

# Package ‘stagePop’

May 12, 2015

**Type** Package

**Title** Modelling the Population Dynamics of a Stage-Structured Species  
in Continuous Time

**Version** 1.1-1

**Date** 2015-05-12

**Author** Helen Kettle

**Maintainer** David Nutter <david.nutter@bioass.ac.uk>

**Description** Provides facilities to implement and run population models of  
stage-structured species...

**NeedsCompilation** no

**URL** <https://github.com/HelenKettle/StagePop>

**Suggests** R.rsp

**VignetteBuilder** R.rsp

**License** GPL-3 | file LICENSE

**Imports** deSolve,PBSddesolve

**Collate** 'SolverOptions.R' 'addNamesToList.R' 'checkICs.R'  
'checkSolution.R' 'deathFunction.R' 'reproductionFunction.R'  
'immigrationFunction.R' 'emigrationFunction.R'  
'durationFunction.R' 'developmentFunction.R'  
'defaultRateFunctions.R' 'derivDede.R' 'derivPBS.R'  
'genericPlot.R' 'initConditions.R' 'namingVariables.R'  
'plotStrains.R' 'popModel.R' 'rateFuncCheck.R' 'rateFunction.R'  
'runStagePopExample.R' 'stagePop-package.R' 'sumStrains.R'  
'vecDivideFunc.R' 'vecToMatList.R'

**Repository** CRAN

**Date/Publication** 2015-05-12 17:25:36

## R topics documented:

stagePop-package . . . . . 2

checkSolution . . . . .	3
deathFuncDefault . . . . .	3
defaultRateFunctions . . . . .	4
derivDede . . . . .	5
derivPBS . . . . .	5
develFuncDefault . . . . .	6
durationFuncDefault . . . . .	7
emigrationFuncDefault . . . . .	7
genericPlot . . . . .	8
immigrationFuncDefault . . . . .	9
popModel . . . . .	9
rateFuncCheck . . . . .	12
RateFunctions . . . . .	13
reproFuncDefault . . . . .	13
runStagePopExample . . . . .	14
SolverOptions . . . . .	14
sumStrains . . . . .	15
<b>Index</b>	<b>16</b>

---

stagePop-package	<i>Stage-Structured Population modelling</i>
------------------	--

---

## Description

stagePop can be used to model the dynamics and interactions of stage-structured populations (i.e. where the life cycle consists of distinct stages - e.g. eggs, juveniles and reproductive adults) in continuous time. It can be used to model predator-prey interactions, host-parasitoid interactions, resource competition and the effects of environmental change on stage-structured species. The model is based on the formulation described by RM Nisbet and WSC Gurney in 1983 in the paper 'The systematic formulation of population models for insects with dynamically varying instar duration. *Theoretical Population Biology*, 23, 114-135'. It uses delay differential equations which are solved using the R-packages 'deSolve' or 'PBSddesolve'.

## Author(s)

Helen Kettle

## References

To be done

---

checkSolution	<i>checkSolution</i>
---------------	----------------------

---

**Description**

Check whether any of the state variables are significantly negative and emit suitable warnings.

**Usage**

```
checkSolution(output, numSpecies, numStages, numStrains, ntol)
```

**Arguments**

output	Model output from <a href="#">popModel</a>
numSpecies	Number of species
numStages	Number of life stages
numStrains	Number of strains for each species
ntol	Negative tolerance value (i.e. a warning is produced if $\text{variable} < -(\text{ntol} * \max(\text{variable}))$ ). The larger ntol is, the larger the negative values that are tolerated.

**Value**

Nothing if there are no problems with the output, otherwise warnings are generated

**See Also**

[popModel](#)

---

deathFuncDefault	<i>Death Rate Function</i>
------------------	----------------------------

---

**Description**

Return the per capita rate of death (per unit time)

**Usage**

```
deathFuncDefault(stage, x, time, species, strain)
```

**Arguments**

stage	(integer). The numbered life stage for which to return a death rate.
x	Vector of state variables within the DDE solver. To access a variable use: x\$speciesName['stageName',strainNumber] e.g. for species 'Bacteria', stage 'reproductive', strain 2 use x\$Bacteria['reproductive',2] If there is only one stage and strain in species 'Food', for example, use x\$Food[1,1]
time	(scalar). The current time point in the DDE solver.
species	(integer). The numbered species for which to return a death rate.
strain	(integer). The numbered strain for which to return a death rate.

**Value**

per capita death rate (units of inverse time) for the strain, stage,species and time specified.

---

defaultRateFunctions *defaultRateFunctions*

---

**Description**

These default implementations will simply generate errors when run. To create implementations, please see the documentation linked to below.

**Usage**

```
defaultRateFunctions
```

**Format**

```
List of 6
$ reproFunc      :function (x, time, species, strain)
$ deathFunc      :function (stage, x, time, species, strain)
$ durationFunc   :function (stage, x, time, species, strain)
$ immigrationFunc: function (stage, x, time, species, strain)
$ develFunc      : NULL
$ emigrationFunc :function (stage, x, time, species, strain)
```

**Details**

The list should contain the following names, each mapped to a function of the correct signature.

- reproFunc
- deathFunc
- durationFunc
- immigrationFunc
- develFunc. Note that by default develFunc is NULL as it is not required for all simulation types

**See Also**[RateFunctions](#)

---

derivDede	<i>derivDede</i>
-----------	------------------

---

**Description**

Internal-use function which allows use of the DeSolve ODE solver. Called by [popModel](#)

**Usage**

```
derivDede(t, y, parms)
```

**Arguments**

t	Time step
y	state variables
parms	Additional parameters passed in from the original invoker of the solver

**Value**

the new model state.

**See Also**[derivPBS](#)

---

derivPBS	<i>derivPBS</i>
----------	-----------------

---

**Description**

Internal-use function which allows use of the PBSddesolv ODE solver. [popModel](#)

**Usage**

```
derivPBS(t, y, parms)
```

**Arguments**

t	Time step
y	state variables
parms	Additional parameters passed in from the original invoker of the solver

**Details**

note PBS does not allow lagged variables to be computed at the current timestep (hence the changes from the dede deriv function)

**Value**

the new model state.

**See Also**

[derivDede](#)

---

develFuncDefault	<i>Development Function</i>
------------------	-----------------------------

---

**Description**

Return the rate of development (per unit time)

**Usage**

```
develFuncDefault(stage, x, time, species, strain)
```

**Arguments**

stage	(integer). The numbered life stage for which to return a development rate.
x	Vector of state variables within the DDE solver. To access a variable use: x\$speciesName['stageName',strainNumber] e.g. for species 'Bacteria', stage 'reproductive', strain 2 use x\$Bacteria['reproductive',2] If there is only one stage and strain in species 'Food', for example, use x\$Food[1,1]
time	(scalar). The current time point in the DDE solver.
species	(integer). The numbered species for which to return a development rate.
strain	(integer). The numbered strain for which to return a development rate.

**Details**

This function will only be called when the timeDependDuration argument to [popModel](#) contains TRUE values, otherwise the development rate within a stage is irrelevant to the model.

**Value**

Development rate (units of inverse time) for the strain, stage, species and time specified.

---

durationFuncDefault *Life stage duration function*

---

### Description

Return the duration of the life stages (time units).

### Usage

durationFuncDefault(stage, x, time, species, strain)

### Arguments

stage	(integer). The numbered life stage for which to return a duration
x	Vector of state variables within the DDE solver. To access a particular variable use: x\$speciesName['stageName',strainNumber] e.g. for species 'Bacteria', stage 'reproductive', strain 2 use x\$Bacteria['reproductive',2] If there is only one stage and strain in species 'Food', for example, use x\$Food[1,1]
time	(scalar). The current time point in the DDE solver.
species	(integer). The numbered species for which to return a life stage duration.
strain	(integer). The numbered strain for which to return a life stage duration.

### Value

Duration of the life stage for the stage, species, strain and time specified.

---

emigrationFuncDefault *Emigration Function*

---

### Description

Return the per capita rate of Emigration (per unit time)

### Usage

emigrationFuncDefault(stage, x, time, species, strain)

**Arguments**

stage	(integer). The numbered life stage for which to return an emigration rate.
x	Vector of state variables within the DDE solver. To access a variable use: x\$speciesName['stageName',strainNumber] e.g. for species 'Bacteria', stage 'reproductive', strain 2 use x\$Bacteria['reproductive',2] If there is only one stage and strain in species 'Food', for example, use x\$Food[1,1]
time	(scalar). The current time point in the DDE solver.
species	(integer). The numbered species for which to return an emigration rate.
strain	(integer). The numbered strain for which to return an emigration rate.

**Value**

Emigration rate (per capita) for the strain, stage, species and time specified.

---

genericPlot

*Generic Plotting Function*

---

**Description**

Plot the state variables over time (sums over all strains if there are multiple strains in a species)

**Usage**

```
genericPlot(output, numSpecies, numStages, speciesNames, stageNames,
  saveFig = FALSE, figType = "eps", figName = "stagePopFig")
```

**Arguments**

output	Model output, from <a href="#">popModel</a>
numSpecies	Number of species in the model
numStages	Number of life stages
speciesNames	species names (vector of strings). Default is NULL.
stageNames	stage names (a list of vectors if there is more than one species).
saveFig	Choose to save the figure (TRUE or FALSE). Default is FALSE.
figType	Figure format can be 'eps', 'tiff' or 'png'. Default is 'eps'
figName	filepath to save figure to. Default is 'stagePopFig'

**See Also**

[popModel](#)



---

immigrationFuncDefault  
*Immigration Function*

---

**Description**

Return the rate of immigration (amount per unit time)

**Usage**

```
immigrationFuncDefault(stage, x, time, species, strain)
```

**Arguments**

stage	(integer). The numbered life stage for which to return an immigration rate.
x	Vector of state variables within the DDE solver. To access a variable use: x\$speciesName['stageName',strainNumber] e.g. for species 'Bacteria', stage 'reproductive', strain 2 use x\$Bacteria['reproductive',2] If there is only one stage and strain in species 'Food', for example, use x\$Food[1,1]
time	(scalar). The current time point in the DDE solver.
species	(integer). The numbered species for which to return an immigration rate.
strain	(integer). The numbered strain for which to return an immigration rate.

**Value**

Immigration rate (amount per unit time) for the strain, stage, species and time specified.

---

popModel                      *popModel*

---

**Description**

Run the core model.

**Usage**

```
popModel(numSpecies, numStages, numStrains = rep(1, numSpecies), timeVec,
speciesNames, stageNames, rateFunctions = defaultRateFunctions, ICs,
timeDependLoss = rep(TRUE, numSpecies), timeDependDuration = rep(FALSE,
numSpecies), solverOptions = list(), checkForNegs = TRUE, ntol = 0.01,
plotFigs = TRUE, saveFig = FALSE, figType = "eps",
figName = "stagePopFig", sumOverStrains = TRUE, plotStrainsFig = TRUE,
saveStrainsFig = FALSE, strainsFigType = "eps",
strainsFigName = "strainFig")
```

**Arguments**

numSpecies	Number of species (integer)
numStages	Number of life-stages in each species (vector)
numStrains	Number of strains in each species (vector). Default is 1.
timeVec	Vector of times the solution should be output out e.g. seq(1,10,0.1)
rateFunctions	A list of rate functions to use. See <a href="#">RateFunctions</a>
timeDependLoss	A vector specifying TRUE/FALSE for each species. If a species has any time dependent per capita death or emigration rates, the entry in the vector must be TRUE. The default is TRUE for each species.
timeDependDuration	A vector specifying TRUE/FALSE for each species. If a species has any time dependent stage durations the entry is TRUE for that species. The default is FALSE for each species.
ICs	is a list of matrices containing the initial conditions for every stage and strain of each species. These must be zero for all stages apart from the reproductive stage (usually the last stage). Each species has a matrix with the number of columns equal to the number of strains in that species and the number of rows equal to the number of stages in that species. E.g. for 2 species, the first with 2 strains and 3 stages, the second with 1 strain and 1 stage, then for zero starting conditions: ICs=list(matrix(0,ncol=2,nrow=3),matrix(0,ncol=1,nrow=1)). Due to the restrictions on initial conditions, it is recommended that more complicated initial conditions are defined through immigration rates in immigrationFunc
solverOptions	Options for the DDE solver. A list containing 'DDEsolver' (can be 'deSolve' or 'PBS'), 'tol' (max error tolerance for DDE solver), 'hbsize' (history buffer size), 'method' (method for DDE solver), 'atol' (absolute tolerance (deSolve only)) and 'dt' (maximum initial timestep (PBS only)). Default is solverOptions=list(DDEsolver='PBS',tol=1e-7,hbsize=1e3,method='lsoda',atol=1e-7,dt=0.1)
checkForNegs	If TRUE the function checkSolution is called and the solution for each variable, x, is checked for negative values that are greater in magnitude than ntol*max(x). If negative values occur then the solution is incorrect and either the problem is incorrectly specified or the tolerances in the DDE solver need to be smaller. The default is TRUE.
ntol	This controls the tolerance on the warning given for negative values when checkSolution is called. The default is 0.01 i.e. negative values whose magnitude is less than 1 percent of the max value of the variable are allowed.
plotFigs	If TRUE, results will be automatically plotted during the model run. The default is TRUE.
speciesNames	A vector of strings containing the name for each species.
stageNames	A list of n vectors (where n is the number of species) containing the names of each stage for each species.
saveFig	Choose to save the figure (TRUE or FALSE). Default is FALSE.
figType	Figure format can be 'eps', 'tiff' or 'png'. Default is 'eps'
figName	filepath to save figure to. Default is 'stagePopFig'

- sumOverStrains** If any of the species contain multiple strains then if this is TRUE the output is given as the sum over all the strains in the species. If this is FALSE then the time series for each strain will be in the output. Default is TRUE.
- plotStrainsFig** If any of the species contain multiple strains then if this is TRUE these will be plotted. Default is TRUE
- saveStrainsFig** If any of the species contain multiple strains then if this is TRUE the figures for the strains will be saved
- strainsFigType** If any of the species contain multiple strains and if saveStrainsFig=TRUE then this is used to choose the type of file the figure is saved as (choose from 'eps', 'pdf', 'png' and 'tiff'). Default is 'eps'.
- strainsFigName** If any of the species contain multiple strains and if saveStrainsFig=TRUE then this is used to choose the name of file the figure is saved as. Default is created by paste('strainFig',SpeciesName[i]).

### Details

The default solver options are:

```
list('DDEsolver'='deSolve', 'tol'=1e-7, 'hbsize'=1e3, 'method'=lsoda, 'atol'=1e-7, 'dt'=0.1)
```

but these may be changed by use of the solverOptions parameter. Please see [SolverOptions](#) for details of this parameter.

### Value

The model output is a matrix where rows are points in time and the columns are the state variables. These are named according to the species names and stage names supplied in inputs; the prefixes 'prob', 'dur' and 'dot' refer to the probability of survival through the stage, the duration of the stage and the rate of change of the variable. 'prob' type variables only appear if the per capita death (or emigration) rate is variable in time and 'dur' only appears if the stage duration is variable in time.

### Examples

```
rateFuncs=list(
  reproFunc=function(x,time,species,strain){
    v=10*x$flies['adults',1]*exp(-x$flies['adults',1]/100)
    return(max(v,0))
  },
  deathFunc=function(stage,x,time,species,strain){
    a=c(0.05,0.1,0.1); return(a[stage])
  },
  durationFunc=function(stage,x,time,species,strain){
    a=c(5,10); return(a[stage])
  },
  immigrationFunc=function(stage,x,time,species,strain){
    v=0
    if (stage==3 & time<1){v=100}; return(v)},
  emigrationFunc=function(stage,x,time,species,strain){return(0)}
)

modelOutput = popModel(
```

```

numSpecies=1,
numStages=3,
ICs=list(matrix(0,nrow=3,ncol=1)),
timeVec=seq(0,100,0.5),
timeDependLoss=FALSE,
timeDependDuration=FALSE,
rateFunctions=rateFuncs,
solverOptions=list(DDEsolver='PBS',tol=1e-4,hbsize=1e4,dt=0.01),
stageNames=list(c('eggs','larvae','adults')),
speciesNames=c('flies')
)

```

---

rateFuncCheck	<i>rateFuncCheck</i>
---------------	----------------------

---

### Description

check the user-defined rate functions exist and are in the correct format

### Usage

```
rateFuncCheck(x, numSpecies, numStages, numStrains, timeDependDuration,
rateFunctions, speciesNames, stageNames, startTime)
```

### Arguments

x	Vector of state variables within the DDE solver. To access a variable use: x\$speciesName['stageName',strainNumber]
numSpecies	Number of species
numStages	Number of life stages
numStrains	Number of strains for each species
timeDependDuration	vector of logicals defining whether the stage duration is time dependent for each species
rateFunctions	List of rate functions to check
speciesNames	Vector of species names
stageNames	Vector of species' stage names
startTime	First entry in timeVec vector

### Details

checks existence of deathFunc, reproFunc, durationFunc, immigrationFunc and develFunc if there is time dependent duration.

Also checks that output from rate functions is a single value

### Value

TRUE/FALSE - TRUE if rate functions are OK, FALSE if not.

---

RateFunctions      *Rate Functions*

---

### Description

The rate functions are user-defined functions which give the rates for each life stage.

### Details

These are passed into the `popModel` function via the `rateFunctions` argument.

The signatures for each rate function are defined in their respective documentation files

### See Also

[deathFunction](#) [durationFunction](#) [reproductionFunction](#) [developmentFunction](#) [immigrationFunction](#)  
[emigrationFunction](#)

---

reproFuncDefault      *Reproduction Function*

---

### Description

Return the rate of reproduction (amount per unit time)

### Usage

```
reproFuncDefault(x, time, species, strain)
```

### Arguments

<code>x</code>	Vector of state variables within the DDE solver. To access a variable use: <code>x\$speciesName['stageName',strainNumber]</code> e.g. for species 'Bacteria', stage 'reproductive', strain 2 use <code>x\$Bacteria['reproductive',2]</code> If there is only one stage and strain in species 'Food', for example, use <code>x\$Food[1,1]</code>
<code>time</code>	(scalar). The current time point in the DDE solver.
<code>species</code>	(integer). The numbered species for which to return a reproductive rate.
<code>strain</code>	(integer). The numbered strain for which to return a rate.

### Details

Unlike the other [RateFunctions](#) this rate function has no stage argument as by definition it only pertains to the first stage of life

### Value

Reproduction rate (amount per unit time) for the strain, species and time specified.

---

runStagePopExample	<i>runStagePopExample</i>
--------------------	---------------------------

---

### Description

This function is similar to the `demo()` function but requires less interaction. It is used to run the canned examples from the `stagePop` package.

### Usage

```
runStagePopExample(name = NULL)
```

### Arguments

name	Name of the example to run. If Name is NULL the list of examples will be printed.
------	---

---

SolverOptions	<i>SolverOptions</i>
---------------	----------------------

---

### Description

Documentation for the `SolverOptions` parameter

### Details

`SolverOptions` (optional) is a list containing: `'DDEsolver'`, `'tol'`, `'hbsize'`, `'method'` and `'atol'`. `DDEsolver` equal to `'deSolve'` or `'PBS'` sets the R package used to solve the DDEs. The `'tol'` option sets the relative tolerances and `'hbsize'` sets the size of the history buffer. The remaining two items, `'method'` and `'atol'` set the numerical integration scheme and the absolute tolerance if `'DDEsolver'='deSolve'` (`PBS` does not have these options). If `solverOptions` is not specified at all, or if only some of the options are specified, the default values: `list('DDEsolver'='deSolve', 'tol'=1e-7, 'hbsize'=1e3, 'method'='lsoda', 'atol'=1e-7, 'dt'=0.1)` will be used.

---

sumStrains	<i>sumStrains</i>
------------	-------------------

---

**Description**

sumStrains

**Usage**

```
sumStrains(namedOutput, numSpecies, numStages, numStrains, speciesNames,
           stageNames, timeDependLoss, timeDependDuration)
```

**Arguments**

namedOutput	The output matrix from popModel
numSpecies	Number of species (integer)
numStages	Number of life-stages in each species (vector)
numStrains	Number of strains in each species (vector)
speciesNames	A vector of strings containing the name for each species. This is only used for plotting purposes.
stageNames	A list of n vectors (where n is the number of species) containing the names of each stage for each species.
timeDependLoss	A vector specifying TRUE/FALSE for each species. If a species has any time dependent per capita death rates (e.g. density dependent death rates) this is TRUE.
timeDependDuration	A vector specifying TRUE/FALSE for each species. If a species has any time dependent stage durations it is TRUE for that species.

# Index

- \*Topic **datasets**
  - defaultRateFunctions, 4
- \*Topic **package**
  - stagePop-package, 2
- \*Topic **programming**
  - stagePop-package, 2
  
- checkSolution, 3
  
- deathFunc (deathFuncDefault), 3
- deathFuncDefault, 3
- deathFunction, 13
- deathFunction (deathFuncDefault), 3
- defaultRateFunctions, 4
- derivDede, 5, 6
- derivPBS, 5, 5
- develFunc (develFuncDefault), 6
- develFuncDefault, 6
- developmentFunction, 13
- developmentFunction (develFuncDefault), 6
- durationFunc (durationFuncDefault), 7
- durationFuncDefault, 7
- durationFunction, 13
- durationFunction (durationFuncDefault), 7
  
- EmigrationFunc (emigrationFuncDefault), 7
- emigrationFuncDefault, 7
- EmigrationFunction
  - (emigrationFuncDefault), 7
- emigrationFunction, 13
- emigrationFunction
  - (emigrationFuncDefault), 7
  
- genericPlot, 8
  
- immigrationFunc
  - (immigrationFuncDefault), 9
- immigrationFuncDefault, 9
  
- immigrationFunction, 13
- immigrationFunction
  - (immigrationFuncDefault), 9
  
- popModel, 3, 5, 6, 8, 9, 13
  
- rateFuncCheck, 12
- RateFunctions, 5, 10, 13, 13
- reproductionFunction, 13
- reproductionFunction
  - (reproFuncDefault), 13
- reproFunc (reproFuncDefault), 13
- reproFuncDefault, 13
- runStagePopExample, 14
  
- SolverOptions, 11, 14
- stagePop (stagePop-package), 2
- stagePop-package, 2
- sumStrains, 15