Description

Calculate an estimate of the mass of different species of whale, based on animal length, based on formulae as listed in "Details".

Usage

```
whaleMassFromLength(L, species = "N. Atl. Right Whale", model = NULL)
```

Arguments

`L`	whale length in m.
`species`	character value specifying the species (see "Details"). If only one value is given, then it will repeated to have the same length as `L`. Otherwise, its length must match the length of `L`.
`model`	either NULL (the default), to choose a model based on the particular species, or a character value specifying the model (see "Details"). If only one value is given, then it will repeated to have the same length as `L`. Otherwise, its length must match the length of `L`.

Details

The permitted values for `model` and `species` are as follows. Note that if `model` is not provided, then Fortune (2012) is used for `"N. Atl. Right Whale"`, and Lockyer (1976) is used for all other species.

- `"moore2005"` (which only works if `species` is `"N. Atl. Right Whale"`) yields $242.988 * exp(0.4 * length)$, which (apart from a unit change on `L`) is the regression equation shown above Figure 1d in Moore et al. (2005) for right whales. A difficult in the Moore et al. (2005) use of a single nonzero digit in the multiplier on `L` is illustrated in "Examples".
- `"fortune2012"` with `species="N. Atl. Right Whale"` yields the formula $exp(-10.095 + 2.825 * log(100 * L))$ for North Atlantic right whales, according to a corrected version of the erroneous formula given in the caption of Figure 4 in Fortune et al (2012). (The error, an exchange of slope and intercept, was confirmed by S. Fortune in an email to D. Kelley dated June 22, 2018.)
- `"fortune2012"` with `species="N. Pac. Right Whale"` yields the formula $exp(-12.286 + 3.158 * log(100 * L))$ for North Pacific right whales, according to a corrected version of the erroneous formula given in the caption of Figure 4 in Fortune et al (2012). (The error, an exchange of slope and intercept, was confirmed by S. Fortune in an email to D. Kelley dated June 22, 2018.)
- `"lockyer1976"` uses formulae from Table 1 of Lockyer (1976). The permitted `species` and the formulae used are as follows (note that the `"Gray Whale"` formula is in the table's caption, not in the table itself).

  - "Blue Whale": $2.899L^{3.25}$
  - "Bryde Whale": $12.965L^{2.74}$
  - "Fin Whale": $7.996L^{2.90}$
  - "Gray Whale": $5.4L^{3.28}$
  - "Humpback Whale": $16.473L^{2.95}$
  - "Minke Whale": $49.574L^{2.31}$
  - "Pac. Right Whale": $13.200L^{3.06}$
  - "Sei Whale": $25.763L^{2.43}$
  - "Sperm Whale": $6.648L^{3.18}$

Value

Mass in kg.

Author(s)

Dan Kelley

References

- Lockyer, C. "Body Weights of Some Species of Large Whales." J. Cons. Int. Explor. Mer. 36, no. 3 (1976): 259-73.
- Moore, M.J., A.R. Knowlton, S.D. Kraus, W.A. McLellan, and R.K. Bonde. "Morphometry, Gross Morphology and Available Histopathology in North Atlantic Right Whale (Eubalaena Glacialis) Mortalities (1970 to 2002)." Journal of Cetacean Research and Management 6, no. 3 (2005): 199-214.
- Fortune, Sarah M. E., Andrew W. Trites, Wayne L. Perryman, Michael J. Moore, Heather M. Pettis, and Morgan S. Lynn. "Growth and Rapid Early Development of North Atlantic Right Whales (Eubalaena Glacialis)." Journal of Mammalogy 93, no. 5 (2012): 1342-54. doi:10.1644/11MAMMA297.1.

See Also

whaleLengthFromMass() is the reverse of this.

Other functions relating to whale characteristics: stressFromStrainFunction(), whaleCompressionForce(), whaleLengthFromMass(), whaleSkinForce(), whaleWaterForce()

Examples

```
library(whalestrike)
L <- seq(5, 15, length.out = 100)
kpt <- 1000 # kg per tonne
# Demonstrate (with dashing) the sensitivity involved in the single-digit
# parameter in Moore's formula, and (with colour) the difference to the
# Fortune et al. (2012) formulae.
plot(L, whaleMassFromLength(L, model = "moore2005") / kpt,
    type = "l", lwd = 2,
    xlab = "Right-whale Length [m]", ylab = "Mass [tonne]"
```

```
)
lines(L, 242.988 * exp(0.35 * L) / kpt, lty = "dotted", lwd = 2)
lines(L, 242.988 * exp(0.45 * L) / kpt, lty = "dashed", lwd = 2)
lines(L, whaleMassFromLength(L,
    species = "N. Atl. Right Whale",
    model = "fortune2012"
) / kpt, col = 2, lwd = 2)
lines(L, whaleMassFromLength(L,
    species = "N. Pac. Right Whale",
    model = "fortune2012"
) / kpt, col = 3, lwd = 2)
grid()
legend("topleft",
    bg = "white", y.intersp = 1.4, lwd = 2, col = 1:3,
    legend = c("moore2005", "fortune2012 Atlantic", "fortune2012 Pacific")
)

# Emulate Figure 1 of Lockyer (1976), with roughly-chosen plot limits.
L <- seq(0, 18, 0.5)
m <- whaleMassFromLength(L, species = "Pac. Right Whale", model = "lockyer1976") / kpt
plot(L, m,
    col = 1, xlab = "Length [m]", ylab = "Mass [tonne]", type = "l", lwd = 2,
    xaxs = "i", yaxs = "i", xlim = c(3, 30), ylim = c(0, 180)
)
L <- seq(0, 28, 0.5)
m <- whaleMassFromLength(L, species = "Blue Whale", model = "lockyer1976") / kpt
lines(L, m, col = 2, lwd = 2)
L <- seq(0, 24, 0.5)
m <- whaleMassFromLength(L, species = "Fin Whale", model = "lockyer1976") / kpt
lines(L, m, col = 3, lwd = 2)
L <- seq(0, 18, 0.5)
m <- whaleMassFromLength(L, species = "Sei Whale", model = "lockyer1976") / kpt
lines(L, m, col = 1, lty = 2, lwd = 2)
L <- seq(0, 17, 0.5)
m <- whaleMassFromLength(L, species = "Bryde Whale", model = "lockyer1976") / kpt
lines(L, m, col = 2, lty = 2, lwd = 2)
L <- seq(0, 12, 0.5)
m <- whaleMassFromLength(L, species = "Minke Whale", model = "lockyer1976") / kpt
lines(L, m, col = 3, lty = 2, lwd = 2)
L <- seq(0, 17, 0.5)
m <- whaleMassFromLength(L, species = "Humpback Whale", model = "lockyer1976") / kpt
lines(L, m, col = 1, lty = 3, lwd = 2)
L <- seq(0, 18, 0.5)
m <- whaleMassFromLength(L, species = "Sperm Whale", model = "lockyer1976") / kpt
lines(L, m, col = 2, lty = 3, lwd = 2)
L <- seq(0, 15, 0.5)
m <- whaleMassFromLength(L, species = "Gray Whale", model = "lockyer1976") / kpt
lines(L, m, col = 3, lty = 3, lwd = 2)
grid()
legend("topleft",
    bg = "white", y.intersp = 1.4, col = c(1:3, 1:3, 1:2),
    lwd = 2, lty = c(rep(1, 3), rep(2, 3), rep(3, 3)),
   legend = c("Right", "Blue", "Fin", "Sei", "Bryde", "Minke", "Humpback", "Sperm", "Gray")
```

```
)
```

`whaleMeasurements` *Get values for various whale measurements*

Description

This uses a data frame containing information about several whale species, compiled by Alexandra Mayette (2026).

Usage

```
whaleMeasurements(species = NULL)
```

Arguments

`species`	either (1) the name of a species or (2) NULL. In the first case, the table is consulted to find a row with the given species name, and that row is returned. In the second case, the whole table is returned.

Details

There are two species in the table that are not in Mayette's table. These are `"Pac. Right Whale"` and `"Bryde Whale"`. For these, Mayette has suggesting using values for the `"N. Atl. Right Whale"` and `"Sei Whale"` cases, respectively, as conditional estimates for use in this package.

Value

The return value contains

- `name` species name, as used in e.g. whaleMassFromLength().
- `Species` proper species name. (This is not used in this package.)
- `length` whale length in metres.
- `bone` whale bone thickness in metres, measured from the centre to the sublayer.
- `sublayer` thickness of sublayer in meters; this was called `muscle` in Mayette's document.
- `blubber` whale blubber thickness in meters.
- `skin` whale skin thickness in metres.

Author(s)

Dan Kelley, using data and advice from Alexandra Mayette

References

Mayette, A. (2026). Measurements of large whale tissue thickness (Data set). Zenodo. doi:10.5281/zenodo.18764979.

Examples

```
library(whalestrike)
# All species in database
whaleMeasurements()
# A particular species
whaleMeasurements("N. Atl. Right Whale")
```

---

whaleShape *Whale side-view shape*

Description

This is a data frame containing 45 points specifying the shape of a right whale, viewed from the side. It was created by digitizing the whale shape (ignoring fins) that is provided in the necropsy reports of Daoust et al. (2018). The data frame contains x and y, which are distances nondimensionalized by the range in x; that is, x ranges from 0 to 1. The point at the front of the whale is designated as x=y=0.

Usage

```
whaleShape()
```

Value

whaleShape returns a data frame with entries named x and y, which trace out the side-view shape of a Right Whale, using values digitized from a diagram in Daoust et al. (2017).

Author(s)

Dan Kelley

References

Daoust, Pierre-Yves, Emilie L. Couture, Tonya Wimmer, and Laura Bourque. "Incident Report. North Atlantic Right Whale Mortality Event in the Gulf of St. Lawrence, 2017." Canadian Wildlife Health Cooperative, Marine Animal Response Society, and Fisheries and Oceans Canada, 2018. https://publications.gc.ca/site/eng/9.850838/publication.html.

Examples

```
library(whalestrike)
shape <- whaleShape()
plot(shape$x, shape$y, asp = 1, type = "l")
polygon(shape$x, shape$y, col = "lightgray")
lw <- 13.7
Rmax <- 0.5 * lw * diff(range(shape$y))
mtext(sprintf("Max. radius %.2fm for %.1fm-long whale", Rmax, lw), side = 3)
```

`whaleSkinForce` *Skin force*

Description

The ship-whale separation is used to calculate the deformation of the skin. The parameters of the calculation are `parms$Ly` (impact area width, m), `parms$Lz` (impact area height, in m), `parms$Ealpha` (skin elastic modulus in Pa), `parms$alpha` (skin thickness in m), and `parms$theta` (skin bevel angle degrees, measured from a vector normal to undisturbed skin).

Usage

```
whaleSkinForce(xs, xw, parms)
```

Arguments

`xs`	Ship position (m).
`xw`	Whale position (m).
`parms`	A named list holding model parameters, created by `parameters()`.

Value

A list containing `force`, the normal force (N), along with `sigmay` and `sigmaz`, which are stresses (Pa) in the y (beam) and z (draft) directions.

Author(s)

Dan Kelley

References

See `whalestrike()` for a list of references.

See Also

Other functions relating to whale characteristics: `stressFromStrainFunction()`, `whaleCompressionForce()`, `whaleLengthFromMass()`, `whaleMassFromLength()`, `whaleWaterForce()`

Other functions relating to forces: `shipWaterForce()`, `stressFromStrainFunction()`, `whaleCompressionForce()`, `whaleWaterForce()`

`whaleWaterForce` *Whale Drag Force*

Description

Compute the retarding force of water on the whale, based on a drag law $(1/2) * rho * Cw * A * vw^2$ where `rho` is 1024 (kg/m^3), `Cw` is `parms$Cw` and `A` is `parms$Sw`.

Usage

```
whaleWaterForce(vw, parms)
```

Arguments

`vw`	Whale velocity (m/s).
`parms`	A named list holding model parameters, created by `parameters()`.

Value

Water drag force (N).

Author(s)

Dan Kelley

See Also

Other functions relating to whale characteristics: `stressFromStrainFunction()`, `whaleCompressionForce()`, `whaleLengthFromMass()`, `whaleMassFromLength()`, `whaleSkinForce()`

Other functions relating to forces: `shipWaterForce()`, `stressFromStrainFunction()`, `whaleCompressionForce()`, `whaleSkinForce()`

Index