# Package ‘HYRISK’

April 4, 2017

**Type** Package

**Title** Hybrid Methods for Addressing Uncertainty in RISK Assessments

**Version** 1.2

**Date** 2017-04-04

**Author** Jeremy Rohmer, Jean-Charles Manceau, Dominique Guyonnet, Faiza Boulahya

**Maintainer** Jeremy Rohmer <j.rohmer@brgm.fr>

**Description** Methods for addressing uncertainty in risk assessments using hybrid representations of uncertainty (probability distributions, fuzzy numbers, intervals, probability distributions with imprecise parameters). The uncertainty propagation procedure combines random sampling using Monte Carlo method with fuzzy interval analysis of Baudrit et al. (2007) <doi:10.1109/TFUZZ.2006.876720>. The sensitivity analysis is based on the pinching method of Ferson and Tucker (2006) <doi:10.1016/j.ress.2005.11.052>.

**License** GPL-3

**Depends** R (>= 3.2.0)

**Imports** datasets,utils,grDevices,graphics,stats,sets, pbapply, reliaR, kerdiest, triangle, rgenoud

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2017-04-04 15:38:45 UTC

## R topics documented:

---

`HYRISK-package` *Hybrid Methods For Addressing Uncertainty In RISK Assessments*

---

## Description

Methods for addressing uncertainty in risk assessments using hybrid representations of uncertainty (probability distributions, fuzzy numbers, intervals, probability distributions with imprecise parameters). The uncertainty propagation procedure combines random sampling using Monte Carlo method with fuzzy interval analysis of Baudrit et al. (2007) <doi:10.1109/TFUZZ.2006.876720>. The sensitivity analysis is based on the pinching method of Ferson and Tucker (2006) <doi:10.1016/j.ress.2005.11.052>.

## Details

This package provides tools for uncertainty analysis:

- Create input uncertain variables represented by an interval, a possibility distribution (trapezoidal or triangular), a probability distribution (normal, lognormal, beta, triangle, Gumbel or user-defined), or an imprecise probability distribution.
- Perform joint uncertainty propagation using IRS of Baudrit et al. (2007) or Hybrid algorithm described by Baudrit et al. (2006).
- Perform uncertainty propagation when the random variables are represented by imprecise probabilities, i.e. probability distribution with ill-known parameters (Pedroni et al., 2013).
- Summarize the uncertan results in the form of a pair of lower and upper cumulative distribution functions CDFs.
- Summarize the uncertainty in the form of an interval of exceedance probabilities, an interval of quantiles, or a global indicator corresponding to the area between the lower and upper CDFs.
- Perform sensitivity analysis using a pinching approach (Ferson and Tucker, 2006).

## Author(s)

Author: Jeremy Rohmer, Jean-Charles Manceau, Faiza Boulahya Maintainer: Jeremy Rohmer <j.rohmer@brgm.fr>

## References

- Baudrit, C., Dubois, D., & Guyonnet, D. 2006. Joint propagation and exploitation of probabilistic and possibilistic information in risk assessment. IEEE transactions on fuzzy systems, 14(5), 593-608.
- Baudrit, C., Guyonnet, D., Dubois, D. 2007. Joint propagation of variability and partial ignorance in a groundwater risk assessment. Journal of Contaminant Hydrology, 93: 72-84.

- Ferson, S., & Tucker, W. T. (2006). Sensitivity analysis using probability bounding. Reliability Engineering & System Safety, 91(10), 1435-1442.
- Pedroni, N., Zio, E., Ferrario, E., Pasanisi, A., & Couplet, M. 2013. Hierarchical propagation of probabilistic and non-probabilistic uncertainty in the parameters of a risk model. Computers & Structures, 126, 199-213.


---

`CREATE_DISTR` *Function to assign the corresponding distribution (probability or possibility) to each uncertain input.*

---

## Description

Function to assign the corresponding distribution (probability or possibility) to each uncertain input based on the input definition using CREATE_INPUT()

## Usage

```
CREATE_DISTR(input, DISCR=1000)
```

## Arguments

| | |
|---|---|
| `input` | List of inputs derived from the *CREATE_INPUT()* function. |
| `DISCR` | Number of discretisations to represent the possibility distribution. By default, it is set at 1000. |

## Details

- Details of the theory and example 1 in Dubois & Guyonnet (2011), available at: https://hal-brgm.archives-ouvertes.fr/file/index/docid/578821/filename/Uncertainties_RA_09_l_dg.pdf
- Details on the representation via imprecise probability distributions and example 2 in Sch\"obi & Sudret (2016), available at: https://arxiv.org/pdf/1608.05565.pdf

## Value

List of inputs updated with additional arguments

- If input$type="proba" or input$type="impr proba", new arguments *qfun* and *rfun* corresponding to the quantile and random sampling functions.
- If input$type="possi", new argument *fuzzy* corresponding to the output provided by *fuzzy_trapezoid_gset()* or *fuzzy_triangular_gset()* of the package *sets*.

## See Also

CREATE_INPUT

## Examples

```
################################################
#### EXAMPLE 1 of Dubois & Guyonnet (2011)
#### Probability and Possibility distributions
################################################
ninput<-5 #Number of input parameters
input<-vector(mode="list", length=ninput) # Initialisation

input[[1]]=CREATE_INPUT(
name="UER",
type="possi",
distr="triangle",
param=c(2.e-2, 5.7e-2, 1.e-1),
monoton="incr"
)
input[[2]]=CREATE_INPUT(
name="EF",
type="possi",
distr="triangle",
param=c(200,250,350),
monoton="incr"
)
input[[3]]=CREATE_INPUT(
name="I",
type="possi",
distr="triangle",
param=c(1,1.5,2.5),
monoton="incr"
)
input[[4]]=CREATE_INPUT(
name="C",
type="proba",
distr="triangle",
param=c(5e-3,20e-3,10e-3)
)
input[[5]]=CREATE_INPUT(
name="ED",
type="proba",
distr="triangle",
param=c(10,50,30)
)

####CREATION OF THE DISTRIBUTIONS ASSOCIATED TO THE PARAMETERS
input=CREATE_DISTR(input)

####PLOT INPUTS
PLOT_INPUT(input)

################################################
#### EXAMPLE 2 of Sch\"obi & Sudret (2016)
#### Imprecise Probability distributions
```

```
#################################################

ninput<-6 #Number of input parameters
input<-vector(mode="list", length=ninput) # Initialisation

# Imprecise normal probability
# whose parameters are described by the 3rd and 5th parameters
input[[1]]=CREATE_INPUT(
name="A",
type="impr proba",
distr="normal",
param=c(3,5),
monoton="dunno"
)

# Imprecise normal probability
# whose parameters are described by the 4th and 6th parameters
input[[2]]=CREATE_INPUT(
name="B",
type="impr proba",
distr="normal",
param=c(4,6),
monoton="dunno"
)

# imprecise paramters of afore-described probability distribution
# mean of input number 1 as an interval
input[[3]]=CREATE_INPUT(
name="mu_A",
type="possi",
distr="interval",
param=c(-0.5,0.5)
)

# mean of input number 2 as an interval
input[[4]]=CREATE_INPUT(
name="mu_B",
type="possi",
distr="interval",
param=c(-0.5,0.5)
)

# standard deviation of input number 1  as an interval
input[[5]]=CREATE_INPUT(
name="s_A",
type="possi",
distr="interval",
param=c(0.7,1)
)

# standard deviation of input number 2  as an interval
input[[6]]=CREATE_INPUT(
name="s_B",
```

```
type="possi",
distr="interval",
param=c(0.7,1)
)

####CREATION OF THE DISTRIBUTIONS ASSOCIATED TO THE PARAMETERS
input=CREATE_DISTR(input)

####PLOT INPUTS
PLOT_INPUT(input)
```

---

`CREATE_INPUT` *Function to define the input variables (imprecise, random or fixed).*

---

## Description

Function to define the input variables (imprecise, random or fixed). Five types are accounted for:

- Probability distribution (either pre-defined (Normal, log-Normal, triangular, Beta, Gumbel) or user-defined).
- Possibility distribution (either trapezoidal or triangular).
- Intervals.
- Fixed scalar value.
- Imprecise probability distributions (Normal, log-Normal, triangular, Beta, Gumbel) with parameters represented by by intervals, possibility distributions, fixed scalar values, or probability distributions.

## Usage

```
CREATE_INPUT(name, type, distr = NULL, param,
monoton = "dunno", quser = NULL, ruser = NULL)
```

## Arguments

`name` String of the name of the input variable.

`type` String to specify the type of input variable:

- "proba": probability distribution.
- "possi": possibility distribution.
- "impr proba": imprecise probability distribution
- "fixed": fixed scalar value.

`distr` String to specify the type of distribution:

- If type="proba" or type="impr proba", distr should be: "normal", "lognormal", "triangle", "beta", "gumbel","user".
- If type="possi", distr should be: "trapeze", "triangle", "interval".

`param` Vector of parameter values. This depends on the choice of *distr* and *type*.

- If type="proba" and distr="normal", param should be a vector of two values. For example c(0,1) corresponds to a normal distribution with mean=0 and standard deviation of 1.
- If type="proba" and distr="lognormal", param should be a vector of two values. For example c(0,1) corresponds to a lognormal distribution with meanlog=0 and standard deviation of sdlog=1.
- If type="proba" and distr="triangle", param should be a vector of three values. For example c(0,2,1) corresponds to a traingle distribution with apex=1 and support[0,2]. See help from package *triangle*.
- If type="proba" and distr="beta", param should be a vector of two values. For example c(1,1) corresponds to a beta distribution with parameters shape1 and shape2 equal to 1 (see help of qbeta in stats package).
- If type="proba" and distr="gumbel", param should be a vector of two values. For example c(0,1) corresponds to a gumbel distribution with location=0 and scale parameter=1.
- If type="possi" and distr="trapeze", param should be a vector of four values. For example c(0,1,2,3) corresponds to a possibility distribution with core=[1,2] and support=[0,3].
- If type="possi" and distr="triangle", param should be a vector of three values. For example c(0,1,2) corresponds to a possibility distribution with core=1 and support=[0,2].
- If type="possi" and distr="interval", param should be a vector of two values. For example c(0,1) corresponds to the interval [0,1].
- If type="impr proba", param should be a vector of integers; each integer points to the rank of the input, which represents the uncertainty on the corresponding parameter of chosen distribution For example, if type="impr proba" and distr="normal", c(2,3) corresponds to an imprecise normal distribution, whose imprecise mean is input[[2]] and standard deviation is input[[3]].

`monoton` String to specify the monotony of the model function regarding the input variable.

- "decr" for decreasing.
- "incr" for increasing.
- "dunno" for unknown monotony or known no-monotony.

`quser` If distr="user", this string specifies the quantile function of a probability distribution non listed in the pre-defined ones. The vector of *param* should be updated according to this law.

`ruser` If distr="user", string to specify the random sampling function of a probability distribution non listed in the pre-defined ones. The vector of *param* should be updated according to this law.

## Details

Details of the theory and the example in Dubois & Guyonnet (2011) Available at: https://hal-brgm.archives-ouvertes.fr/file/index/docid/578821/filename/Uncertainties_RA_09_l_dg.pdf

### Value

list with the afore-described arguments.

### See Also

CREATE_DISTR PLOT_INPUT

### Examples

```
#################################################
#### EXAMPLE 1 of Dubois & Guyonnet (2011)
#### Probability and Possibility distributions
#################################################

ninput<-5 #Number of input parameters
input<-vector(mode="list", length=ninput) # Initialisation

input[[1]]=CREATE_INPUT(
name="UER",
type="possi",
distr="triangle",
param=c(2.e-2, 5.7e-2, 1.e-1),
monoton="incr"
)
input[[2]]=CREATE_INPUT(
name="EF",
type="possi",
distr="triangle",
param=c(200,250,350),
monoton="incr"
)
input[[3]]=CREATE_INPUT(
name="I",
type="possi",
distr="triangle",
param=c(1,1.5,2.5),
monoton="incr"
)
input[[4]]=CREATE_INPUT(
name="C",
type="proba",
distr="triangle",
param=c(5e-3,20e-3,10e-3)
)
input[[5]]=CREATE_INPUT(
name="ED",
type="proba",
distr="triangle",
param=c(10,50,30)
)
```

```
##################################################
#### EXAMPLE 2 of Sch\"obi & Sudret (2016)
#### Imprecise Probability distributions
##################################################

ninput<-6 #Number of input parameters
input<-vector(mode="list", length=ninput) # Initialisation

# Imprecise normal probability
# whose parameters are described by the 3rd and 5th parameters
input[[1]]=CREATE_INPUT(
name="A",
type="impr proba",
distr="normal",
param=c(3,5),
monoton="dunno"
)

# Imprecise normal probability
# whose parameters are described by the 4th and 6th parameters
input[[2]]=CREATE_INPUT(
name="B",
type="impr proba",
distr="normal",
param=c(4,6),
monoton="dunno"
)

# imprecise paramters of afore-described probability distribution
# mean of input number 1 as an interval
input[[3]]=CREATE_INPUT(
name="mu_A",
type="possi",
distr="interval",
param=c(-0.5,0.5)
)

# mean of input number 2 as an interval
input[[4]]=CREATE_INPUT(
name="mu_B",
type="possi",
distr="interval",
param=c(-0.5,0.5)
)

# standard deviation of input number 1  as an interval
input[[5]]=CREATE_INPUT(
name="s_A",
type="possi",
distr="interval",
param=c(0.7,1)
)
```

```
# standard deviation of input number 2  as an interval
input[[6]]=CREATE_INPUT(
name="s_B",
type="possi",
distr="interval",
param=c(0.7,1)
)
```

---

`PINCHING_fun` *Function to perform pinching.*

---

## Description

Function to pinch a imprecise variable to a fixed value following Ferson and Tucker (2006) and perform the propagation. Note that it only handles imprecise parametes. In this case of an imprecise probability distributions, only the imprecise parameters are handled.

## Usage

```
PINCHING_fun(which, value, N, input, FUN,
     choice_opt="L-BFGS-B", param_opt=NULL,
     corr = 0.01, NL = 10, mode = "IRS")
```

## Arguments

| | |
|---|---|
| `which` | Integer to specify the rank of the input variable as specifed in *CREATE_INPUT()*, i.e. parameter 1, 2,..., etc. |
| `value` | Scalar value to which the imprecise variable is pinched. |
| `N` | Integer corresponding to the number of random samples. |
| `input` | List of inputs as provided by *CREATE_INPUT()*. |
| `FUN` | Model assessment function. |
| `choice_opt` | Option of constrainted optimization algorithm, see `PROPAG`. |
| `param_opt` | Parameters necessary for conducting the optimization algorithm, see `PROPAG`. |
| `corr` | Tolerance to avoid empty alpha-cuts. By default, *corr*=0.01. |
| `NL` | Integer to specify the number of alpha-cuts needed for hybrid propagation described by Baudrit et al. (2006). By default, *NL*=10. |
| `mode` | String to specify the mode of propagation: "IRS" (Baudrit et al. 2007) or "HYBRID" (Baudrit et al. 2006), see `PROPAG`. |

### References


- Baudrit, C., Dubois, D., & Guyonnet, D. 2006. Joint propagation and exploitation of probabilistic and possibilistic information in risk assessment. IEEE transactions on fuzzy systems, 14(5), 593-608.
- Baudrit, C., Guyonnet, D., Dubois, D. 2007. Joint propagation of variability and partial ignorance in a groundwater risk assessment. Journal of Contaminant Hydrology, 93: 72-84.
- Ferson, S., & Tucker, W. T. (2006). Sensitivity analysis using probability bounding. Reliability Engineering & System Safety, 91(10), 1435-1442.


### See Also

SENSI_PINCHING PROPAG

### Examples

```
#See example described for \emph{SENSI_PINCHING}.
```

---

PLOT_CDF *Plot function depicting the lower and upper CDFs*

---

### Description

Create a plot summarizing the uncertainty propagation in the form of a pair of lower and upper Cumulative Distribution Function CDFs.

### Usage

```
PLOT_CDF(Z0, new = TRUE, color1 = 1, color2 = 2, ...)
```

### Arguments

| | |
|---|---|
| Z0 | Output of the uncertainty propagation function *PROPAG()*. |
| new | Logical (by default TRUE) to create a new plot or to add CDFs to an existing plot. By default *new*=TRUE. |
| color1 | Color of the upper CDF. By default *color1*=1. |
| color2 | Color of the lower CDF. By default *color2*=2. |
| ... | Further arguments usually passed to the plot function (e.g. *xlab*, *ylab*, *main*, etc.) |

### See Also

PROPAG

## PLOT_INPUT — *Function for plotting the inputs.*

### Description

Plot the input variables. Regarding imprecise probability distribution, it plots the p-box of the imprecise probability distribution itself as well as its parameters. Note that plotting the imprecise probability distribution requires solving an optimisation problem using the "L-BFGS-B" algorithm by assuming independdance of dependence among the alpha-cuts of imprecise parameters.

### Usage

```
PLOT_INPUT(input,N=1000,mode="IND")
```

### Arguments

| | |
|---|---|
| `input` | List of inputs with arguments as specified in *CREATE_INPUT()*. |
| `N` | Integer corresponding to the number of random samples (to be used for plotting imprecise probability distributions only). By default N=1000. |
| `mode` | String to specify the dependence or the independence among the imprecise parameters (to be used for plotting imprecise probability distributions only): "IND" (independence) or "DEP" (dependence), see `PROPAG`. |

### Value

It provides a plot of the input variables.

### See Also

`CREATE_INPUT`

## PROBA_INTERVAL — *Interval of probability of exceedance.*

### Description

Function for summarizing the uncertainty propagation's results in the form of an interval of probability of exceedance for a given threshold.

### Usage

```
PROBA_INTERVAL(Z0, threshold)
```

### Arguments

| | |
|---|---|
| Z0 | Output of the uncertainty propagation function *PROPAG()*. |
| threshold | Scalar value specifying the threshold for which the probability of exceedance should be calculated. |

### Details

Details of the theory and the example in Dubois & Guyonnet (2011) Available at: https://hal-brgm.archives-ouvertes.fr/file/index/docid/578821/filename/Uncertainties_RA_09_l_dg.pdf

### Value

List with arguments *Plow* and *Pupp* corresponding to lower and upper probability values.

### See Also

PROPAG QUAN_INTERVAL UNCERTAINTY

### Examples

```
#See example described for \emph{UNCERTAINTY}.
```

---

## PROPAG — *Function for conducting joint propagation of uncertainty*

---

### Description

Function for conducting joint propagation of probability, imprecise probability and possibilisty distributions (or intervals) using IRS (Baudrit et al., 2006) or hybrid method (Guyonnet et al., 2003).

### Usage

```
PROPAG(N, input, FUN, choice_opt="L-BFGS-B", param_opt = NULL,
       mode = "IRS", corr = 0.01, NL = 10)
```

### Arguments

| | |
|---|---|
| N | Integer corresponding to the number of random samples. |
| input | List of inputs as provided by the function *CREATE_INPUT()*. |
| FUN | Model assessment function. |
| choice_opt | Option for the constrainted optimization algorithm:<br>• "L-BFGS-B": Limited-memory BFGS based on *optimr* package.<br>• "L-BFGS-B_MULTI": Limited-memory BFGS with multiple starts. |

- "GENOUD": genetic algorithm plus derivative optimizer based on *rgenoud* package, see Mebane & Sekhon (2011).

`param_opt` Parameters needed by the optimization algorithm:

- If "L-BFGS-B": param_opt=NULL.
- If "L-BFGS-B_MULTI": *param_opt* specifies the number of multi starts (e.g. 10).
- If "GENOUD": *param_opt* specifies vector of paramters (population size, maxixmum of generations, solution tolerance), see Mebane & Sekhon (2011).

`mode` Type of hybrid uncertainty propagation:

- "IRS" Independent Random Sampling of Baudrit et al. (2007).
- "HYBRID" Hybrid propagation described by Baudrit et al. (2006).

`corr` Tolerance to avoid empty alpha-cuts. By default, *corr*=0.01.

`NL` Integer to specify the number of alpha-cuts needed for hybrid propagation described by Baudrit et al. (2006). By default, *NL*=10.

## Details

- Details of the theory and example 1 in Dubois & Guyonnet (2011), available at: `https://hal-brgm.archives-ouvertes.fr/file/index/docid/578821/filename/Uncertainties_RA_09_l_dg.pdf`
- Details on the representation via imprecise probability distributions in Pedroni et al. (2013).
- Details on example 2 in Sch\"obi & Sudret (2016), available at: `https://arxiv.org/pdf/1608.05565.pdf`

## Value

Matrix Z0 of 2 rows and *N* columns (for IRS mode) or *N*x*NL* (for HYBRID mode). This corresponds to the set of random intervals (row Number 1: lower bound; row Number 2: upper bound), which can be summarized in different forms, see Baudrit et al. (2006).

## References


- Baudrit, C., Dubois, D., & Guyonnet, D. 2006. Joint propagation and exploitation of probabilistic and possibilistic information in risk assessment. IEEE transactions on fuzzy systems, 14(5), 593-608.
- Baudrit, C., Guyonnet, D., Dubois, D. 2007. Joint propagation of variability and partial ignorance in a groundwater risk assessment. Journal of Contaminant Hydrology, 93: 72-84.
- Mebane, W., Jr. and Sekhon, J. S. 2011. Genetic Optimization Using Derivatives: The rgenoud package for R. Journal of Statistical Software, 42(11): 1-26.
- Pedroni, N., Zio, E., Ferrario, E., Pasanisi, A., & Couplet, M. 2013. Hierarchical propagation of probabilistic and non-probabilistic uncertainty in the parameters of a risk model. Computers & Structures, 126, 199-213.
- Sch\"obi, R., Sudret, B. 2016. Uncertainty propagation of p-boxes using sparse polynomial chaos expansions, https://arxiv.org/pdf/1608.05565.pdf

## See Also

CREATE_INPUT CREATE_DISTR PLOT_CDF

## Examples

```
## Not run:

##################################################
#### EXAMPLE 1 of Dubois & Guyonnet (2011)
#### Probability and Possibility distributions
##################################################

#### Model function
FUN<-function(X){
UER=X[1]
EF=X[2]
I=X[3]
C=X[4]
ED=X[5]
return(UER*I*C*EF*ED/(70*70*365))
}

ninput<-5 #Number of input parameters
input<-vector(mode="list", length=ninput) # Initialisation

input[[1]]=CREATE_INPUT(
name="UER",
type="possi",
distr="triangle",
param=c(2.e-2, 5.7e-2, 1.e-1),
monoton="incr"
)
input[[2]]=CREATE_INPUT(
name="EF",
type="possi",
distr="triangle",
param=c(200,250,350),
monoton="incr"
)
input[[3]]=CREATE_INPUT(
name="I",
type="possi",
distr="triangle",
param=c(1,1.5,2.5),
monoton="incr"
)
input[[4]]=CREATE_INPUT(
name="C",
type="proba",
distr="triangle",
param=c(5e-3,20e-3,10e-3)
)
```

```
input[[5]]=CREATE_INPUT(
name="ED",
type="proba",
distr="triangle",
param=c(10,50,30)
)

####CREATION OF THE DISTRIBUTIONS ASSOCIATED TO THE PARAMETERS
input=CREATE_DISTR(input)

####VISU INPUT
PLOT_INPUT(input)

#################################################
#### PROPAGATION

#OPTIMZATION CHOICES
choice_opt=NULL #no optimization needed
param_opt=NULL

#PROPAGATION RUN
Z0_IRS<-PROPAG(N=1000,input,FUN,choice_opt,param_opt,mode="IRS")
Z0_HYBRID<-PROPAG(N=250,input,FUN,choice_opt,param_opt,mode="HYBRID")

#################################################
#### POST-PROCESSING

# VISU - PROPAGATION
PLOT_CDF(Z0_IRS,xlab="Z",ylab="CDF",main="EX 1",lwd=1.5)
PLOT_CDF(Z0_HYBRID,new=FALSE,color1=3,color2=4,lwd=1.5)

#################################################
#### EXAMPLE 2 of Schobi & Sudret (2016)
#### Imprecise Probability distributions
#################################################

#### Model function
FUN<-function(X){
A=X[1]
B=X[2]
return(100*(B-A^2)^2+(1-A)^2)
}

ninput<-6 #Number of input parameters
input<-vector(mode="list", length=ninput) # Initialisation

# Imprecise normal probability
# whose parameters are described by the 3rd and 5th parameters
input[[1]]=CREATE_INPUT(
name="A",
type="impr proba",
distr="normal",
param=c(3,5),
```

```
monoton="dunno"
)

# Imprecise normal probability
# whose parameters are described by the 4th and 6th parameters
input[[2]]=CREATE_INPUT(
name="B",
type="impr proba",
distr="normal",
param=c(4,6),
monoton="dunno"
)

# imprecise paramters of afore-described probability distribution
# mean of input number 1 as an interval
input[[3]]=CREATE_INPUT(
name="mu_A",
type="possi",
distr="interval",
param=c(-0.5,0.5)
)

# mean of input number 2 as an interval
input[[4]]=CREATE_INPUT(
name="mu_B",
type="possi",
distr="interval",
param=c(-0.5,0.5)
)

# standard deviation of input number 1  as an interval
input[[5]]=CREATE_INPUT(
name="s_A",
type="possi",
distr="interval",
param=c(0.7,1)
)

# standard deviation of input number 2  as an interval
input[[6]]=CREATE_INPUT(
name="s_B",
type="possi",
distr="interval",
param=c(0.7,1)
)


####CREATION OF THE DISTRIBUTIONS ASSOCIATED TO THE PARAMETERS
input=CREATE_DISTR(input)

####VISU INPUT (needs propagation parameters to plot impr proba distributions)
PLOT_INPUT(input)
```

```
##################################################
### PROPAGATION
# OPTIMZATION CHOICES (could take some time)
choice_opt="GENOUD"
param_opt=c(50,3,1.e-1)

#PROPAGATION RUN
Z0_IRS<-PROPAG(N=1000,input,FUN,choice_opt,param_opt,mode="IRS")

##################################################
### VISU - PROPAGATION
PLOT_CDF(Z0_IRS,xlab="Z",ylab="CDF",main="EX 2",lwd=1.5)


## End(Not run)
```

---

QUAN_INTERVAL *Interval of quantiles at a given level.*

---

## Description

function for summarizing the uncertainty propagation's results in the form of an interval of quantiles at a given level. By default, the median value is provided (level=0.5).

## Usage

```
QUAN_INTERVAL(Z0, level = 0.5)
```

## Arguments

| | |
|---|---|
| Z0 | Output of the uncertainty propagation function PROPAG() |
| level | Value of the quantile level (e.g. 0.80 corresponds to the 80% quantile). |

## Details

Details of the theory and the example in Dubois & Guyonnet (2011) Available at: https://hal-brgm.archives-ouvertes.fr/file/index/docid/578821/filename/Uncertainties_RA_09_l_dg.pdf

## Value

List with arguments *Qlow* and *Qupp* corresponding to lower and upper quantile values.

## See Also

PROPAG PROBA_INTERVAL UNCERTAINTY

## Examples

```
#See example described for \emph{UNCERTAINTY}.
```

---

SENSI_PINCHING *Sensitivity analysis using the pinching approach.*

---

## Description

Function for conducting a sensitivity analysis using the pinching approach of Ferson and Tucker (2006).

## Usage

```
SENSI_PINCHING(Z0, Z0p, mode = "global",
       threshold = NULL, level = NULL, disc=0.01)
```

## Arguments

| | |
|---|---|
| `Z0` | Output of the uncertainty propagation function *PROPAG()*. |
| `Z0p` | Output of the pinching function *PINCHING_fun()*. |
| `mode` | String to specify the mode to represent the epistemic uncertainty:<br>• If "proba": interval of exceedance probability given the threshold.<br>• If "quantile": interval of quantile given the level.<br>• If "global": global indicator of uncertainty corresponding to the area between the upper and lower CDFs. |
| `threshold` | Threshold value to compute the interval of exceedance probabilities. |
| `level` | Level value to compute the interval of quantiles. |
| `disc` | discretisation value to compute the global indicator |

## Value

A number between 0 and 100

## References

Ferson, S., & Tucker, W. T. (2006). Sensitivity analysis using probability bounding. Reliability Engineering & System Safety, 91(10), 1435-1442.

## See Also

PINCHING_fun

## Examples

```
## Not run:
#################################################
#### EXAMPLE 1 of Dubois & Guyonnet (2011)
#### Probability and Possibility distributions
#################################################

#### Model function
FUN<-function(X){
UER=X[1]
EF=X[2]
I=X[3]
C=X[4]
ED=X[5]
return(UER*I*C*EF*ED/(70*70*365))
}

ninput<-5 #Number of input parameters
input<-vector(mode="list", length=ninput) # Initialisation

input[[1]]=CREATE_INPUT(
name="UER",
type="possi",
distr="triangle",
param=c(2.e-2, 5.7e-2, 1.e-1),
monoton="incr"
)
input[[2]]=CREATE_INPUT(
name="EF",
type="possi",
distr="triangle",
param=c(200,250,350),
monoton="incr"
)
input[[3]]=CREATE_INPUT(
name="I",
type="possi",
distr="triangle",
param=c(1,1.5,2.5),
monoton="incr"
)
input[[4]]=CREATE_INPUT(
name="C",
type="proba",
distr="triangle",
param=c(5e-3,20e-3,10e-3)
)
input[[5]]=CREATE_INPUT(
name="ED",
type="proba",
distr="triangle",
param=c(10,50,30)
```

```
)

####CREATION OF THE DISTRIBUTIONS ASSOCIATED TO THE PARAMETERS
input=CREATE_DISTR(input)

####VISU INPUT
PLOT_INPUT(input)

##################################################
#### PROPAGATION

#OPTIMZATION CHOICES
choice_opt=NULL #no optimization needed
param_opt=NULL

#PROPAGATION RUN
Z0_IRS<-PROPAG(N=1000,input,FUN,choice_opt,param_opt,mode="IRS")

##################################################
#### PINCHING
Z0p<-PINCHING_fun(
which=1,##first input variable
value=5.7e-2, ##pinched at the scalar value of 5.7e-2
N=1000,
input,
FUN,
choice_opt,
param_opt,
mode="IRS"
)

# VISU - PROPAGATION
PLOT_CDF(Z0_IRS,xlab="Z",ylab="CDF",main="EX PINCHING",lwd=1.5)
PLOT_CDF(Z0p,color1=3,color2=4,new=FALSE,lwd=1.5)

## quantile mode
sensi.quan<-SENSI_PINCHING(Z0_IRS,Z0p,mode="quantile",level=0.75)
print(paste("Quantile-based sensitivity measure: ",sensi.quan,sep=""))

## proba mode
sensi.proba<-SENSI_PINCHING(Z0_IRS,Z0p,mode="proba",threshold=2e-6)
print(paste("Probability-based sensitivity measure: ",sensi.proba,sep=""))

## global mode
sensi.global<-SENSI_PINCHING(Z0_IRS,Z0p,mode="global",disc=0.01)
print(paste("global sensitivity measure: ",sensi.global,sep=""))


## End(Not run)
```

SUMMARY_1CDF *Function for summarizing the uncertainty propagation's results in the form of a unique CDF.*

## Description

Function for summarizing the uncertainty propagation's results in the form of a unique CDF via the weighting average approach of Dubois and Guyonnet (2011).

## Usage

```
SUMMARY_1CDF(Z0, aversion = 0.5)
```

## Arguments

| | |
|---|---|
| Z0 | Output of the uncertainty propagation function *PROPAG()* |
| aversion | Weight value representing the decision-maker risk aversion i.e. the balance between the lower and upper CDFs. by default, *alpha*=0.5. |

## Details

Details of the theory and the example in Dubois & Guyonnet (2011) Available at: https://hal-brgm.archives-ouvertes.fr/file/index/docid/578821/filename/Uncertainties_RA_09_l_dg.pdf

## Value

Vector of the same size as the number of columns of *Z0*.

## References

Dubois D., Guyonnet D. 2011. Risk-informed decision-making under epistemic uncertainty. International Journal of General Systems, 40(2), 145-167.

## See Also

PROPAG

## Examples

```
## Not run:
##################################################
#### EXAMPLE 1 of Dubois & Guyonnet (2011)
#### Probability and Possibility distributions
##################################################

#### Model function
FUN<-function(X){
UER=X[1]
EF=X[2]
```

```
I=X[3]
C=X[4]
ED=X[5]
return(UER*I*C*EF*ED/(70*70*365))
}

ninput<-5 #Number of input parameters
input<-vector(mode="list", length=ninput) # Initialisation

input[[1]]=CREATE_INPUT(
name="UER",
type="possi",
distr="triangle",
param=c(2.e-2, 5.7e-2, 1.e-1),
monoton="incr"
)
input[[2]]=CREATE_INPUT(
name="EF",
type="possi",
distr="triangle",
param=c(200,250,350),
monoton="incr"
)
input[[3]]=CREATE_INPUT(
name="I",
type="possi",
distr="triangle",
param=c(1,1.5,2.5),
monoton="incr"
)
input[[4]]=CREATE_INPUT(
name="C",
type="proba",
distr="triangle",
param=c(5e-3,20e-3,10e-3)
)
input[[5]]=CREATE_INPUT(
name="ED",
type="proba",
distr="triangle",
param=c(10,50,30)
)

####CREATION OF THE DISTRIBUTIONS ASSOCIATED TO THE PARAMETERS
input=CREATE_DISTR(input)

####VISU INPUT
PLOT_INPUT(input)

#################################################
#### PROPAGATION

#OPTIMZATION CHOICES
```

```
choice_opt=NULL #no optimization needed
param_opt=NULL

#PROPAGATION RUN
Z0_IRS<-PROPAG(N=1000,input,FUN,choice_opt,param_opt,mode="IRS")

##################################################
#### POST-PROCESSING

# VISU - PROPAGATION
PLOT_CDF(Z0_IRS,xlab="Z",ylab="CDF",main="EX 1",lwd=1.5)

# One CDF with risk aversion of 1/3
Z<-SUMMARY_1CDF(Z0_IRS,aversion=1/3)
lines(ecdf(Z),col=5,lwd=1.5)


## End(Not run)
```

---

UNCERTAINTY *Global indicator for summarizing the epistemic uncertainty.*

---

## Description

Function for summarizing the uncertainty propagation's results in the form of a global indicator corresponding the area between the upper and lower CDFs.

## Usage

```
UNCERTAINTY(Z0, disc=0.01)
```

## Arguments

| | |
|---|---|
| Z0 | Output of the uncertainty propagation function *PROPAG()*. |
| disc | Integer to specify number of the uniform discretisation of the pair of CDFs. By default, *disc*=0.01 |

## Details

Details of the theory and the example in Dubois & Guyonnet (2011) Available at: https://hal-brgm.archives-ouvertes.fr/file/index/docid/578821/filename/Uncertainties_RA_09_l_dg.pdf

## Value

Value of the area between both CDFs.

## See Also

PROPAG PROBA_INTERVAL QUAN_INTERVAL

### Examples

```
## Not run:
#################################################
#### EXAMPLE 1 of Dubois & Guyonnet (2011)
#### Probability and Possibility distributions
#################################################

#### Model function
FUN<-function(X){
UER=X[1]
EF=X[2]
I=X[3]
C=X[4]
ED=X[5]
return(UER*I*C*EF*ED/(70*70*365))
}

ninput<-5 #Number of input parameters
input<-vector(mode="list", length=ninput) # Initialisation

input[[1]]=CREATE_INPUT(
name="UER",
type="possi",
distr="triangle",
param=c(2.e-2, 5.7e-2, 1.e-1),
monoton="incr"
)
input[[2]]=CREATE_INPUT(
name="EF",
type="possi",
distr="triangle",
param=c(200,250,350),
monoton="incr"
)
input[[3]]=CREATE_INPUT(
name="I",
type="possi",
distr="triangle",
param=c(1,1.5,2.5),
monoton="incr"
)
input[[4]]=CREATE_INPUT(
name="C",
type="proba",
distr="triangle",
param=c(5e-3,20e-3,10e-3)
)
input[[5]]=CREATE_INPUT(
name="ED",
type="proba",
distr="triangle",
param=c(10,50,30)
```

```
)

####CREATION OF THE DISTRIBUTIONS ASSOCIATED TO THE PARAMETERS
input=CREATE_DISTR(input)

####VISU INPUT
PLOT_INPUT(input)

#################################################
#### PROPAGATION

#OPTIMZATION CHOICES
choice_opt=NULL #no optimization needed
param_opt=NULL

#PROPAGATION RUN
Z0_IRS<-PROPAG(N=1000,input,FUN,choice_opt,param_opt,mode="IRS")

#################################################
#### POST-PROCESSING

# VISU - PROPAGATION
PLOT_CDF(Z0_IRS,xlab="Z",ylab="CDF",main="EX 1",lwd=1.5)

# interval of quantiles
level=0.95
quant<-QUAN_INTERVAL(Z0_IRS,level)
Qlw<-quant$Qlow
Qup<-quant$Qupp
print(paste("interval of quantiles at level:",level," : ",
"Qlow:",round(Qlw/10^floor(log10(Qlw)),
abs(floor(log10((Qup-Qlw)/10^ceiling(log10(Qlw))))))*10^floor(log10(Qlw)),
" & Qup:",round(Qup/10^floor(log10(Qup)),
abs(floor(log10((Qup-Qlw)/10^ceiling(log10(Qup))))))*10^floor(log10(Qup)),sep="")
)

# interval of probabilities
thres=1e-5
prob<-PROBA_INTERVAL(Z0_IRS,thres)
print(paste("interval of probabilities at threshold:",thres," : ",
"Plow:",prob$Plow,
" & Pup:",round(prob$Pupp,3),sep="")
)

# Global indicator of uncertainty
unc<-UNCERTAINTY(Z0_IRS)
print(paste("Epistemic uncertainty : ",unc,sep=""))


## End(Not run)
```

# Index