ByoDyn

4.8

Generated by Doxygen 1.9.8

Chapter 1

ByoDyn

Software tool for the study of biochemical networks within the framework of systems biology. SBML compatible, ByoDyn is a group effort of the Computational Biochemistry and Biophysics Lab

If you want more information, have a look to the documentation files at the docs directory.

You will find documents about installing ByoDyn, a tutorial and quick start guide, a user reference manual and the API of the program.

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

Chapter 6

Namespace Documentation

6.1 affectors Namespace Reference

Functions

- **complexExtraBack** (model, file, option, cellIndex, definition, fieldsDefinition)
- **complexExtraFwd** (model, file, option, cellIndex, definition, fieldsDefinition)
- **constant** (model, file, option, cellIndex, definition, fieldsDefinition)
- **constitutive** (model, file, option, cellIndex, definition, fieldsDefinition)
- **degradation** (model, file, option, cellIndex, definition, fieldsDefinition)
- **dissociationExtraBack** (model, file, option, cellIndex, definition, fieldsDefinition)
- **dissociationExtraFwd** (model, file, option, cellIndex, definition, fieldsDefinition)
- **inhibition** (model, file, option, cellIndex, definition, fieldsDefinition)
- **NonDimBindingDegradation** (model, file, option, cellIndex, definition, fieldsDefinition)
- **NonDimConstitutiveDegradation** (model, file, option, cellIndex, definition, fieldsDefinition)

  From here below, non dimension affectors.
- **NonDimInhibitionDegradation** (model, file, option, cellIndex, definition, fieldsDefinition)
- **NonDimTranscriptionDegradation** (model, file, option, cellIndex, definition, fieldsDefinition)
- **NonDimTranslationDegradation** (model, file, option, cellIndex, definition, fieldsDefinition)
- **NonDimTranslationDegradationBinding** (model, file, option, cellIndex, definition, fieldsDefinition)
- **SBML** (model, file, option, cellIndex, definition, fieldsDefinition)
- **transcription** (model, file, option, cellIndex, definition, fieldsDefinition)
- **translation** (model, file, option, cellIndex, definition, fieldsDefinition)

6.1.1 Function Documentation

6.1.1.1 complexExtraBack()

```
affectors.complexExtraBack (
              model,
              file,
              option,
              cellIndex,
              definition,
              fieldsDefinition )
```

```
This function writes one of the mathematical terms (the other is done by complexExtraFwd) of a biological extr
This term affects the nodes defined as receptor and ligand.
d[RECEPTOR]i / dt = - 1 / 64 k_binding_RECEPTOR.LIGAND [RECEPTOR]i <Sum>(j = 1;j = k) [LIGAND]j
d[LIGAND]i / dt = - 1 / 64  k_binding_RECEPTOR.LIGAND [LIGAND]i <Sum>(j = 1;j = k) [RECEPTOR]j
```

References **centralFunctions.neighboursFinder()**.

6.1.1.2 complexExtraFwd()

```
affectors.complexExtraFwd (
              model,
              file,
              option,
              cellIndex,
              definition,
              fieldsDefinition )

This function writes one of the mathematical terms (the other is done by complexExtraBack) of biological extra
This term affects the node defined as the complex
d[COMPLEX]i / dt = +  1 / 64 * k_binding_RECEPTOR.LIGAND [RECEPTOR]i  <Sum>(j = 1;j = k) [LIGAND]j
```

References **centralFunctions.neighboursFinder()**.

6.1.1.3 constant()

```
affectors.constant (
              model,
              file,
              option,
              cellIndex,
              definition,
              fieldsDefinition )

This function writes the mathematical term of a non changing biological node.
d [node] / dt =  0
```

6.1.1.4 constitutive()

```
affectors.constitutive (
              model,
              file,
              option,
              cellIndex,
              definition,
              fieldsDefinition )

This function writes the mathematical term of a biological constitutive expression.
d [node] / dt = constant > 0
```

6.1.1.5 degradation()

```
affectors.degradation (
              model,
              file,
              option,
              cellIndex,
              definition,
              fieldsDefinition )

This function writes the mathematical term of a biological degradation.
d [node] / dt = -k * [node]
```

6.1.1.6 dissociationExtraBack()

```
affectors.dissociationExtraBack (
              model,
              file,
              option,
              cellIndex,
              definition,
              fieldsDefinition )
```

```
This function writes one of the mathematical terms (the other is done by dissociationExtraFwd) of a biological
This term affects the node defined as the complex.
d[COMPLEX]i / dt = -  1 / 8 k_dissociation_COMPLEX <Sum>(j = 1;j = k) [COMPLEX]i
```

References **centralFunctions.neighboursFinder()**.

6.1.1.7 dissociationExtraFwd()

```
affectors.dissociationExtraFwd (
              model,
              file,
              option,
              cellIndex,
              definition,
              fieldsDefinition )
```

```
This function writes one of the mathemathical terms (the other is done by dissociationExtraFwd) of a biologica
This term affects the nodes defined as receptor and ligand.
The formula is different depending on being the receptor or the ligand
d[RECEPTOR]i / dt = + 1 / 8 k_dissociation_COMPLEX <Sum>(j = 1;j = k) [COMPLEX]i
d[LIGAND]i / dt = + 1/8 k_dissociation_COMPLEX <Sum>(j = 1;j = k) [COMPLEX]j
```

References **centralFunctions.neighboursFinder()**.

6.1.1.8 inhibition()

```
affectors.inhibition (
              model,
              file,
              option,
              cellIndex,
              definition,
              fieldsDefinition )
```

```
This function writes the mathematical term of a biological inhibition.
d [PROTEIN] / dt = + k_basal / 1 + ([INHIBITOR]/k_inhibition)^s
```

6.1.1.9 NonDimBindingDegradation()

```
affectors.NonDimBindingDegradation (
              model,
              file,
              option,
              cellIndex,
              definition,
              fieldsDefinition )

This affector compiles an complex binding and a degradation.
dcomplex/dtau = (k/n^2) * k_bind [LIGAND]_0 RECEPTOR_i \sum LIGAND_j - k_deg COMPLEX_i
```

References **centralFunctions.neighboursFinder()**.

6.1.1.10 NonDimConstitutiveDegradation()

```
affectors.NonDimConstitutiveDegradation (
              model,
              file,
              option,
              cellIndex,
              definition,
              fieldsDefinition )
```

From here below, non dimension affectors.

```
This affector compiles the constitutive gene expression and degradation.
dnode/dtau = k_deg*(rate - node)
```

6.1.1.11 NonDimInhibitionDegradation()

```
affectors.NonDimInhibitionDegradation (
              model,
              file,
              option,
              cellIndex,
              definition,
              fieldsDefinition )

This affector compiles an inhibition and a degradation.
dnode/dtau = k_deg * (1 - (INHIBITOR^m/kappa^m+INHIBITION^m) - node)
```

6.1.1.12 NonDimTranscriptionDegradation()

```
affectors.NonDimTranscriptionDegradation (
              model,
              file,
              option,
              cellIndex,
              definition,
              fieldsDefinition )

This affector compiles transcription and degradation.
dnode/dtau = k_deg(ACTIVATOR^m/(kappa^m+ACTIVATOR^m) - node)
```

6.1.1.13 NonDimTranslationDegradation()

```
affectors.NonDimTranslationDegradation (
              model,
              file,
              option,
              cellIndex,
              definition,
              fieldsDefinition )


This affector compiles translation and degradation.
dnode/dtau = k_deg(gene -PROTEIN)
```

6.1.1.14 NonDimTranslationDegradationBinding()

```
affectors.NonDimTranslationDegradationBinding (
              model,
              file,
              option,
              cellIndex,
              definition,
              fieldsDefinition )


This affector compiles the translation, degradation and ligand binding.
dNODE/dtau = k_deg(node - NODE) - (k/n^2) * k_bind [LIGAND]_0 RECEPTOR_i \sum LIGAND_j
```

References **centralFunctions.neighboursFinder()**.

6.1.1.15 SBML()

```
affectors.SBML (
              model,
              file,
              option,
              cellIndex,
              definition,
              fieldsDefinition )


This function converts the SBML formula format into the integrators' format.
It calls different functions from the formulas module.
```

References **formulas.formulaLatex()**, **formulas.readWriteFormula()**, **formulas.writeOpenModelicaFormula()**, and **formulas.writeXPPFormula()**.

6.1.1.16 transcription()

```
affectors.transcription (
              model,
              file,
              option,
              cellIndex,
              definition,
              fieldsDefinition )
```

```
This function writes the mathematical term of a biological transcription.
d[mRNA] / dt = v * [ACTIVATOR]^m / k^m + [ACTIVATOR]^m
```

6.1.1.17 translation()

```
affectors.translation (
              model,
              file,
              option,
              cellIndex,
              definition,
              fieldsDefinition )
```

```
This function writes the mathematical term of a biological translation.
d[PROTEIN] / dt = + k_translation_gene * [gene]
```

6.2 central Namespace Reference

Classes

- class **ClassFile**

Functions

- **main** (runnerFile)

  Main Program #.
- **modelReader** (metamodel)

  Main Functions #
- **optionReader** (runnerFile)
- **profile** ()
- **runner** (metamodel, model)
- **sbmlReader** (metamodel)
- **tagsReader** (metamodel)

6.2.1 Function Documentation

6.2.1.1 main()

```
central.main (
              runnerFile )
```

Main Program #.

```
This is the main function of the entire program.
It builds an object called "metamodel" with the options to run ByoDyn, another object called "model" with the
```

References **main()**, **parallel.mainProcessor()**, **modelReader()**, **optionReader()**, **parallel.receiveAny()**, **runner()**, and **parallel.sendAll()**.

Referenced by **testers.ClassTest.__execute()**, **initiator.initial()**, and **main()**.

6.2.1.2 modelReader()

```
central.modelReader (
              metamodel )
```

Main Functions #

```
This function reads the model file (the name of the file is inside the metamodel object) and creates a model o
It works for either formats, "tags" or "SBML".
```

References **sbmlReader()**, and **tagsReader()**.

Referenced by **main()**.

6.2.1.3 optionReader()

```
central.optionReader (
              runnerFile )
```

```
This function reads the runner file of ByoDyn and creates the metamodel object that contains its information.
```

References **starter.central()**.

Referenced by **main()**.

6.2.1.4 profile()

```
central.profile ( )
```

```
This function determines the number of times ByoDyn is run for profiling.
```

6.2.1.5 runner()

```
central.runner (
              metamodel,
              model )
```

```
This function discriminates the different main running options of ByoDyn and calls its appropiate functions.
The different possibilities are simulation, parameter estimation, sensitivity analysis, model format exporting
```

References **dynamicsReconstructer.central()**, **exporter.central()**, **simulator.central()**, **surface.central()**, **optimiser.central()**, **fitnessFunctionEvaluator.central()**, **identifiabilityAnalyzer.central()**, **optimal↩ ExperimentalDesign.central()**, **sampler.central()**, **sensitivityAnalyzer.central()**, **parallel.currentProcessor()**, **cluster.main()**, **parallel.mainProcessor()**, **parallel.receiveAny()**, and **parallel.sendAll()**.

Referenced by **main()**.

6.2.1.6 sbmlReader()

```
central.sbmlReader (
              metamodel )
```

```
This function reads a model in SBML format and create the model object.
```

Referenced by **modelReader()**.

6.2.1.7 tagsReader()

```
central.tagsReader (
              metamodel )
```

```
This function reads the model in "tags" format and creates the model object.
```

Referenced by **modelReader()**.

6.3 centralFunctions Namespace Reference

Functions

- **callingOctave** (outputfiles)
- **callingPython** (outputfiles)
- **createInputOctave** (metamodel, model, outputfiles)
- **createInputPython** (metamodel, model, outputfiles)
- **createOctaveOutputs** (model, metamodel, outputfiles)
- **matlabIntegrator** (metamodel, model, outputfiles)
- **neighboursFinder** (model, cellIndex)
- **octaveIntegration** (metamodel, model, outputfiles)
- **pythonIntegration** (metamodel, model, outputfiles)
- **writeFormulaOctave** (model, node, octave, option, cellIndex)
- **writeInitialConditionsOctave** (model, metamodel, octave, **function**)
- **writeParametersOctave** (model, octave)

6.3.1 Function Documentation

6.3.1.1 callingOctave()

```
centralFunctions.callingOctave (
              outputfiles )
```

This function calls Octave to integrate the system of equations of the model.
The system has been written in a file with the Octave format.

Referenced by **octaveIntegration()**.

6.3.1.2 callingPython()

```
centralFunctions.callingPython (
              outputfiles )
```

This function calls Python to integrate the system of equations of the model.
The system has been written in a file with Python format.

Referenced by **pythonIntegration()**.

6.3.1.3 createInputOctave()

```
centralFunctions.createInputOctave (
              metamodel,
              model,
              outputfiles )
```

This function creates the input file for the Octave integration.

References **formulas.readWriteFormula()**, **writeFormulaOctave()**, **writeInitialConditionsOctave()**, and **writeParametersOctave()**.

Referenced by **octaveIntegration()**.

6.3.1.4 createInputPython()

```
centralFunctions.createInputPython (
              metamodel,
              model,
              outputfiles )
```

This function creates the input file for the Python integration.

Referenced by **pythonIntegration()**.

6.3.1.5 createOctaveOutputs()

```
centralFunctions.createOctaveOutputs (
              model,
              metamodel,
              outputfiles )
```

```
This function merges the Octave outputs and sets the header of the file consisting on the model nodes and the
```

Referenced by **octaveIntegration()**.

6.3.1.6 matlabIntegrator()

```
centralFunctions.matlabIntegrator (
              metamodel,
              model,
              outputfiles )
```

```
This function is under construction.
```

Referenced by **simulator.central()**.

6.3.1.7 neighboursFinder()

```
centralFunctions.neighboursFinder (
              model,
              cellIndex )
```

```
This function determines the cell neighbour nodes of a given node.
The system needs to be multicellular.
```

Referenced by **affectors.complexExtraBack()**, **affectors.complexExtraFwd()**, **affectors.dissociationExtra↩ Back()**, **affectors.dissociationExtraFwd()**, **affectors.NonDimBindingDegradation()**, and **affectors.NonDim↩ TranslationDegradationBinding()**.

6.3.1.8 octaveIntegration()

```
centralFunctions.octaveIntegration (
              metamodel,
              model,
              outputfiles )
```

```
This function directs the main steps to simulate the system using Octave as the integrator.
```

References **callingOctave()**, **createInputOctave()**, and **createOctaveOutputs()**.

Referenced by **simulator.central()**, **simulator.obtainSimulationValues()**, **dynamicsReconstructer.run()**, and **optimiser.scoreObtainer()**.

6.3.1.9 pythonIntegration()

```
centralFunctions.pythonIntegration (
              metamodel,
              model,
              outputfiles )
```

```
This function directs the main steps to simulate the system using SciPy as the integrator.
```

References **callingPython()**, and **createInputPython()**.

Referenced by **simulator.central()**, **simulator.eventsDealer()**, **simulator.obtainSimulationValues()**, **dynamicsReconstructer.run()**, and **optimiser.scoreObtainer()**.

6.3.1.10 writeFormulaOctave()

```
centralFunctions.writeFormulaOctave (
              model,
              node,
              octave,
              option,
              cellIndex )
```

```
This function calls to the different functions of the module affectors from the lib directory.
Each affector writes the specific formula.
There is a general function for the SBML files.
```

Referenced by **createInputOctave()**.

6.3.1.11 writeInitialConditionsOctave()

```
centralFunctions.writeInitialConditionsOctave (
              model,
              metamodel,
              octave,
              function )
```

```
This function writes the initial conditions for the octave option.
It takes into account the model rules for the non constant parameters and the non constant compartments.
It also writes down the integration time and time step.
```

Referenced by **createInputOctave()**.

6.3.1.12 writeParametersOctave()

```
centralFunctions.writeParametersOctave (
              model,
              octave )
```

```
This function sets the parameters values for the octave input file.
It also takes into account the compartment values as plausible parameters of the model as they are commonly us
```

Referenced by **createInputOctave()**.

6.4 checker Namespace Reference

Classes

- class **ClassChecker**

Functions

- **main** ()

6.4.1 Function Documentation

6.4.1.1 main()

```
checker.main ( )
```

```
This function calls the different tests available.
```

Referenced by **initiator.initial()**.

6.5 cluster Namespace Reference

Functions

- **dataTransformer** (metamodel, outputfiles)
- **defaultRunner** (clusteringRange, metamodel, outputfiles)
- **main** (metamodel, outputfiles)
- **octaveCodeWriter** (metamodel, outputfiles)
- **octaveExecuter** (outputfiles)
- **plotter** (outputfiles, dimension, labels)
- **resolutionChecker** (metamodel, outputfiles)
- **surfacePlotter** (outputfiles, labels)
- **volumePlotter** (outputfiles, labels)

6.5.1 Function Documentation

6.5.1.1 dataTransformer()

```
cluster.dataTransformer (
              metamodel,
              outputfiles )
```

```
This function converts the data points to the adequate scale.
```

Referenced by **main()**, and **resolutionChecker()**.

6.5.1.2 defaultRunner()

```
cluster.defaultRunner (
              clusteringRange,
              metamodel,
              outputfiles )
```

```
This function runs the clustering for the defined range of resolution values.
```

References **octaveCodeWriter()**, and **octaveExecuter()**.

Referenced by **main()**.

6.5.1.3 main()

```
cluster.main (
              metamodel,
              outputfiles )
```

```
This is the main function for the cluster algorithm.
This function directs the flow of algorithm.
```

References **dataTransformer()**, **defaultRunner()**, **octaveCodeWriter()**, **octaveExecuter()**, **plotter()**, and **resolutionChecker()**.

Referenced by **central.runner()**.

6.5.1.4 octaveCodeWriter()

```
cluster.octaveCodeWriter (
              metamodel,
              outputfiles )
```

```
This function writes the a simple code of octave to call the clustering algorithm.
```

Referenced by **defaultRunner()**, and **main()**.

6.5.1.5 octaveExecuter()

```
cluster.octaveExecuter (
              outputfiles )
```

```
This function executes the octave code for clustering.
```

Referenced by **defaultRunner()**, and **main()**.

6.5.1.6 plotter()

```
cluster.plotter (
              outputfiles,
              dimension,
              labels )
```

```
This function plots the results of the clustering.
```

References **surfacePlotter()**, and **volumePlotter()**.

Referenced by **main()**.

6.5.1.7 resolutionChecker()

```
cluster.resolutionChecker (
              metamodel,
              outputfiles )
```

```
This function defines different resolution bandwidths and run the clustering for each value.
The results are analysed
```

References **dataTransformer()**.

Referenced by **main()**.

6.5.1.8 surfacePlotter()

```
cluster.surfacePlotter (
              outputfiles,
              labels )
```

```
This function creates a plot of the clusters in 2D. Based on sampler.plot2D
```

Referenced by **plotter()**.

6.5.1.9 volumePlotter()

```
cluster.volumePlotter (
              outputfiles,
              labels )
```

```
This function creates a plot of the clusters in 3D. Based on sampler.plot3D
```

Referenced by **plotter()**.

6.6 dynamicsReconstructer Namespace Reference

Functions

- **central** (metamodel, model, outputfiles)
- **initialConditionsDetector** (metamodel, model)
- **modelDetermination** (model, simulation, solutions, initialConditions)
- **parametersDetector** (metamodel, model)
- **plotter** (solutions, model, outputfiles, metamodel)
- **run** (model, metamodel, outputfiles)
- **runner** (metamodel, model, outputfiles, solutions, initialConditions)
- **storeInfo** (outputfiles, simulation, model)

6.6.1 Function Documentation

6.6.1.1 central()

```
dynamicsReconstructer.central (
              metamodel,
              model,
              outputfiles )
```

```
This is the central function of the module.
It checks that the parameters introduced are correct, reads the solutions and runs the model.
```

References **initialConditionsDetector()**, **parametersDetector()**, **plotter()**, and **runner()**.

Referenced by **central.runner()**.

6.6.1.2 initialConditionsDetector()

```
dynamicsReconstructer.initialConditionsDetector (
              metamodel,
              model )
```

```
This function checks for the solutions of the initial conditions in case of
```

Referenced by **central()**.

6.6.1.3 modelDetermination()

```
dynamicsReconstructer.modelDetermination (
              model,
              simulation,
              solutions,
              initialConditions )
```

```
This function sets the model parameter values.
```

Referenced by **runner()**.

6.6.1.4 parametersDetector()

```
dynamicsReconstructer.parametersDetector (
              metamodel,
              model )
```

```
This function checks that the parameters' solutions' files located at the solutionsDirectory are part of the m
```

Referenced by **central()**.

6.6.1.5 plotter()

```
dynamicsReconstructer.plotter (
              solutions,
              model,
              outputfiles,
              metamodel )
```

```
This function plots the reconstructed trajectories.
```

Referenced by **central()**.

6.6.1.6 run()

```
dynamicsReconstructer.run (
              model,
              metamodel,
              outputfiles )
```

```
This function calls the model integrators to render the dynamics.
```

References **centralFunctions.octaveIntegration()**, **centralFunctions.pythonIntegration()**, and **simulator.setIntegrationOption()**.

Referenced by **runner()**.

6.6.1.7 runner()

```
dynamicsReconstructer.runner (
              metamodel,
              model,
              outputfiles,
              solutions,
              initialConditions )
```

```
This function, for each solution,  creates and simulates the corresponding model and stores the dynamics.
```

References **modelDetermination()**, **run()**, and **storeInfo()**.

Referenced by **central()**.

6.6.1.8 storeInfo()

```
dynamicsReconstructer.storeInfo (
              outputfiles,
              simulation,
              model )
```

```
This function stores the dynamics of the solutions.
It creates a file for each node of the system.
```

Referenced by **runner()**.

6.7 errorMessages Namespace Reference

Classes

- class **ClassByoDynException**
- class **ClassCentralException**
- class **ClassCheckerException**
- class **ClassClusterException**
- class **ClassDynamicsReconstructerException**
- class **ClassFormulasException**
- class **ClassIdentifiabilityAnalyzerException**
- class **ClassInitiatorException**
- class **ClassMatrixWorkerException**
- class **ClassOptimalExperimentalDesignException**
- class **ClassOptimiserException**
- class **ClassSamplerException**
- class **ClassSBMLWorkerException**
- class **ClassSensitivityAnalyzerException**
- class **ClassSimulatorEulerException**
- class **ClassSimulatorException**
- class **ClassSimulatorStochasticException**
- class **ClassSimulatorXPPException**
- class **ClassStarterException**
- class **ClassSurfaceException**

Functions

- **ErrorHandler** (type, value, traceback)

6.7.1 Function Documentation

6.7.1.1 ErrorHandler()

```
errorMessages.ErrorHandler (
              type,
              value,
              traceback )
```

```
        This function is responsible of providing a nice format of the error message when an exception happens
```

6.8 exporter Namespace Reference

Functions

- **central** (metamodel, model, outputfiles)

6.8.1 Function Documentation

6.8.1.1 central()

```
exporter.central (
              metamodel,
              model,
              outputfiles )
```

```
This central function directs the flow of the exporting module.
```

References **sbmlWorker.sbmlWriter()**.

Referenced by **central.runner()**.

6.9 fitnessFunctionEvaluator Namespace Reference

Functions

- **central** (model, metamodel, outputfiles)
- **scoreWriter** (score, outputfiles)

6.9.1 Function Documentation

6.9.1.1 central()

```
fitnessFunctionEvaluator.central (
              model,
              metamodel,
              outputfiles )
```

```
The function checks for the compatibility of the simulation options and inserts a specified parameter value if
it checks that the data points correspond to the simulation time,
it checks the target nodes and finally evaluates the fitness function.
```

References **optimiser.checkDataPoints()**, **optimiser.checkTargetNodes()**, **simulator.compatibilityChecker()**, **optimiser.scoreObtainer()**, and **scoreWriter()**.

Referenced by **central.runner()**.

6.9.1.2 scoreWriter()

```
fitnessFunctionEvaluator.scoreWriter (
              score,
              outputfiles )
```

```
This function writes in the log file the score obtained for the fitness function evaluation.
```

Referenced by **central()**.

6.10 formulas Namespace Reference

Functions

- **checkBrackets** (formula)
- **formatPowers** (formula, option)
- **formulaLatex** (xmlFormula, file, model)
- **getMathExpression** (ASTNode)
- **includeFunctions** (model, formula)
- **piecewise** (a, b, c)
- **readWriteFormula** (model, file, cellIndex, option, formula)
- **replaceConstants** (model, constant)
- **solveFormula** (model, formula)
- **translateMathFactor** (mathFactor)
- **writeOpenModelicaFormula** (formula, file)
- **writeXPPFormula** (formula, file)

6.10.1 Function Documentation

6.10.1.1 checkBrackets()

```
formulas.checkBrackets (
              formula )
```

```
This function checks if the use of brackets is correct in a math expression.
It returns the formula modified if the number of brackets is not correct and prints a warning in the standard
```

Referenced by **readWriteFormula()**, **solveFormula()**, and **writeOpenModelicaFormula()**.

6.10.1.2 formatPowers()

```
formulas.formatPowers (
              formula,
              option )
```

```
This function checks for the powers and it substitute 'pow(k, n)' to 'k**n'.
```

References **formatPowers()**.

Referenced by **formatPowers()**.

6.10.1.3 formulaLatex()

```
formulas.formulaLatex (
              xmlFormula,
              file,
              model )
```

```
This function converts the MathML of SBML into latex format.
```

References **getMathExpression()**.

Referenced by **simulator.createPDFFormulae()**, and **affectors.SBML()**.

6.10.1.4 getMathExpression()

```
formulas.getMathExpression (
              ASTNode )
```

```
This function return a string with the math expression in latex syntax and the operator of a ASTNode.
ASTNode is a object codifying a formula using a abstract syntax tree object.
It does recursive calls to itself in order to build the math expression.
```

References **getMathExpression()**, and **translateMathFactor()**.

Referenced by **simulator.createPDFFormulae()**, **formulaLatex()**, and **getMathExpression()**.

6.10.1.5 includeFunctions()

```
formulas.includeFunctions (
              model,
              formula )
```

```
This function get a formula and return a new formula without external functions used.
A external function is a non-python function like the user-defined functions in SBML.
It replaces the function definitions with the function output in order to create a string without external fur
```

References **includeFunctions()**.

Referenced by **includeFunctions()**, and **solveFormula()**.

6.10.1.6 piecewise()

```
formulas.piecewise (
              a,
              b,
              c )
```

```
This function converts the format of piecewise function in MathML to the piecewise function in Scipy.
```

6.10.1.7 readWriteFormula()

```
formulas.readWriteFormula (
              model,
              file,
              cellIndex,
              option,
              formula )
```

This funtion inputs the SBML string of the formula and writes out on Python or Octave formats.

References **checkBrackets()**.

Referenced by **centralFunctions.createInputOctave()**, and **affectors.SBML()**.

6.10.1.8 replaceConstants()

```
formulas.replaceConstants (
              model,
              constant )
```

This functions replaces a parameter for its value or a variable for its initial condition value.

Referenced by **solveFormula()**.

6.10.1.9 solveFormula()

```
formulas.solveFormula (
              model,
              formula )
```

This function evaluates a algebraic string, replacing all the parameters and variables for its values in time

References **checkBrackets()**, **includeFunctions()**, and **replaceConstants()**.

Referenced by **sbmlWorker.ClassModelSBML.__searchDelayFunction()**, and **sbmlWorker.ClassModel↩ SBML.checkAssignmentRule()**.

6.10.1.10 translateMathFactor()

```
formulas.translateMathFactor (
              mathFactor )
```

This function converts operators to python format.

Referenced by **getMathExpression()**.

6.10.1.11 writeOpenModelicaFormula()

```
formulas.writeOpenModelicaFormula (
              formula,
              file )
```

```
This function inputs the SBML string of the formula and writes out on OpenModelica format.
```

References **checkBrackets()**.

Referenced by **simulatorOpenModelica.ClassSimulatorOpenModelica.__writeModel()**, and **affectors.SBML()**.

6.10.1.12 writeXPPFormula()

```
formulas.writeXPPFormula (
              formula,
              file )
```

```
This function inputs the SBML string of the formula and writes out on XPP format.
```

Referenced by **simulatorXPP.ClassSimulatorXPP.createInput()**, and **affectors.SBML()**.

6.11 gssa Namespace Reference

Functions

- **simulate** (evalPropensities, stoichiometry, x_0, time, hurdle, seed=None)

6.11.1 Function Documentation

6.11.1.1 simulate()

```
gssa.simulate (
              evalPropensities,
              stoichiometry,
              x_0,
              time,
              hurdle,
              seed = None )
```

```
Gillespie's Stochastic Simulation Algorithm (SSA)
    evalPropensities  list of propensity functions
    stoichiometry     stoichiometrty matrix, each column corresponds to a reaction
    x_0               initial conditions
    time              simulation time
    hurdle            the time step for which the system state is written (if set
                      to None all reactions are written in the output)
    seed              for the pseudo-random number generator
```

6.12 identifiabilityAnalyzer Namespace Reference

Classes

- class **ClassIdentifiabilityAnalyzer**

Functions

- **central** (model, metamodel, outputfiles)

6.12.1 Function Documentation

6.12.1.1 central()

```
identifiabilityAnalyzer.central (
              model,
              metamodel,
              outputfiles )
```

```
This is the central function of the module.
It directs the flow of the program for the identifiability analysis.
```

References **sensitivityAnalyzer.componentsChecker()**.

Referenced by **central.runner()**.

6.13 initiator Namespace Reference

Functions

- **createExamples** ()
- **initial** (runnerFile=None)
- **printingHelp** ()
- **printingVersion** (version)
- **versionDefinitor** ()

Variables

- str **benchmarkdir** = os.environ.get('BYODYN_PATH') + '/benchmark'
- str **libdir** = os.environ.get('BYODYN_PATH') + '/lib'
- str **srcdir** = os.environ.get('BYODYN_PATH') + '/src'

6.13.1 Function Documentation

6.13.1.1 createExamples()

```
initiator.createExamples ( )
```

The function creates the examples of ByoDyn at the examples directory.

Referenced by **initial()**.

6.13.1.2 initial()

```
initiator.initial (
              runnerFile = None )
```

This function reads the command line options, it configures the ByoDyn options and checks its consistency and

References **createExamples()**, **checker.main()**, **central.main()**, **parallel.mainProcessor()**, **printingHelp()**, **printingVersion()**, **parallel.receiveAny()**, **parallel.sendAll()**, and **versionDefinitor()**.

6.13.1.3 printingHelp()

```
initiator.printingHelp ( )
```

This function prints the command line help.

Referenced by **initial()**.

6.13.1.4 printingVersion()

```
initiator.printingVersion (
              version )
```

This function prints the current ByoDyn version.

Referenced by **initial()**.

6.13.1.5 versionDefinitor()

```
initiator.versionDefinitor ( )
```

This funciton defines the ByoDyn version globally.

Referenced by **initial()**.

6.13.2 Variable Documentation

6.13.2.1 benchmarkdir

```
str initiator.benchmarkdir = os.environ.get('BYODYN_PATH') + '/benchmark'
```

6.13.2.2 libdir

```
str initiator.libdir = os.environ.get('BYODYN_PATH') + '/lib'
```

6.13.2.3 srcdir

```
str initiator.srcdir = os.environ.get('BYODYN_PATH') + '/src'
```

6.14 localOptimiser Namespace Reference

Functions

- **central** (model, metamodel, outputfiles)
- **globalDefiner** (model, metamodel, outputfiles)
- **locatingModuleDir** (metamodel)
- **optimisation** (threshold)
- **pcalcr** (x, p)
- **savingResults** (value, model, metamodel, outputfiles)

6.14.1 Function Documentation

6.14.1.1 central()

```
localOptimiser.central (
              model,
              metamodel,
              outputfiles )
```

```
This is the main function of the module.
First it appends the compiling directory to the system path.
Then it makes global the three main objects of the program (model, metamodel and outputfiles) and the module l
It calls the optimisation routine, obtains the new value of the parameters and the fitness function value.
It updates the model and sends the very last fitness function.
```

References **globalDefiner()**, **optimisation()**, and **savingResults()**.

Referenced by **optimiser.central()**, **optimiser.evaluateSolution()**, and **optimiser.gaPopulationScores↩ ObtainerLocal()**.

6.14.1.2 globalDefiner()

```
localOptimiser.globalDefiner (
              model,
              metamodel,
              outputfiles )
```

```
This function adds the compilation directory to the system path.
Then it makes global the model, metamodel and outputfiles objects and the localSearch fortran function.
It finally defines the metamodel.hybridScore variable as True for the correct calculation of the score: the Fo
```

References **locatingModuleDir()**.

Referenced by **central()**.

6.14.1.3 locatingModuleDir()

```
localOptimiser.locatingModuleDir (
              metamodel )
```

```
This function appends the compiling directory to the system path.
```

Referenced by **globalDefiner()**.

6.14.1.4 optimisation()

```
localOptimiser.optimisation (
              threshold )
```

```
This function interacts with the Fortran code directly.
It calls the main routine of the Fortran program.
It sends the variables: x for the parameter values and b for the parameter ranges. And optionally it can send
Finally it collects the optimised parameter vector and the fitness function value.
```

Referenced by **central()**.

6.14.1.5 pcalcr()

```
localOptimiser.pcalcr (
              x,
              p )
```

```
This function is called by the Fortran program localSearch.
This function obtains the fitness function value for the new parameter position given by the Fortran program.
```

References **optimiser.scoreObtainer()**.

6.14.1.6 savingResults()

```
localOptimiser.savingResults (
              value,
              model,
              metamodel,
              outputfiles )

This function saves the results of the local optimisation.
```

References **sbmlWorker.sbmlWriter()**.

Referenced by **central()**.

6.15 matrixWorker Namespace Reference

Classes

- class **ClassMatrix**

6.16 optimalExperimentalDesign Namespace Reference

Functions

- **addNewPoint** (identifiability, sensitivity, metamodel, model, outputfiles, parametersToStudy)
- **central** (model, metamodel, outputfiles)
- **chooseValue** (criteria, values)
- **getCriteria** (criteria, identifiability)
- **rankTargets** (identifiability, sensitivity, metamodel, model, outputfiles, parametersToStudy)
- **timePointsDeterminer** (metamodel)

6.16.1 Function Documentation

6.16.1.1 addNewPoint()

```
optimalExperimentalDesign.addNewPoint (
              identifiability,
              sensitivity,
              metamodel,
              model,
              outputfiles,
              parametersToStudy )

This function runs the addNewPoint optimal experimental design protocol.
In this task we evaluate the criteria value at different timePoints after adding a putative target of the give
We can also run the protocol for a list of species.
```

References **chooseValue()**, **getCriteria()**, and **timePointsDeterminer()**.

Referenced by **central()**.

6.16.1.2 central()

```
optimalExperimentalDesign.central (
              model,
              metamodel,
              outputfiles )
```

```
This is the central function of the module.
It directs the flow of the program for the optimal experimental design.
```

References **addNewPoint()**, **sensitivityAnalyzer.componentsChecker()**, and **rankTargets()**.

Referenced by **central.runner()**.

6.16.1.3 chooseValue()

```
optimalExperimentalDesign.chooseValue (
              criteria,
              values )
```

```
This function returns the maximmum or minimmun value of values lists depending on the criteria type.
```

Referenced by **addNewPoint()**.

6.16.1.4 getCriteria()

```
optimalExperimentalDesign.getCriteria (
              criteria,
              identifiability )
```

```
This function returns the criteria value specified by the user in runner.
```

Referenced by **addNewPoint()**, and **rankTargets()**.

6.16.1.5 rankTargets()

```
optimalExperimentalDesign.rankTargets (
              identifiability,
              sensitivity,
              metamodel,
              model,
              outputfiles,
              parametersToStudy )
```

```
This function is still in development status. It sorts of targets indicating the more and the less important.
Less important means with low effect to improve the identifiability criteria.
```

References **getCriteria()**.

Referenced by **central()**.

6.16.1.6 timePointsDeterminer()

```
optimalExperimentalDesign.timePointsDeterminer (
              metamodel )
```

```
This function returns the time points of study of the optimal experimental design.
It is based on the metamodel arguments of time&timestep and OEDResolution.
```

Referenced by **addNewPoint()**.

6.17 optimiser Namespace Reference

Functions

- **calculateConfidenceIntervals** (model, metamodel, outputfiles)
- **central** (metamodel, model, outputfiles, solutions)
- **checkDataPoints** (model, metamodel)
- **checkInitialConditionsToVary** (model, metamodel)
- **checkParametersToVary** (model, metamodel)
- **checkTargetNodes** (model, metamodel)
- **evaluateIteration** (karyotype, model, metamodel, outputfiles, scores, iteration)
- **evaluateSolution** (iteration, score, metamodel, model, outputfiles)
- **evaluateStopping** (metamodel, iteration, scores)
- **gaInitialisePopulation** (model, metamodel)
- **gaNaturalSelection** (karyotype, iteration, scores, metamodel, outputfiles)
- **gaPopulationScoresObtainer** (iteration, karyotype, metamodel, model, outputfiles)
- **gaPopulationScoresObtainerLocal** (iteration, karyotype, metamodel, model, outputfiles)
- **gaSex** (karyotype, bestChromosomes, sexChromosomes, model, metamodel)
- **getSimulateValue** (originalSimulationValues, node, target, model, metamodel)
- **linearVariation** (minimalValue, maximalValue)
- **logarithmicVariation** (minimalValue, maximalValue)
- **normalVariationForInitialCondition** (metamodel, model)
- **randomSearch** (metamodel, model, outputfiles)
- **scoreObtainer** (metamodel, model, outputfiles)
- **storeSolution** (score, metamodel, model, workingDirectory)

6.17.1 Function Documentation

6.17.1.1 calculateConfidenceIntervals()

```
optimiser.calculateConfidenceIntervals (
              model,
              metamodel,
              outputfiles )
```

```
This function calculates the covariance matrix and returns the object identifiability.
This object is necessary to calculate the confidence intervals.
The implementation of confidence intervals is still under development, not finished yet.
```

Referenced by **evaluateSolution()**.

6.17.1.2 central()

```
optimiser.central (
              metamodel,
              model,
              outputfiles,
              solutions )

This function directs the optimisation to the different optimisation options.
```

References **localOptimiser.central()**, **checkDataPoints()**, **checkInitialConditionsToVary()**, **checkParametersToVary()**, **checkTargetNodes()**, **evaluateSolution()**, **evaluateStopping()**, **gaInitialisePopulation()**, **gaNaturalSelection()**, **gaPopulationScoresObtainer()**, **gaPopulationScoresObtainerLocal()**, **gaSex()**, **linearVariation()**, **logarithmicVariation()**, **parallel.mainProcessor()**, **normalVariationForInitialCondition()**, **randomSearch()**, **parallel.receiveAny()**, **scoreObtainer()**, and **parallel.sendAll()**.

Referenced by **central.runner()**.

6.17.1.3 checkDataPoints()

```
optimiser.checkDataPoints (
              model,
              metamodel )

This function checks for the consistency of the target.
It checks if the target constrain has a time value larger than the simulation time.
```

Referenced by **central()**, **fitnessFunctionEvaluator.central()**, and **sampler.central()**.

6.17.1.4 checkInitialConditionsToVary()

```
optimiser.checkInitialConditionsToVary (
              model,
              metamodel )

This function checks for the consistency of the initial conditions to vary.
It checks that the values of the initial conditions are part of the nodes.
Apart it checks for incompatibilities about model boundary conditions.
```

Referenced by **central()**.

6.17.1.5 checkParametersToVary()

```
optimiser.checkParametersToVary (
              model,
              metamodel )

This function checks for the consistency of the introduced parameters to vary:
the parameter to vary has to be one of the model's,
the parameter to vary cannot be a constant parameter and
the lower value of the exploring range has to be specified before the higher value.
```

Referenced by **central()**, and **sampler.central()**.

6.17.1.6 checkTargetNodes()

```
optimiser.checkTargetNodes (
              model,
              metamodel )
```

```
This function checks that the nodes of the experimental data are part of the model.
```

Referenced by **central()**, **fitnessFunctionEvaluator.central()**, and **sampler.central()**.

6.17.1.7 evaluateIteration()

```
optimiser.evaluateIteration (
              karyotype,
              model,
              metamodel,
              outputfiles,
              scores,
              iteration )
```

```
This function helps the evaluation of the stopping of the program.
It evaluates at the same time the stopper options and the stopper variables.
It modifies the stopper options for the proper evaluation of the stopping by evaluateStopping.
```

References **evaluateSolution()**.

Referenced by **gaPopulationScoresObtainer()**, and **gaPopulationScoresObtainerLocal()**.

6.17.1.8 evaluateSolution()

```
optimiser.evaluateSolution (
              iteration,
              score,
              metamodel,
              model,
              outputfiles )
```

```
This function evaluates if the solution is susceptible of being stored.
It involves the creation of solution directories and files.
```

References **calculateConfidenceIntervals()**, **localOptimiser.central()**, **sbmlWorker.sbmlWriter()**, and **store↩ Solution()**.

Referenced by **central()**, **evaluateIteration()**, and **gaPopulationScoresObtainer()**.

6.17.1.9 evaluateStopping()

```
optimiser.evaluateStopping (
              metamodel,
              iteration,
              scores )

This function evaluates if the program needs to stop due to the score value or the iteration number.
```

Referenced by **central()**.

6.17.1.10 gaInitialisePopulation()

```
optimiser.gaInitialisePopulation (
              model,
              metamodel )

This function create the initial random karyotype for the genetic algorithm.
```

References **linearVariation()**, and **logarithmicVariation()**.

Referenced by **central()**.

6.17.1.11 gaNaturalSelection()

```
optimiser.gaNaturalSelection (
              karyotype,
              iteration,
              scores,
              metamodel,
              outputfiles )

This function selects the best fit elements from the last generation karyotype.
```

Referenced by **central()**.

6.17.1.12 gaPopulationScoresObtainer()

```
optimiser.gaPopulationScoresObtainer (
              iteration,
              karyotype,
              metamodel,
              model,
              outputfiles )

This function evaluates the score of the karyotype by calling scoreObtainer function.
Special emphasis is given for the parallel code.
```

References **parallel.currentProcessor()**, **evaluateIteration()**, **evaluateSolution()**, **parallel.mainProcessor()**, **parallel.receive()**, **parallel.receiveAny()**, **scoreObtainer()**, **parallel.send()**, **parallel.sendAll()**, and **parallel.↩ totalProcessors()**.

Referenced by **central()**.

6.17.1.13 gaPopulationScoresObtainerLocal()

```
optimiser.gaPopulationScoresObtainerLocal (
              iteration,
              karyotype,
              metamodel,
              model,
              outputfiles )
```

```
This function evaluates the fitness function for the hybridOnePhase algorithm.
For each element it launches a local search algorithm and takes as fitness function value the result of the lo
```

References **localOptimiser.central()**, and **evaluateIteration()**.

Referenced by **central()**.

6.17.1.14 gaSex()

```
optimiser.gaSex (
              karyotype,
              bestChromosomes,
              sexChromosomes,
              model,
              metamodel )
```

```
This function directs the crossing over and mutation from one to the next generation.
```

References **linearVariation()**, and **logarithmicVariation()**.

Referenced by **central()**.

6.17.1.15 getSimulateValue()

```
optimiser.getSimulateValue (
              originalSimulationValues,
              node,
              target,
              model,
              metamodel )
```

```
This function retrieves the simulation value of a single node at a single time point.
```

6.17.1.16 linearVariation()

```
optimiser.linearVariation (
              minimalValue,
              maximalValue )
```

```
This function generates a random value for a parameter explored in a linear scale range.
```

Referenced by **central()**, **sampler.ClassSample.createRandomSetOfParameters()**, **gaInitialisePopulation()**, **gaSex()**, and **randomSearch()**.

6.17.1.17 logarithmicVariation()

```
optimiser.logarithmicVariation (
              minimalValue,
              maximalValue )

This function generates a random value for a parameter explored in a logarithmic
```

Referenced by **central()**, **sampler.ClassSample.createRandomSetOfParameters()**, **gaInitialisePopulation()**, **gaSex()**, and **randomSearch()**.

6.17.1.18 normalVariationForInitialCondition()

```
optimiser.normalVariationForInitialCondition (
              metamodel,
              model )

This function variates the initial condition of the model based on a normal distribution.
```

Referenced by **central()**, and **randomSearch()**.

6.17.1.19 randomSearch()

```
optimiser.randomSearch (
              metamodel,
              model,
              outputfiles )

This function evaluates the score of a new random point in the parameter space.
```

References **linearVariation()**, **logarithmicVariation()**, **normalVariationForInitialCondition()**, and **scoreObtainer()**.

Referenced by **central()**.

6.17.1.20 scoreObtainer()

```
optimiser.scoreObtainer (
              metamodel,
              model,
              outputfiles )

This function obtains the fitness function value given a model and the experimental target.
It involves the simulation of the model.
A set of target values are determined, with its corresponding importance, that is, the ponderation.
Then a set of corresponding values at the simulation are searched. Because at the corresponding time there are
The values are substracted from one list to the other.
```

References **centralFunctions.octaveIntegration()**, **centralFunctions.pythonIntegration()**, and **simulator.setIntegrationOption()**.

Referenced by **sampler.ClassSample.calculateFitnessFunction()**, **surface.central()**, **central()**, **fitnessFunctionEvaluator.central()**, **gaPopulationScoresObtainer()**, **localOptimiser.pcalcr()**, and **randomSearch()**.

6.17.1.21 storeSolution()

```
optimiser.storeSolution (
              score,
              metamodel,
              model,
              workingDirectory )
```

```
This function saves a solution in the tags or SBML format.
```

References **sbmlWorker.sbmlWriter()**.

Referenced by **evaluateSolution()**.

6.18 parallel Namespace Reference

Functions

- **currentProcessor** ()
- **mainProcessor** ()
- **receive** (Source, Tag)
- **receiveAny** (Tag)
- **runsOnParallelMachine** ()
- **send** (Object, Destination, Tag)
- **sendAll** (Object, Tag)
- **totalProcessors** ()
- **waitProcessors** ()

Variables

- **communicator** = None
- dict **internalBuffer** = {}

6.18.1 Function Documentation

6.18.1.1 currentProcessor()

```
parallel.currentProcessor ( )
```

```
This function provides which is the current processor.
It returns an integer.
```

Referenced by **optimiser.gaPopulationScoresObtainer()**, **simulator.obtainSimulationValues()**, and **central.runner()**.

6.18.1.2 mainProcessor()

```
parallel.mainProcessor ( )
```

```
This function discriminates if the current processor is the main processor, that is, processor 0
It returns a boolean variable.
```

Referenced by **optimiser.central()**, **optimiser.gaPopulationScoresObtainer()**, **initiator.initial()**, **central.main()**, and **central.runner()**.

6.18.1.3 receive()

```
parallel.receive (
              Source,
              Tag )
```

```
This function receives an object from a determined processor.
It takes as arguments:
Source, the source processor, and
Tag, an aditional identifier of the object.
It returns the object.
```

Referenced by **optimiser.gaPopulationScoresObtainer()**.

6.18.1.4 receiveAny()

```
parallel.receiveAny (
              Tag )
```

```
This function receives an object from any processor.
It takes as argument, Tag, an aditional identifier of the object.
It returns the object.
```

Referenced by **optimiser.central()**, **optimiser.gaPopulationScoresObtainer()**, **initiator.initial()**, **central.main()**, and **central.runner()**.

6.18.1.5 runsOnParallelMachine()

```
parallel.runsOnParallelMachine ( )
```

```
This function determines if the program runs in parallel or serial environment.
It returns a boolean variable.
```

6.18.1.6 send()

```
parallel.send (
              Object,
              Destination,
              Tag )
```

```
This function sends an object to a defined processor.
It takes as arguments:
Object, the object,
Destination, the destination processor, and
Tag, an aditional identifier of the object.
It returns nothing.
```

Referenced by **optimiser.gaPopulationScoresObtainer()**.

6.18.1.7 sendAll()

```
parallel.sendAll (
              Object,
              Tag )
```

```
This function send a given object to all processors.
It takes as arguments:
Object, the object to be sent, and
Tag, an aditional identifier of the object.
It returns nothing.
```

Referenced by **optimiser.central()**, **optimiser.gaPopulationScoresObtainer()**, **initiator.initial()**, **central.↩ main()**, and **central.runner()**.

6.18.1.8 totalProcessors()

```
parallel.totalProcessors ( )
```

```
This function provides the amount of processors.
It returns an integer.
```

Referenced by **optimiser.gaPopulationScoresObtainer()**.

6.18.1.9 waitProcessors()

```
parallel.waitProcessors ( )
```

```
This function makes all the processors to wait up to that point.
```

6.18.2 Variable Documentation

6.18.2.1 communicator

```
parallel.communicator = None
```

6.18.2.2 internalBuffer

```
dict parallel.internalBuffer = {}
```

6.19 pca Namespace Reference

Classes

- class **ClassPCA**

Functions

- **central** ()

6.19.1 Function Documentation

6.19.1.1 central()

```
pca.central ( )
```

6.20 profiler Namespace Reference

Functions

- **main** ()

Variables

- str **ERROR_FILE** = '%s/profilingErrorFile' % str(**scratchDir**)
- str **PROFILE_FILE** = '%s/profilingOutputFile' % str(**scratchDir**)
- **scratchDir** = os.environ.get('BYODYN_SCRATCH_DIR')
- **SCRIPTNAME** = central
- str **STAT_FILE** = '%s/profilingStatFile' % str(**scratchDir**)

6.20.1 Function Documentation

6.20.1.1 main()

```
profiler.main ( )
```

```
        This is the main function of the profiling.
```

References **main()**.

Referenced by **main()**.

6.20.2 Variable Documentation

6.20.2.1 ERROR_FILE

```
str profiler.ERROR_FILE = '%s/profilingErrorFile' % str( scratchDir)
```

6.20.2.2 PROFILE_FILE

```
str profiler.PROFILE_FILE = '%s/profilingOutputFile' % str( scratchDir)
```

6.20.2.3 scratchDir

```
profiler.scratchDir = os.environ.get('BYODYN_SCRATCH_DIR')
```

6.20.2.4 SCRIPTNAME

```
profiler.SCRIPTNAME = central
```

6.20.2.5 STAT_FILE

```
str profiler.STAT_FILE = '%s/profilingStatFile' % str( scratchDir)
```

6.21 sampler Namespace Reference

Classes

- class **ClassSample**
- class **ClassSamplerMonteCarlo**

Functions

- **central** (model, metamodel, outputfiles)
- **plot1D** (metamodel, outputfiles, sample)
- **plot2D** (metamodel, outputfiles, sample)
- **plot3D** (metamodel, outputfiles, sample)
- **plotResults** (metamodel, outputfiles, sample)
- **rainbowColourCalculator** (value)
- **storeResults** (listOfSample, file, metamodel)

6.21.1 Function Documentation

6.21.1.1 central()

```
sampler.central (
              model,
              metamodel,
              outputfiles )
```

```
This is the central function of the module.
It directs the flow of the Monte Carlo sampling depending on the options.
```

References **optimiser.checkDataPoints()**, **optimiser.checkParametersToVary()**, **optimiser.checkTargetNodes()**, **simulator.compatibilityChecker()**, **plotResults()**, and **storeResults()**.

Referenced by **central.runner()**.

6.21.1.2 plot1D()

```
sampler.plot1D (
              metamodel,
              outputfiles,
              sample )
```

```
This function creates a histogram of the distribution of the accepted sampled points.
```

Referenced by **plotResults()**.

6.21.1.3 plot2D()

```
sampler.plot2D (
              metamodel,
              outputfiles,
              sample )
```

```
This function plots a 2D graphs with the accepted sampled points.
```

References **rainbowColourCalculator()**.

Referenced by **plotResults()**.

6.21.1.4 plot3D()

```
sampler.plot3D (
              metamodel,
              outputfiles,
              sample )
```

This function creates a gnuplot script to plot in 3D the Monte Carlo results.

Referenced by **plotResults()**.

6.21.1.5 plotResults()

```
sampler.plotResults (
              metamodel,
              outputfiles,
              sample )
```

This function creates a 1D, 2D or 3D graphs if the parameter space is 1, 2 or 3 dimensional.
Otherwise it gives a message expressing the impossibility of the graph.

References **plot1D()**, **plot2D()**, and **plot3D()**.

Referenced by **central()**.

6.21.1.6 rainbowColourCalculator()

```
sampler.rainbowColourCalculator (
              value )
```

Given a normalised value on the 0-4 range, this function returns the corresponding RGB colour code based on th

Referenced by **plot2D()**.

6.21.1.7 storeResults()

```
sampler.storeResults (
              listOfSample,
              file,
              metamodel )
```

This method stores the sampling solution on its specific file.

Referenced by **central()**.

6.22 sbmlWorker Namespace Reference

Classes

- class **ClassEvent**
- class **ClassFunction**
- class **ClassModelSBML**
- class **ClassReaction**
- class **ClassRules**

Functions

- **divCompartment** (sbmlModel, formula, node)
- **getBooleanExpression** (ASTNode)
- **sbmlWriter** (w, model, metamodel, file)
- **translateLocalParametersNamesInFormula** (formula, reaction)
- **translateMathFactor** (mathFactor)

6.22.1 Function Documentation

6.22.1.1 divCompartment()

```
sbmlWorker.divCompartment (
              sbmlModel,
              formula,
              node )
```

```
This function divides the formula by compartment volume if it is needed.
```

Referenced by **sbmlWorker.ClassModelSBML.__readTopology()**.

6.22.1.2 getBooleanExpression()

```
sbmlWorker.getBooleanExpression (
              ASTNode )
```

```
This function reads an ASTNode (equation in Abstract Syntax Tree class)
and builds the equation as a math string using recursive calls.
It returns the math string and the operator.
```

References **getBooleanExpression()**, and **translateMathFactor()**.

Referenced by **sbmlWorker.ClassModelSBML.__readEvents()**, and **getBooleanExpression()**.

6.22.1.3 sbmlWriter()

```
sbmlWorker.sbmlWriter (
              w,
              model,
              metamodel,
              file )
```

```
This function writes a SBML model given the object model.
```

Referenced by **exporter.central()**, **optimiser.evaluateSolution()**, **localOptimiser.savingResults()**, and **optimiser.storeSolution()**.

6.22.1.4 translateLocalParametersNamesInFormula()

```
sbmlWorker.translateLocalParametersNamesInFormula (
              formula,
              reaction )
```

```
This function replaces in a formula all the names of parameters,
because we need convert the local paramters to a new format ('parameterId''ReactionId')
in order to work correctly with SBML definition of parameters
where the local and global parameters can have the same ids.
```

Referenced by **sbmlWorker.ClassModelSBML.__readTopology()**.

6.22.1.5 translateMathFactor()

```
sbmlWorker.translateMathFactor (
              mathFactor )
```

```
This function converts some string logical affectors to Open Modelica format.
```

Referenced by **getBooleanExpression()**.

6.23 sensitivityAnalyzer Namespace Reference

Classes

- class **ClassSensitivityAnalyzer**

Functions

- **central** (model, metamodel, outputfiles)
- **componentsChecker** (model, metamodel)
- **getNodeName** (value, model)
- **getPositionNode** (field, model)

6.23.1 Function Documentation

6.23.1.1 central()

```
sensitivityAnalyzer.central (
              model,
              metamodel,
              outputfiles )
```

```
This is the central function of the module.
It directs the flow of the program for the sensitivity analysis and the identifiability analysis.
```

References **componentsChecker()**.

Referenced by **central.runner()**.

6.23.1.2 componentsChecker()

```
sensitivityAnalyzer.componentsChecker (
              model,
              metamodel )
```

```
This function is an initial checking of the parameters we want to study:
first that they exist on the model and second that they are constant during the simulation.
Finally we select the parameters to study on a new working variable.
At the very end we set the new parameters values set by "parameter" variable.
```

Referenced by **identifiabilityAnalyzer.central()**, **optimalExperimentalDesign.central()**, and **central()**.

6.23.1.3 getNodeName()

```
sensitivityAnalyzer.getNodeName (
              value,
              model )
```

```
Given the position of a node in model.nodes list
this function returns its name.
In the case of multicellular models the index positions
are added in the format: "nodeName(yindex,xindex)".
```

Referenced by **sensitivityAnalyzer.ClassSensitivityAnalyzer.__obtainSensitivity()**.

6.23.1.4 getPositionNode()

```
sensitivityAnalyzer.getPositionNode (
              field,
              model )
```

```
Given a target field ("node/xindex,yindex/experimentalValue/variance"),
this function returns the column position of this node
in the simulation output file.
```

Referenced by **sensitivityAnalyzer.ClassSensitivityAnalyzer.__obtainSensitivity()**.

6.24 simulator Namespace Reference

Classes

- class **ClassEpithelium**

Functions

- **central** (metamodel, model, outputfiles)
- **checkResults** (metamodel, trajectoriesFile)
- **compatibilityChecker** (metamodel, model)
- **createEpiplot** (model, metamodel, outputfiles)
- **createGnuplot** (model, metamodel, outputfiles, type)
- **createPDFFormulae** (model, metamodel, outputfiles)
- **eventsDealer** (metamodel, model, outputfiles)
- **flatFileWriter** (outputfiles, grid)
- **isAGene** (node)
- **lastStateRetriever** (metamodel, model, outputfiles)
- **obtainSimulationValues** (metamodel, model, outputfiles, type)
- **plotMaker** (outputfiles, grid, model)
- **setIntegrationOption** (model, tester)

6.24.1 Function Documentation

6.24.1.1 central()

```
simulator.central (
              metamodel,
              model,
              outputfiles )
```

```
This function controls the flow of the simulation depending on the simulation options.
It checks the correctness of the input data,
it creates a pdf format file with the system of equations,
it integrates the system with the selected integrator and
it finally builds the graphs of the simulation.
```

References **checkResults()**, **compatibilityChecker()**, **createEpiplot()**, **createGnuplot()**, **createPDFFormulae()**, **eventsDealer()**, **centralFunctions.matlabIntegrator()**, **centralFunctions.octaveIntegration()**, **centralFunctions.pythonIntegration()**, and **setIntegrationOption()**.

Referenced by **central.runner()**.

6.24.1.2 checkResults()

```
simulator.checkResults (
              metamodel,
              trajectoriesFile )
```

```
This function checks the integration output file.
It returns an error if the results are missing and a warning if the results are incomplete.
```

Referenced by **central()**.

6.24.1.3 compatibilityChecker()

```
simulator.compatibilityChecker (
              metamodel,
              model )

This function checks the compatibility of some simulation options:
if the model holds events has to be integrated by OpenModelica and
that a certain parameter of the model exists for which its value is specified.
```

Referenced by **central()**, **fitnessFunctionEvaluator.central()**, and **sampler.central()**.

6.24.1.4 createEpiplot()

```
simulator.createEpiplot (
              model,
              metamodel,
              outputfiles )

This function creates the multicellular plots.
It calls the class ClassEpithelium, fills the object
and creates a flat file data and the graph.
```

References **flatFileWriter()**, and **plotMaker()**.

Referenced by **central()**.

6.24.1.5 createGnuplot()

```
simulator.createGnuplot (
              model,
              metamodel,
              outputfiles,
              type )

This function creates the postscript plots for the dynamic trajectories.
It is responsible for the plotting of both the concentration versus time plots
and the change of concentration versus time plots.
```

Referenced by **central()**, and **obtainSimulationValues()**.

6.24.1.6 createPDFFormulae()

```
simulator.createPDFFormulae (
              model,
              metamodel,
              outputfiles )

This function creates the latex format files with the system of equations.
It calls the different affector functions at the affectors module of the lib directory.
```

References **formulas.formulaLatex()**, and **formulas.getMathExpression()**.

Referenced by **central()**.

6.24.1.7 eventsDealer()

```
simulator.eventsDealer (
              metamodel,
              model,
              outputfiles )
```

```
This function simulates the model in the case it holds events.
It runs the different simulations depending on the number of events and
it joins them finally.
```

References **lastStateRetriever()**, and **centralFunctions.pythonIntegration()**.

Referenced by **central()**.

6.24.1.8 flatFileWriter()

```
simulator.flatFileWriter (
              outputfiles,
              grid )
```

```
This function writes a flat file with the information of the ClassEpithelium.
```

Referenced by **createEpiplot()**.

6.24.1.9 isAGene()

```
simulator.isAGene (
              node )
```

```
This function returns a boolean determining if the node is or not a gene.
The function's answer is based on whether the first character of the string is capital or not.
This function is only used to determine the font of the node at the multicellular plot.
```

6.24.1.10 lastStateRetriever()

```
simulator.lastStateRetriever (
              metamodel,
              model,
              outputfiles )
```

```
This function returns the values of the nodes at the last time of the simulation.
```

Referenced by **eventsDealer()**.

6.24.1.11 obtainSimulationValues()

```
simulator.obtainSimulationValues (
              metamodel,
              model,
              outputfiles,
              type )
```

```
This function obtains the simulation values of a given model.
It is used by the optimiser and the sensitivityAnalyzer modules.
```

References **createGnuplot()**, **parallel.currentProcessor()**, **centralFunctions.octaveIntegration()**, **centralFunctions.pythonIntegration()**, and **setIntegrationOption()**.

Referenced by **sensitivityAnalyzer.ClassSensitivityAnalyzer.calculateSens()**.

6.24.1.12 plotMaker()

```
simulator.plotMaker (
              outputfiles,
              grid,
              model )
```

```
This function creates the postscript files for the multicellular plots.
```

Referenced by **createEpiplot()**.

6.24.1.13 setIntegrationOption()

```
simulator.setIntegrationOption (
              model,
              tester )
```

```
This function set the integration option depending of model characteristics
and the software installed on the current machine.
```

Referenced by **central()**, **obtainSimulationValues()**, **dynamicsReconstructer.run()**, and **optimiser.scoreObtainer()**.

6.25 simulatorEuler Namespace Reference

Classes

- class **ClassSimulatorEuler**

6.26 simulatorOpenModelica Namespace Reference

Classes

- class **ClassSimulatorOpenModelica**

6.27 simulatorRungeKutta Namespace Reference

Classes

- class **ClassSimulatorRungeKutta**

6.28 simulatorStochastic Namespace Reference

Classes

- class **ClassSimulatorStochastic**

Variables

- **stochasticdir** = os.path.join(os.environ.get('BYODYN_PATH'), 'lib', 'stochastic')

6.28.1 Variable Documentation

6.28.1.1 stochasticdir

```
simulatorStochastic.stochasticdir = os.path.join(os.environ.get('BYODYN_PATH'), 'lib', 'stochastic')
```

6.29 simulatorStochasticTables Namespace Reference

Variables

- dict **ButcherTableau**

6.29.1 Variable Documentation

6.29.1.1 ButcherTableau

```
dict simulatorStochasticTables.ButcherTableau
```

Initial value:

```
00001 =  {\
00002 'tauLeap': {'f':1., 'c':[.0], 'b':[1.], 'a':[[]]},\
00003 'RK2': {'f':1., 'c':[.0, .5], 'b':[.0, 1.], 'a':[[.5]]},\
00004 'RK4': {'f':1., 'c':[.0, .5, .5, 1.], 'b':[1./6., 1./3., 1./3., 1./6.], 'a':[[.5], [.0, .5], [.0, .0,
      1.]]},\
00005 'CHEB_1': {'f':1.0,'c':[0.0], 'b':[1.0], 'a':[ ]},\
00006 'DAMP_1_0.000': {'f':1.0,'c':[0.0], 'b':[1.0], 'a':[ ]},\
00007 'CHEB_2': {'f':0.25,'c':[0.0,0.25], 'b':[0.5,0.5], 'a':[[0.25] ]},\
00008 'DAMP_2_0.000': {'f':0.25,'c':[0.0,0.25], 'b':[0.5,0.5], 'a':[[0.25] ]},\
00009 'DAMP_2_0.365': {'f':0.316529925544,'c':[0.0,0.29006178744], 'b':[0.5,0.5], 'a':[[0.29006178744] ]},\
00010 'DAMP_2_1.460': {'f':0.49934981685,'c':[0.0,0.365824041648], 'b':[0.5,0.5], 'a':[[0.365824041648] ]},\
00011 'CHEB_3': {'f':0.111111111111,'c':[0.0,0.111111111111,0.444444444444],
      'b':[0.333333333333,0.444444444444,0.222222222222],
      'a':[[0.111111111111],[0.222222222222,0.222222222222] ]},\
00012 'DAMP_3_0.000': {'f':0.111111111111,'c':[0.0,0.111111111111,0.444444444444],
      'b':[0.333333333333,0.444444444444,0.222222222222],
      'a':[[0.111111111111],[0.222222222222,0.222222222222] ]},\
00013 'DAMP_3_0.244': {'f':0.129706668705,'c':[0.0,0.12628299636,0.480120457268],
      'b':[0.333333333333,0.43685850182,0.229808164847],
      'a':[[0.12628299636],[0.240060228634,0.240060228634] ]},\
00014 'DAMP_3_0.419': {'f':0.142498656358,'c':[0.0,0.136159667398,0.50105203779],
      'b':[0.333333333333,0.431920166301,0.234746500366],
      'a':[[0.136159667398],[0.250526018895,0.250526018895] ]},\
00015 'DAMP_3_0.715': {'f':0.163237731406,'c':[0.0,0.151223837638,0.529783465224],
      'b':[0.333333333333,0.424388081181,0.242278585486],
      'a':[[0.151223837638],[0.264891732612,0.264891732612] ]},\
00016 'CHEB_4': {'f':0.0625,'c':[0.0,0.0625,0.25,0.5625], 'b':[0.25,0.375,0.25,0.125],
      'a':[[0.0625],[0.125,0.125],[0.1875,0.25,0.125] ]},\
00017 'DAMP_4_0.000': {'f':0.0625,'c':[0.0,0.0625,0.25,0.5625], 'b':[0.25,0.375,0.25,0.125],
      'a':[[0.0625],[0.125,0.125],[0.1875,0.25,0.125] ]},\
00018 'DAMP_4_0.140': {'f':0.0684179428626,'c':[0.0,0.0678244786742,0.266691364575,0.583688792183],
      'b':[0.25,0.370755081525,0.25,0.129244918475],
      'a':[[0.0678244786742],[0.133345682288,0.133345682288],[0.194562930728,0.257932156755,0.131193704701]
      ]},\
00019 'DAMP_4_0.270': {'f':0.0737474142271,'c':[0.0,0.0725235788342,0.280850261703,0.600570923159],
      'b':[0.25,0.367033579128,0.25,0.132966420872],
      'a':[[0.0725235788342],[0.140425130851,0.140425130851],[0.20019030772,0.264023672163,0.136356943277]
      ]}
00020 }
```

6.30 simulatorXPP Namespace Reference

Classes

- class **ClassSimulatorXPP**

Functions

- **convertName** (name)
- **replaceNames** (expr, names, newNames, conflictives)

6.30.1 Function Documentation

6.30.1.1 convertName()

```
simulatorXPP.convertName (
              name )
```

```
This function prepares the format of the names suitable for XPPAUT format.
Mainly string names of more than 7 letters are converted into a new one composed of the first 5 and the last
Also underscores are removed.
```

6.30.1.2 replaceNames()

```
simulatorXPP.replaceNames (
              expr,
              names,
              newNames,
              conflictives )
```

This function replace from an expression the original names by the XPPAUT-adequate format.

6.31 starter Namespace Reference

Classes

- class **ClassMetaModel**

Functions

- **central** (runnerFile)
- **fortranModulesCreator** (metamodel)
- **incompatibilityChecker** (metamodel)

6.31.1 Function Documentation

6.31.1.1 central()

```
starter.central (
              runnerFile )
```

This function directs the reading of the running options of ByoDyn.
Then it checks for incompatibilities.

References **fortranModulesCreator()**, and **incompatibilityChecker()**.

Referenced by **central.optionReader()**.

6.31.1.2 fortranModulesCreator()

```
starter.fortranModulesCreator (
              metamodel )
```

This function builds a module that can be called from Python from Fortran.
It must be compiled now because there are some functions that are static and some arguments have to be declare

Referenced by **central()**.

6.31.1.3 incompatibilityChecker()

```
starter.incompatibilityChecker (
              metamodel )
```

```
This function checks that the options of the metamodel do not contain incompatibilities.
```

Referenced by **central()**.

6.32 surface Namespace Reference

Functions

- **central** (metamodel, model, outputfiles)
- **checkSurfaceParameters** (metamodel, model)
- **rainbowColorCalculation** (value, minValue, maxValue)
- **surfacePlotter** (metamodel, outputfiles, plot, surface, xGrid, yGrid)
- **surfaceTextSaver** (outputfiles, metamodel, plot, surface)

6.32.1 Function Documentation

6.32.1.1 central()

```
surface.central (
              metamodel,
              model,
              outputfiles )
```

```
This function directs the creation of the surface of fitness function values.
It checks that the selected parameters are in model.
It variates the parameters and calls the scoreObtainer function from the
Optimiser module to calculate the fitness function value.
```

References **checkSurfaceParameters()**, **optimiser.scoreObtainer()**, **surfacePlotter()**, and **surfaceText↩ Saver()**.

Referenced by **central.runner()**.

6.32.1.2 checkSurfaceParameters()

```
surface.checkSurfaceParameters (
              metamodel,
              model )
```

```
This function checks that the selected parameters for the surface plot do exist in the model.
```

Referenced by **central()**.

6.32.1.3 rainbowColorCalculation()

```
surface.rainbowColorCalculation (
              value,
              minValue,
              maxValue )
```

```
This function calculates the color based on a rainbow scale
```

Referenced by **surfacePlotter()**.

6.32.1.4 surfacePlotter()

```
surface.surfacePlotter (
              metamodel,
              outputfiles,
              plot,
              surface,
              xGrid,
              yGrid )
```

```
This function creates the postscript surface plot.
```

References **rainbowColorCalculation()**.

Referenced by **central()**.

6.32.1.5 surfaceTextSaver()

```
surface.surfaceTextSaver (
              outputfiles,
              metamodel,
              plot,
              surface )
```

```
This function writes in a text file the results of the fitness function evaluation along the grid.
```

Referenced by **central()**.

6.33 tagParser Namespace Reference

Classes

- class **ClassModelTags**

6.34 tauleap Namespace Reference

Functions

- **simulate** (evalPropensities, stoichiometry, x_0, time, tau, seed=None)

6.34.1 Function Documentation

6.34.1.1 simulate()

```
tauleap.simulate (
              evalPropensities,
              stoichiometry,
              x_0,
              time,
              tau,
              seed = None )
```

```
Gillespie's Poisson tau-leap stochastic simulation algorithm
    evalPropensities  list of propensity functions
    stoichiometry     stoichiometrty matrix, each column corresponds to a reaction
    x_0               initial conditions
    time              simulation time
    tau               the time step for which the system state is written (if set
                      to None all reactions are written in the output)
    seed              for the pseudo-random number generator
```

6.35 testers Namespace Reference

Classes

- class **ClassCluster2DTest**
- class **ClassCluster3DTest**
- class **ClassExportingTest**
- class **ClassFigureFormatTest**
- class **ClassFitnessFunctionCalculationTest**
- class **ClassFitnessFunctionSurfaceTest**
- class **ClassGeneticAlgorithmTest**
- class **ClassHybridOnePhaseTest**
- class **ClassHybridTwoPhasesTest**
- class **ClassIdentifiabilityTest**
- class **ClassLocalSearchOptimisationTest**
- class **ClassNumberOfSimulationsStopperTest**
- class **ClassOEDTest**
- class **ClassOptionalOutputFormatTest**
- class **ClassPlotKeysTest**
- class **ClassRandomSearchOptimisationTest**
- class **ClassSciPyTest**
- class **ClassScoreStopperTest**
- class **ClassSensitivityTest**
- class **ClassSeparatedGraphsTest**
- class **ClassSimulationMethodsTest**

- class **ClassStochasticGeneralTest**
- class **ClassStochasticLastStateTest**
- class **ClassStochasticSeparatedGraphsTest**
- class **ClassStochasticSingleFigureTest**
- class **ClassTagFormatTest**
- class **ClassTest**
- class **ClassTrajectoriesReconstructionTest**
- class **ClassWithoutGraphicsTest**

Chapter 7

Class Documentation

7.1 errorMessages.ClassByoDynException Class Reference

Inheritance diagram for errorMessages.ClassByoDynException:

7.2 errorMessages.ClassCentralException Class Reference

Inheritance diagram for errorMessages.ClassCentralException:

Collaboration diagram for errorMessages.ClassCentralException:

Public Member Functions

- **printExceptionInfo** (self)

Public Member Functions inherited from errorMessages.ClassByoDynException

- **__init__** (self, **errorString**)

Additional Inherited Members

Public Attributes inherited from errorMessages.ClassByoDynException

- **errorString**

7.2.1 Detailed Description

```
This class deals with the specific errors of the central module.
```

7.2.2 Member Function Documentation

7.2.2.1 printExceptionInfo()

```
errorMessages.ClassCentralException.printExceptionInfo (
              self )
```

```
This method prints the specific error string corresponding to the module and then the specific
```

Reimplemented from **errorMessages.ClassByoDynException** (p. **??**).

The documentation for this class was generated from the following file:

- **errorMessages.py**

7.3 checker.ClassChecker Class Reference

Public Member Functions

- **__init__** (self)
- **chooser** (self)
- **detector** (self)
- **run** (self)

Public Attributes

- **listOfTests**
- **log**
- **logFile**
- **OctaveFound**
- **OpenModelicaFound**
- **ScipyFound**
- **XPPFound**

Private Member Functions

- **__printLog** (self, summaryMessage)
- **__writeLogFile** (self)

7.3.1 Detailed Description

```
Class for the checking of the program.
```

7.3.2 Constructor & Destructor Documentation

7.3.2.1 __init__()

```
checker.ClassChecker.__init__ (
              self )
```

The constructor.

7.3.3 Member Function Documentation

7.3.3.1 __printLog()

```
checker.ClassChecker.__printLog (
              self,
              summaryMessage )  [private]
```

This method prints the results of the tests.

References **checker.ClassChecker.logFile**.

Referenced by **checker.ClassChecker.run()**.

7.3.3.2 __writeLogFile()

```
checker.ClassChecker.__writeLogFile (
              self )  [private]
```

This method saves the results from the tests to a file.

References **checker.ClassChecker.listOfTests**, **checker.ClassChecker.log**, and **checker.ClassChecker.log↩File**.

Referenced by **checker.ClassChecker.run()**.

7.3.3.3 chooser()

```
checker.ClassChecker.chooser (
              self )
```

This method selects the appropriate tests to run depending on the machine software'

References **checker.ClassChecker.listOfTests**, **checker.ClassChecker.OctaveFound**, **checker.Class↩Checker.OpenModelicaFound**, **checker.ClassChecker.ScipyFound**, and **checker.ClassChecker.XPPFound**.

7.3.3.4 detector()

```
checker.ClassChecker.detector (
              self )
```

```
        This method detects if scipy, octave, xpp and openModelical are available on the current machine.
```

References **checker.ClassChecker.OctaveFound**, **checker.ClassChecker.OpenModelicaFound**, **checker.ClassChecker.ScipyFound**, and **checker.ClassChecker.XPPFound**.

7.3.3.5 run()

```
checker.ClassChecker.run (
              self )
```

```
        This method runs the available tests.
```

References **checker.ClassChecker.__printLog()**, **checker.ClassChecker.__writeLogFile()**, **checker.ClassChecker.listOfTests**, and **checker.ClassChecker.log**.

7.3.4 Member Data Documentation

7.3.4.1 listOfTests

```
checker.ClassChecker.listOfTests
```

Referenced by **checker.ClassChecker.__writeLogFile()**, **checker.ClassChecker.chooser()**, and **checker.ClassChecker.run()**.

7.3.4.2 log

```
checker.ClassChecker.log
```

Referenced by **checker.ClassChecker.__writeLogFile()**, and **checker.ClassChecker.run()**.

7.3.4.3 logFile

```
checker.ClassChecker.logFile
```

Referenced by **checker.ClassChecker.__printLog()**, and **checker.ClassChecker.__writeLogFile()**.

7.3.4.4 OctaveFound

```
checker.ClassChecker.OctaveFound
```

Referenced by **checker.ClassChecker.chooser()**, and **checker.ClassChecker.detector()**.

7.3.4.5 OpenModelicaFound

`checker.ClassChecker.OpenModelicaFound`

Referenced by **checker.ClassChecker.chooser()**, and **checker.ClassChecker.detector()**.

7.3.4.6 ScipyFound

`checker.ClassChecker.ScipyFound`

Referenced by **checker.ClassChecker.chooser()**, and **checker.ClassChecker.detector()**.

7.3.4.7 XPPFound

`checker.ClassChecker.XPPFound`

Referenced by **checker.ClassChecker.chooser()**, and **checker.ClassChecker.detector()**.

The documentation for this class was generated from the following file:

- **checker.py**

7.4 errorMessages.ClassCheckerException Class Reference

Inheritance diagram for errorMessages.ClassCheckerException:

Collaboration diagram for errorMessages.ClassCheckerException:

Public Member Functions

- **printExceptionInfo** (self)

Public Member Functions inherited from errorMessages.ClassByoDynException

- **__init__** (self, **errorString**)

Additional Inherited Members

Public Attributes inherited from errorMessages.ClassByoDynException

- **errorString**

7.4.1 Detailed Description

```
This class deals with the specific errors of the checker module from benchmark directory.
```

7.4.2 Member Function Documentation

7.4.2.1 printExceptionInfo()

```
errorMessages.ClassCheckerException.printExceptionInfo (
              self )

                This method prints the specific error string corresponding to the module and then the specific
```

Reimplemented from **errorMessages.ClassByoDynException** (p. **??**).

The documentation for this class was generated from the following file:

- **errorMessages.py**

7.5 testers.ClassCluster2DTest Class Reference

Inheritance diagram for testers.ClassCluster2DTest:

Collaboration diagram for testers.ClassCluster2DTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Additional Inherited Members

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.5.1 Detailed Description

```
Class for testing the 2D clustering.
```

7.5.2 Member Function Documentation

7.5.2.1 analyseResults()

```
testers.ClassCluster2DTest.analyseResults (
              self )
```

```
        This method analyses the results of the test for the 2D clustering.
```

References **testers.ClassTest.obtainedResult**, **testers.ClassFitnessFunctionSurfaceTest.obtained↩ Result**, **testers.ClassOEDTest.obtainedResult**, **testers.ClassSciPyTest.obtainedResult**, **testers.Class↩ SensitivityTest.obtainedResult**, **testers.ClassTagFormatTest.obtainedResult**, **testers.ClassTrajectories↩ ReconstructionTest.obtainedResult**, and **testers.ClassWithoutGraphicsTest.obtainedResult**.

Referenced by **testers.ClassTest.run()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.6 testers.ClassCluster3DTest Class Reference

Inheritance diagram for testers.ClassCluster3DTest:

Collaboration diagram for testers.ClassCluster3DTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Additional Inherited Members

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.6.1 Detailed Description

```
Class for testing the 3D clustering.
```

7.6.2 Member Function Documentation

7.6.2.1 analyseResults()

```
testers.ClassCluster3DTest.analyseResults (
              self )
```

```
        This method analyses the results of the test for the 3D clustering.
```

References **testers.ClassTest.obtainedResult**, **testers.ClassFitnessFunctionSurfaceTest.obtained↩ Result**, **testers.ClassOEDTest.obtainedResult**, **testers.ClassSciPyTest.obtainedResult**, **testers.Class↩ SensitivityTest.obtainedResult**, **testers.ClassTagFormatTest.obtainedResult**, **testers.ClassTrajectories↩ ReconstructionTest.obtainedResult**, and **testers.ClassWithoutGraphicsTest.obtainedResult**.

Referenced by **testers.ClassTest.run()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.7 errorMessages.ClassClusterException Class Reference

Inheritance diagram for errorMessages.ClassClusterException:

Collaboration diagram for errorMessages.ClassClusterException:

Public Member Functions

- **printExceptionInfo** (self)

Public Member Functions inherited from errorMessages.ClassByoDynException

- **__init__** (self, **errorString**)

Additional Inherited Members

Public Attributes inherited from errorMessages.ClassByoDynException

- **errorString**

7.7.1 Detailed Description

```
This class deals with the specific errors of the cluster module.
```

7.7.2 Member Function Documentation

7.7.2.1 printExceptionInfo()

```
errorMessages.ClassClusterException.printExceptionInfo (
             self )
```

```
This method prints the specific error string corresponding to the module and then the specific
```

Reimplemented from **errorMessages.ClassByoDynException** (p. **??**).

The documentation for this class was generated from the following file:

- **errorMessages.py**

7.8 errorMessages.ClassDynamicsReconstructerException Class Reference

Inheritance diagram for errorMessages.ClassDynamicsReconstructerException:

Collaboration diagram for errorMessages.ClassDynamicsReconstructerException:

Public Member Functions

- **printExceptionInfo** (self)

Public Member Functions inherited from errorMessages.ClassByoDynException

- **__init__** (self, **errorString**)

Additional Inherited Members

Public Attributes inherited from errorMessages.ClassByoDynException

- **errorString**

7.8.1 Detailed Description

```
This class deals with the specific errors of the dynamicsReconstructer module.
```

7.8.2 Member Function Documentation

7.8.2.1 printExceptionInfo()

```
errorMessages.ClassDynamicsReconstructerException.printExceptionInfo (
              self )
```

```
This method prints the specific error string corresponding to the module and then the specific
```

Reimplemented from **errorMessages.ClassByoDynException** (p. **??**).

The documentation for this class was generated from the following file:

- **errorMessages.py**

7.9 simulator.ClassEpithelium Class Reference

Public Member Functions

- **__init__** (self)
- **dataObtainer** (grid, model, outputfiles)

Public Attributes

- **color**
- **concentration**
- **length**
- **nodes**
- **time**
- **width**

7.9.1 Detailed Description

```
Class for the multicellular plots.
```

7.9.2 Constructor & Destructor Documentation

7.9.2.1 __init__()

```
simulator.ClassEpithelium.__init__ (
              self )
```

```
        This is the constructer.
```

7.9.3 Member Function Documentation

7.9.3.1 dataObtainer()

```
simulator.ClassEpithelium.dataObtainer (
              grid,
              model,
              outputfiles )
```

```
        This method initialises the object of the class ClassEpithelium.
        It takes the values from different sources, either the object "model" or from the simulationResults fi
```

7.9.4 Member Data Documentation

7.9.4.1 color

```
simulator.ClassEpithelium.color
```

7.9.4.2 concentration

```
simulator.ClassEpithelium.concentration
```

7.9.4.3 length

```
simulator.ClassEpithelium.length
```

7.9.4.4 nodes

```
simulator.ClassEpithelium.nodes
```

Referenced by **sbmlWorker.ClassModelSBML.__putRateruleInTopology()**, **sbmlWorker.ClassModel↩ SBML.__readNodes()**, **sbmlWorker.ClassModelSBML.__readTopology()**, **sbmlWorker.ClassModelSBML.↩ checkAssignmentRule()**, **sbmlWorker.ClassModelSBML.checkCompatibilities()**, **sbmlWorker.Class↩ ModelSBML.getDefaultValue()**, **tagParser.ClassModelTags.readInput()**, **sbmlWorker.ClassModelSBML.↩ summary()**, and **tagParser.ClassModelTags.summary()**.

7.9.4.5 time

```
simulator.ClassEpithelium.time
```

7.9.4.6 width

```
simulator.ClassEpithelium.width
```

The documentation for this class was generated from the following file:

- **simulator.py**

7.10 sbmlWorker.ClassEvent Class Reference

Public Member Functions

- **__init__** (self)

Public Attributes

- **assignment**
- **assignmentAST**
- **delay**
- **id**
- **trigger**

7.10.1 Detailed Description

```
Class for SBML events.
```

7.10.2 Constructor & Destructor Documentation

7.10.2.1 __init__()

```
sbmlWorker.ClassEvent.__init__ (
              self )
```

```
        This is the constructor.
```

7.10.3 Member Data Documentation

7.10.3.1 assignment

```
sbmlWorker.ClassEvent.assignment
```

7.10.3.2 assignmentAST

```
sbmlWorker.ClassEvent.assignmentAST
```

7.10.3.3 delay

```
sbmlWorker.ClassEvent.delay
```

7.10.3.4 id

```
sbmlWorker.ClassEvent.id
```

Referenced by **testers.ClassTest.__execute()**.

7.10.3.5 trigger

```
sbmlWorker.ClassEvent.trigger
```

The documentation for this class was generated from the following file:

- **sbmlWorker.py**

7.11 testers.ClassExportingTest Class Reference

Inheritance diagram for testers.ClassExportingTest:

Collaboration diagram for testers.ClassExportingTest:

Public Member Functions

- **__init__** (self)
- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Public Attributes

- **id**
- **runner**

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.11.1 Detailed Description

```
Class for testing exporting.
```

7.11.2 Constructor & Destructor Documentation

7.11.2.1 __init__()

```
testers.ClassExportingTest.__init__ (
              self )

        The constructor.
```

Reimplemented from **testers.ClassTest** (p. 180).

7.11.3 Member Function Documentation

7.11.3.1 analyseResults()

```
testers.ClassExportingTest.analyseResults (
              self )

        This method analyses the results of the test for exporting.
```

Referenced by **testers.ClassTest.run()**.

7.11.4 Member Data Documentation

7.11.4.1 id

```
testers.ClassExportingTest.id
```

Referenced by **testers.ClassTest.__execute()**.

7.11.4.2 runner

```
testers.ClassExportingTest.runner
```

Referenced by **testers.ClassTest.__execute()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.12 testers.ClassFigureFormatTest Class Reference

Inheritance diagram for testers.ClassFigureFormatTest:

Collaboration diagram for testers.ClassFigureFormatTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Additional Inherited Members

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.12.1 Detailed Description

```
Class for testing the output format of graphics.
```

7.12.2 Member Function Documentation

7.12.2.1 analyseResults()

```
testers.ClassFigureFormatTest.analyseResults (
              self )

        This method analyses the format of the output graphics.
```

References **testers.ClassTest.obtainedResult**, **testers.ClassFitnessFunctionSurfaceTest.obtained↩Result**, **testers.ClassOEDTest.obtainedResult**, **testers.ClassSciPyTest.obtainedResult**, **testers.Class↩SensitivityTest.obtainedResult**, **testers.ClassTagFormatTest.obtainedResult**, **testers.ClassTrajectories↩ReconstructionTest.obtainedResult**, and **testers.ClassWithoutGraphicsTest.obtainedResult**.

Referenced by **testers.ClassTest.run()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.13 central.ClassFile Class Reference

7.13.1 Detailed Description

```
This class specifies the paths for all required files that ByoDyn uses, both input and output files.
```

The documentation for this class was generated from the following file:

- **central.py**

7.14 testers.ClassFitnessFunctionCalculationTest Class Reference

Inheritance diagram for testers.ClassFitnessFunctionCalculationTest:

Collaboration diagram for testers.ClassFitnessFunctionCalculationTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Additional Inherited Members

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.14.1 Detailed Description

```
Class for testing the fitness function calculation functionality.
```

7.14.2 Member Function Documentation

7.14.2.1 analyseResults()

```
testers.ClassFitnessFunctionCalculationTest.analyseResults (
              self )

        This method analyses for the result of the fitness function calculation.
```

References **testers.ClassTest.obtainedResult**, **testers.ClassFitnessFunctionSurfaceTest.obtained↩ Result**, **testers.ClassOEDTest.obtainedResult**, **testers.ClassSciPyTest.obtainedResult**, **testers.Class↩ SensitivityTest.obtainedResult**, **testers.ClassTagFormatTest.obtainedResult**, **testers.ClassTrajectories↩ ReconstructionTest.obtainedResult**, and **testers.ClassWithoutGraphicsTest.obtainedResult**.

Referenced by **testers.ClassTest.run()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.15 testers.ClassFitnessFunctionSurfaceTest Class Reference

Inheritance diagram for testers.ClassFitnessFunctionSurfaceTest:

Collaboration diagram for testers.ClassFitnessFunctionSurfaceTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Public Attributes

- **obtainedResult**

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.15.1 Detailed Description

```
Class for testing the fitness function surface functionality.
```

7.15.2 Member Function Documentation

7.15.2.1 analyseResults()

```
testers.ClassFitnessFunctionSurfaceTest.analyseResults (
              self )

        This method analyses the results of the fitness function surface functionality.
```

References **testers.ClassTest.obtainedResult**, **testers.ClassFitnessFunctionSurfaceTest.obtained↩Result**, **testers.ClassOEDTest.obtainedResult**, **testers.ClassSciPyTest.obtainedResult**, **testers.Class↩SensitivityTest.obtainedResult**, **testers.ClassTagFormatTest.obtainedResult**, **testers.ClassTrajectories↩ReconstructionTest.obtainedResult**, and **testers.ClassWithoutGraphicsTest.obtainedResult**.

Referenced by **testers.ClassTest.run()**.

7.15.3 Member Data Documentation

7.15.3.1 obtainedResult

```
testers.ClassFitnessFunctionSurfaceTest.obtainedResult
```

Referenced by **testers.ClassCluster2DTest.analyseResults()**, **testers.ClassCluster3DTest.analyseResults()**, **testers.ClassFigureFormatTest.analyseResults()**, **testers.ClassFitnessFunctionCalculationTest.analyse↩Results()**, **testers.ClassFitnessFunctionSurfaceTest.analyseResults()**, **testers.ClassGeneticAlgorithm↩Test.analyseResults()**, **testers.ClassHybridOnePhaseTest.analyseResults()**, **testers.ClassHybridTwo↩PhasesTest.analyseResults()**, **testers.ClassIdentifiabilityTest.analyseResults()**, **testers.ClassLocal↩SearchOptimisationTest.analyseResults()**, **testers.ClassNumberOfSimulationsStopperTest.analyse↩Results()**, **testers.ClassOEDTest.analyseResults()**, **testers.ClassOptionalOutputFormatTest.analyse↩Results()**, **testers.ClassPlotKeysTest.analyseResults()**, **testers.ClassRandomSearchOptimisationTest.↩analyseResults()**, **testers.ClassSciPyTest.analyseResults()**, **testers.ClassScoreStopperTest.analyse↩Results()**, **testers.ClassSensitivityTest.analyseResults()**, **testers.ClassStochasticGeneralTest.analyse↩Results()**, **testers.ClassStochasticLastStateTest.analyseResults()**, **testers.ClassStochasticSeparated↩GraphsTest.analyseResults()**, **testers.ClassStochasticSingleFigureTest.analyseResults()**, **testers.↩ClassTagFormatTest.analyseResults()**, **testers.ClassTrajectoriesReconstructionTest.analyseResults()**, and **testers.ClassTest.dataFilesChecker()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.16 errorMessages.ClassFormulasException Class Reference

Inheritance diagram for errorMessages.ClassFormulasException:

Collaboration diagram for errorMessages.ClassFormulasException:

Public Member Functions

- **printExceptionInfo** (self)

Public Member Functions inherited from errorMessages.ClassByoDynException

- **__init__** (self, **errorString**)

Additional Inherited Members

Public Attributes inherited from errorMessages.ClassByoDynException

- **errorString**

7.16.1 Detailed Description

```
This class deals with the specific errors of the formulas module of the library directory.
```

7.16.2 Member Function Documentation

7.16.2.1 printExceptionInfo()

```
errorMessages.ClassFormulasException.printExceptionInfo (
              self )
```

```
This method prints the specific error string corresponding to the module and then the specific
```

Reimplemented from **errorMessages.ClassByoDynException** (p. **??**).

The documentation for this class was generated from the following file:

- **errorMessages.py**

7.17 sbmlWorker.ClassFunction Class Reference

Public Member Functions

- **__init__** (self)

Public Attributes

- **arguments**
- **id**
- **mathAST**
- **output**

7.17.1 Detailed Description

Class for SBML function definitions.

7.17.2 Constructor & Destructor Documentation

7.17.2.1 __init__()

```
sbmlWorker.ClassFunction.__init__ (
              self )
```

This is the constructor.

7.17.3 Member Data Documentation

7.17.3.1 arguments

sbmlWorker.ClassFunction.arguments

7.17.3.2 id

sbmlWorker.ClassFunction.id

Referenced by **testers.ClassTest.__execute()**.

7.17.3.3 mathAST

sbmlWorker.ClassFunction.mathAST

7.17.3.4 output

sbmlWorker.ClassFunction.output

The documentation for this class was generated from the following file:

- **sbmlWorker.py**

7.18 testers.ClassGeneticAlgorithmTest Class Reference

Inheritance diagram for testers.ClassGeneticAlgorithmTest:

Collaboration diagram for testers.ClassGeneticAlgorithmTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Additional Inherited Members

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.18.1 Detailed Description

```
Class for testing the results of the genetic algorithm.
```

7.18.2 Member Function Documentation

7.18.2.1 analyseResults()

```
testers.ClassGeneticAlgorithmTest.analyseResults (
              self )
```

```
        This method analyses the results of the genetic algorithm.
        It checks that the number of iteration is 5, that the best value of the iterations are getting lower a
```

References **testers.ClassTest.obtainedResult**, **testers.ClassFitnessFunctionSurfaceTest.obtained↩Result**, **testers.ClassOEDTest.obtainedResult**, **testers.ClassSciPyTest.obtainedResult**, **testers.Class↩SensitivityTest.obtainedResult**, **testers.ClassTagFormatTest.obtainedResult**, **testers.ClassTrajectories↩ReconstructionTest.obtainedResult**, and **testers.ClassWithoutGraphicsTest.obtainedResult**.

Referenced by **testers.ClassTest.run()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.19 testers.ClassHybridOnePhaseTest Class Reference

Inheritance diagram for testers.ClassHybridOnePhaseTest:

Collaboration diagram for testers.ClassHybridOnePhaseTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Additional Inherited Members

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.19.1 Detailed Description

```
Class for testing the results of the hybrid one phase optimisation.
```

7.19.2 Member Function Documentation

7.19.2.1 analyseResults()

```
testers.ClassHybridOnePhaseTest.analyseResults (
              self )

        This method analyses the results of the hybrid one phase optimisation.
```

References **testers.ClassTest.obtainedResult**, **testers.ClassFitnessFunctionSurfaceTest.obtained↩Result**, **testers.ClassOEDTest.obtainedResult**, **testers.ClassSciPyTest.obtainedResult**, **testers.Class↩SensitivityTest.obtainedResult**, **testers.ClassTagFormatTest.obtainedResult**, **testers.ClassTrajectories↩ReconstructionTest.obtainedResult**, and **testers.ClassWithoutGraphicsTest.obtainedResult**.

Referenced by **testers.ClassTest.run()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.20 testers.ClassHybridTwoPhasesTest Class Reference

Inheritance diagram for testers.ClassHybridTwoPhasesTest:

Collaboration diagram for testers.ClassHybridTwoPhasesTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Additional Inherited Members

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.20.1 Detailed Description

```
Class for testing the results of the hybrid two phases optimisation.
```

7.20.2 Member Function Documentation

7.20.2.1 analyseResults()

```
testers.ClassHybridTwoPhasesTest.analyseResults (
              self )

        This method analyses the results of the hybrid two phases optimisation.
```

References **testers.ClassTest.obtainedResult**, **testers.ClassFitnessFunctionSurfaceTest.obtained↩Result**, **testers.ClassOEDTest.obtainedResult**, **testers.ClassSciPyTest.obtainedResult**, **testers.Class↩SensitivityTest.obtainedResult**, **testers.ClassTagFormatTest.obtainedResult**, **testers.ClassTrajectories↩ReconstructionTest.obtainedResult**, and **testers.ClassWithoutGraphicsTest.obtainedResult**.

Referenced by **testers.ClassTest.run()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.21 identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer Class Reference

Public Member Functions

- **__init__** (self)
- **calculateFIM** (self, parametersToStudy, metamodelTarget)
- **confidenceIntervals** (self, i, threshold)
- **criteriaWriter** (self, outputfiles, MA, D, E, ME)
- **getDcriteria** (self)
- **getEcriteria** (self)
- **getMAcriteria** (self)
- **getMEcriteria** (self)
- **residualsComputation** (expValues, model, metamodel, outputfiles)
- **setCorrelationMatrix** (self, parametersToStudy)
- **setCOV** (self, parametersToStudy)
- **setSensitivity** (self, coefficients)

Public Attributes

- **CorrelationMatrix**
- **COV**
- **FIM**
- **sensitivityCoefficients**

7.21.1 Detailed Description

```
Class for the identifiability analysis.
```

7.21.2 Constructor & Destructor Documentation

7.21.2.1 __init__()

```
identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.__init__ (
              self )
```

```
This is the constructor.
```

7.21.3 Member Function Documentation

7.21.3.1 calculateFIM()

```
identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.calculateFIM (
              self,
              parametersToStudy,
              metamodelTarget )
```

```
This method calculates of Fisher Information Matrix (FIM) from sensitivity coefficients.
```

7.21.3.2 confidenceIntervals()

```
identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.confidenceIntervals (
              self,
              i,
              threshold )
```

This method calculates the confidence interval for one parameter.

References **identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.COV**, and **pca.ClassPCA.COV**.

7.21.3.3 criteriaWriter()

```
identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.criteriaWriter (
              self,
              outputfiles,
              MA,
              D,
              E,
              ME )
```

This method write the criteria information in its output file.

7.21.3.4 getDcriteria()

```
identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.getDcriteria (
              self )
```

This method returns the D-optimal design criteria, which it is to maximise the determinant of FIM.

References **identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.FIM**.

7.21.3.5 getEcriteria()

```
identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.getEcriteria (
              self )
```

This method returns the E-optimal design criteria, which it is to maximise the lowest FIM eigenvalue.

References **identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.FIM**.

7.21.3.6 getMAcriteria()

```
identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.getMAcriteria (
              self )
```

This method returns the modified A-optimal design criteria, which it is to maximise the trace of FIM.

References **identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.FIM**.

7.21.3.7 getMEcriteria()

```
identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.getMEcriteria (
              self )
```

This method returns the modified E-optimal design criteria, which it is to minimise the ratio among the larges

References **identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.FIM**.

7.21.3.8 residualsComputation()

```
identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.residualsComputation (
              expValues,
              model,
              metamodel,
              outputfiles )
```

This method calculates the residual for one experimental value.

7.21.3.9 setCorrelationMatrix()

```
identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.setCorrelationMatrix (
              self,
              parametersToStudy )
```

This method calculate the Correlation Marix from the COV.

References **identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.CorrelationMatrix**, **identifiabilityAnalyzer.↩ ClassIdentifiabilityAnalyzer.COV**, **pca.ClassPCA.COV**, **pca.ClassPCA.setCOV()**, and **identifiability↩ Analyzer.ClassIdentifiabilityAnalyzer.setCOV()**.

7.21.3.10 setCOV()

```
identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.setCOV (
              self,
              parametersToStudy )
```

This mehtod set the covariance Matrix and calculates its EigenValues, it is the inverse of FIM.

Referenced by **identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.setCorrelationMatrix()**.

7.21.3.11 setSensitivity()

```
identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.setSensitivity (
              self,
              coefficients )
```

This method set the sensitivity coefficients.

References **identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.sensitivityCoefficients**.

7.21.4 Member Data Documentation

7.21.4.1 CorrelationMatrix

```
identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.CorrelationMatrix
```

Referenced by **identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.setCorrelationMatrix()**.

7.21.4.2 COV

```
identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.COV
```

Referenced by **identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.confidenceIntervals()**, **identifiability↩ Analyzer.ClassIdentifiabilityAnalyzer.setCorrelationMatrix()**, **pca.ClassPCA.setCOV()**, and **pca.Class↩ PCA.setEigen()**.

7.21.4.3 FIM

```
identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.FIM
```

Referenced by **identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.getDcriteria()**, **identifiabilityAnalyzer.↩ ClassIdentifiabilityAnalyzer.getEcriteria()**, **identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.getMAcriteria()**, and **identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.getMEcriteria()**.

7.21.4.4 sensitivityCoefficients

```
identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.sensitivityCoefficients
```

Referenced by **identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.setSensitivity()**.

The documentation for this class was generated from the following file:

- **identifiabilityAnalyzer.py**

7.22 errorMessages.ClassIdentifiabilityAnalyzerException Class Reference

Inheritance diagram for errorMessages.ClassIdentifiabilityAnalyzerException:

Collaboration diagram for errorMessages.ClassIdentifiabilityAnalyzerException:

Public Member Functions

- **printExceptionInfo** (self)

Public Member Functions inherited from errorMessages.ClassByoDynException

- **__init__** (self, **errorString**)

Additional Inherited Members

Public Attributes inherited from errorMessages.ClassByoDynException

- **errorString**

7.22.1 Detailed Description

```
This class deals with the specific errors of the identifiabilityAnalyzer module.
```

7.22.2 Member Function Documentation

7.22.2.1 printExceptionInfo()

```
errorMessages.ClassIdentifiabilityAnalyzerException.printExceptionInfo (
              self )
```

```
This method prints the specific error string corresponding to the module and then the specific
```

Reimplemented from **errorMessages.ClassByoDynException** (p. **??**).

The documentation for this class was generated from the following file:

- **errorMessages.py**

7.23 testers.ClassIdentifiabilityTest Class Reference

Inheritance diagram for testers.ClassIdentifiabilityTest:

Collaboration diagram for testers.ClassIdentifiabilityTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Additional Inherited Members

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.23.1 Detailed Description

```
Class for testing the functionalities of identifiability.
```

7.23.2 Member Function Documentation

7.23.2.1 analyseResults()

```
testers.ClassIdentifiabilityTest.analyseResults (
              self )
```

```
        This method analyses the results of the identifiability.
```

References **testers.ClassTest.expectedResult**, **testers.ClassOEDTest.expectedResult**, **testers.ClassSciPyTest.expectedResult**, **testers.ClassTrajectoriesReconstructionTest.expectedResult**, **testers.ClassTest.obtainedResult**, **testers.ClassFitnessFunctionSurfaceTest.obtainedResult**, **testers.ClassOEDTest.obtainedResult**, **testers.ClassSciPyTest.obtainedResult**, **testers.ClassSensitivityTest.obtainedResult**, **testers.ClassTagFormatTest.obtainedResult**, **testers.ClassTrajectoriesReconstructionTest.obtainedResult**, and **testers.ClassWithoutGraphicsTest.obtainedResult**.

Referenced by **testers.ClassTest.run()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.24 errorMessages.ClassInitiatorException Class Reference

Inheritance diagram for errorMessages.ClassInitiatorException:

Collaboration diagram for errorMessages.ClassInitiatorException:

Public Member Functions

- **printExceptionInfo** (self)

Public Member Functions inherited from errorMessages.ClassByoDynException

- **__init__** (self, **errorString**)

Additional Inherited Members

Public Attributes inherited from errorMessages.ClassByoDynException

- **errorString**

7.24.1 Detailed Description

```
This class deals with the specific errors of the initiator module at the bin directory.
```

7.24.2 Member Function Documentation

7.24.2.1 printExceptionInfo()

```
errorMessages.ClassInitiatorException.printExceptionInfo (
              self )
```

```
This method prints the specific error string corresponding to the module and then the specific
```

Reimplemented from **errorMessages.ClassByoDynException** (p. **??**).

The documentation for this class was generated from the following file:

- **errorMessages.py**

7.25 testers.ClassLocalSearchOptimisationTest Class Reference

Inheritance diagram for testers.ClassLocalSearchOptimisationTest:

Collaboration diagram for testers.ClassLocalSearchOptimisationTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Additional Inherited Members

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.25.1 Detailed Description

```
Class for testing the results of local optimisations.
```

7.25.2 Member Function Documentation

7.25.2.1 analyseResults()

```
testers.ClassLocalSearchOptimisationTest.analyseResults (
              self )
```

```
        This method analyses the results of the local optimisations.
```

References **testers.ClassTest.obtainedResult**, **testers.ClassFitnessFunctionSurfaceTest.obtained↩Result**, **testers.ClassOEDTest.obtainedResult**, **testers.ClassSciPyTest.obtainedResult**, **testers.Class↩SensitivityTest.obtainedResult**, **testers.ClassTagFormatTest.obtainedResult**, **testers.ClassTrajectories↩ReconstructionTest.obtainedResult**, and **testers.ClassWithoutGraphicsTest.obtainedResult**.

Referenced by **testers.ClassTest.run()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.26 matrixWorker.ClassMatrix Class Reference

Public Member Functions

- **__init__** (self)
- **blackAndWhiteColorCalculator** (self, value, max)
- **blueRedColorCalculator** (self, value, max)
- **fileWriter** (self, message, outputFile)
- **getCovarianceMatrix** (self)
- **getEmpiricalMean** (self)
- **getMaxEigenValue** (self)
- **getMinEigenValue** (self)
- **getTrace** (self)
- **isSingular** (self)
- **plotWriter** (self, metamodel, outputFile, color)
- **setColNames** (self, names)
- **setEigenValues** (self)
- **setRowNames** (self, names)

Public Attributes

- **colNames**
- **eigenValues**
- **eigenVectors**
- **m**
- **rowNames**

7.26.1 Constructor & Destructor Documentation

7.26.1.1 __init__()

```
matrixWorker.ClassMatrix.__init__ (
              self )
```

Constructor of the class ClassMatrix.

7.26.2 Member Function Documentation

7.26.2.1 blackAndWhiteColorCalculator()

```
matrixWorker.ClassMatrix.blackAndWhiteColorCalculator (
              self,
              value,
              max )
```

This function calculates the colour intensity in the black-white scale.

Referenced by **matrixWorker.ClassMatrix.plotWriter()**.

7.26.2.2 blueRedColorCalculator()

```
matrixWorker.ClassMatrix.blueRedColorCalculator (
              self,
              value,
              max )

        This function calculates the colour intensity in a blue-red scale.
```

Referenced by **matrixWorker.ClassMatrix.plotWriter()**.

7.26.2.3 fileWriter()

```
matrixWorker.ClassMatrix.fileWriter (
              self,
              message,
              outputFile )

        This function writes a text file with a matrix.
```

References **matrixWorker.ClassMatrix.colNames**, **matrixWorker.ClassMatrix.m**, and **matrixWorker.Class↩ Matrix.rowNames**.

7.26.2.4 getCovarianceMatrix()

```
matrixWorker.ClassMatrix.getCovarianceMatrix (
              self )
```

References **matrixWorker.ClassMatrix.m**.

7.26.2.5 getEmpiricalMean()

```
matrixWorker.ClassMatrix.getEmpiricalMean (
              self )
```

References **matrixWorker.ClassMatrix.m**.

7.26.2.6 getMaxEigenValue()

```
matrixWorker.ClassMatrix.getMaxEigenValue (
              self )

        This function returns the largest eigenvalue of a matrix.
```

References **matrixWorker.ClassMatrix.eigenValues**, **pca.ClassPCA.eigenValues**, and **matrixWorker.Class↩ Matrix.setEigenValues()**.

7.26.2.7 getMinEigenValue()

```
matrixWorker.ClassMatrix.getMinEigenValue (
              self )
```

This function returns the lowest eigenvalue of a matrix.

References **matrixWorker.ClassMatrix.eigenValues**, **pca.ClassPCA.eigenValues**, and **matrixWorker.ClassMatrix.setEigenValues()**.

7.26.2.8 getTrace()

```
matrixWorker.ClassMatrix.getTrace (
              self )
```

This function calculates the trace of a matrix: summing up the diagonal.

References **matrixWorker.ClassMatrix.m**.

7.26.2.9 isSingular()

```
matrixWorker.ClassMatrix.isSingular (
              self )
```

This function returns true if the matrix is singular.

References **matrixWorker.ClassMatrix.m**.

7.26.2.10 plotWriter()

```
matrixWorker.ClassMatrix.plotWriter (
              self,
              metamodel,
              outputFile,
              color )
```

This function creates a postscript file with a black and white or color matrix.

References **matrixWorker.ClassMatrix.blackAndWhiteColorCalculator()**, **matrixWorker.ClassMatrix.blueRedColorCalculator()**, **matrixWorker.ClassMatrix.colNames**, **matrixWorker.ClassMatrix.m**, and **matrixWorker.ClassMatrix.rowNames**.

7.26.2.11 setColNames()

```
matrixWorker.ClassMatrix.setColNames (
              self,
              names )
```

```
        This function sets the names of the columns of the matrix.
```

References **matrixWorker.ClassMatrix.colNames**.

7.26.2.12 setEigenValues()

```
matrixWorker.ClassMatrix.setEigenValues (
              self )
```

```
        This function calculates the eigenvalues of a matrix using SciPy.
```

References **matrixWorker.ClassMatrix.eigenValues**, **pca.ClassPCA.eigenValues**, **matrixWorker.Class↩ Matrix.eigenVectors**, **pca.ClassPCA.eigenVectors**, and **matrixWorker.ClassMatrix.m**.

Referenced by **matrixWorker.ClassMatrix.getMaxEigenValue()**, and **matrixWorker.ClassMatrix.getMin↩ EigenValue()**.

7.26.2.13 setRowNames()

```
matrixWorker.ClassMatrix.setRowNames (
              self,
              names )
```

```
        This function sets the names of the rows of the matrix.
```

References **matrixWorker.ClassMatrix.rowNames**.

7.26.3 Member Data Documentation

7.26.3.1 colNames

```
matrixWorker.ClassMatrix.colNames
```

Referenced by **matrixWorker.ClassMatrix.fileWriter()**, **matrixWorker.ClassMatrix.plotWriter()**, and **matrix↩ Worker.ClassMatrix.setColNames()**.

7.26.3.2 eigenValues

```
matrixWorker.ClassMatrix.eigenValues
```

Referenced by **matrixWorker.ClassMatrix.getMaxEigenValue()**, **matrixWorker.ClassMatrix.getMinEigen↩ Value()**, **pca.ClassPCA.setEigen()**, and **matrixWorker.ClassMatrix.setEigenValues()**.

7.26.3.3 eigenVectors

matrixWorker.ClassMatrix.eigenVectors

Referenced by **pca.ClassPCA.setEigen()**, and **matrixWorker.ClassMatrix.setEigenValues()**.

7.26.3.4 m

matrixWorker.ClassMatrix.m

Referenced by **matrixWorker.ClassMatrix.fileWriter()**, **matrixWorker.ClassMatrix.getCovarianceMatrix()**, **matrixWorker.ClassMatrix.getEmpiricalMean()**, **matrixWorker.ClassMatrix.getTrace()**, **matrixWorker.ClassMatrix.isSingular()**, **matrixWorker.ClassMatrix.plotWriter()**, and **matrixWorker.ClassMatrix.setEigenValues()**.

7.26.3.5 rowNames

matrixWorker.ClassMatrix.rowNames

Referenced by **matrixWorker.ClassMatrix.fileWriter()**, **matrixWorker.ClassMatrix.plotWriter()**, and **matrixWorker.ClassMatrix.setRowNames()**.

The documentation for this class was generated from the following file:

- **matrixWorker.py**

7.27 errorMessages.ClassMatrixWorkerException Class Reference

Inheritance diagram for errorMessages.ClassMatrixWorkerException:

Collaboration diagram for errorMessages.ClassMatrixWorkerException:

Public Member Functions

- **printExceptionInfo** (self)

Public Member Functions inherited from errorMessages.ClassByoDynException

- **__init__** (self, **errorString**)

Additional Inherited Members

Public Attributes inherited from errorMessages.ClassByoDynException

- **errorString**

7.27.1 Detailed Description

```
This class deals with the specific errors of the formulas module of the library directory.
```

7.27.2 Member Function Documentation

7.27.2.1 printExceptionInfo()

```
errorMessages.ClassMatrixWorkerException.printExceptionInfo (
              self )

                This method prints the specific error string corresponding to the module and then the specific
```

Reimplemented from **errorMessages.ClassByoDynException** (p. **??**).

The documentation for this class was generated from the following file:

- **errorMessages.py**

7.28 starter.ClassMetaModel Class Reference

Public Member Functions

- **__init__** (self)
- **metamodelParser** (self, optionFile)

Public Attributes

- **bandwidth**
- **clusterInputData**
- **confidenceIntervals**
- **figureFormat**
- **fixedParameters**
- **gaMutationRate**
- **gaPopulation**
- **gaTranslocationRate**
- **graphics**
- **hybridPath**
- **hybridScore**
- **identifiabilityCriteria**
- **identifiabilityOutputs**
- **initialConcentrationsToVary**
- **initialConditionsSolutions**
- **integrationMethod**
- **integrationOption**
- **integrationTolerance**
- **lastIterationScore**
- **modelFile**

- **modelFormat**
- **oedResolution**
- **oedTargetSpecies**
- **oedTask**
- **onlyLastState**
- **optimisationMethod**
- **optionalOutputFormat**
- **parameters**
- **parametersToVary**
- **plottingKeys**
- **RKCustomButcherTableau**
- **RKDampedParameters**
- **runningType**
- **runWithBackup**
- **sampleMethod**
- **sampleSize**
- **sensitivityParameters**
- **separatedGraphs**
- **showingPlot**
- **simulationNumber**
- **simulationTime**
- **simulationTimeStep**
- **solutionsDirectory**
- **statisticalOptimisation**
- **statisticalOptimisationSolutions**
- **stochasticMethod**
- **stochasticOption**
- **stochasticRuns**
- **stopper**
- **strictCheckSBML**
- **surfaceParameters**
- **surfaceResolution**
- **target**
- **threshold**
- **velocities**

7.28.1 Detailed Description

Class for the running options of Byodyn.

7.28.2 Constructor & Destructor Documentation

7.28.2.1 __init__()

```
starter.ClassMetaModel.__init__ (
              self )
```

The constructor.

7.28.3 Member Function Documentation

7.28.3.1 metamodelParser()

```
starter.ClassMetaModel.metamodelParser (
              self,
              optionFile )
```

```
This method gets the information of the running obtions
and it sets the object.
```

References **starter.ClassMetaModel.bandwidth**, **starter.ClassMetaModel.clusterInputData**, **starter.ClassMetaModel.figureFormat**, **starter.ClassMetaModel.fixedParameters**, **starter.ClassMetaModel.gaMutationRate**, **starter.ClassMetaModel.gaPopulation**, **starter.ClassMetaModel.gaTranslocationRate**, **starter.ClassMetaModel.graphics**, **starter.ClassMetaModel.identifiabilityCriteria**, **starter.ClassMetaModel.identifiabilityOutputs**, **starter.ClassMetaModel.initialConcentrationsToVary**, **starter.ClassMetaModel.initialConditionsSolutions**, **starter.ClassMetaModel.integrationMethod**, **starter.ClassMetaModel.integrationOption**, **starter.ClassMetaModel.integrationTolerance**, **starter.ClassMetaModel.modelFile**, **starter.ClassMetaModel.modelFormat**, **starter.ClassMetaModel.oedResolution**, **starter.ClassMetaModel.oedTargetSpecies**, **starter.ClassMetaModel.oedTask**, **starter.ClassMetaModel.onlyLastState**, **starter.ClassMetaModel.optimisationMethod**, **starter.ClassMetaModel.optionalOutputFormat**, **sampler.ClassSample.parameters**, **sbmlWorker.ClassModelSBML.parameters**, **starter.ClassMetaModel.parameters**, **tagParser.ClassModelTags.parameters**, **starter.ClassMetaModel.parametersToVary**, **starter.ClassMetaModel.plottingKeys**, **starter.ClassMetaModel.RKCustomButcherTableau**, **starter.ClassMetaModel.RKDampedParameters**, **starter.ClassMetaModel.runningType**, **starter.ClassMetaModel.runWithBackup**, **starter.ClassMetaModel.sampleMethod**, **sampler.ClassSamplerMonteCarlo.sampleSize**, **starter.ClassMetaModel.sampleSize**, **starter.ClassMetaModel.sensitivityParameters**, **starter.ClassMetaModel.separatedGraphs**, **starter.ClassMetaModel.showingPlot**, **starter.ClassMetaModel.simulationTime**, **starter.ClassMetaModel.simulationTimeStep**, **starter.ClassMetaModel.solutionsDirectory**, **starter.ClassMetaModel.statisticalOptimisation**, **starter.ClassMetaModel.statisticalOptimisationSolutions**, **starter.ClassMetaModel.stochasticMethod**, **starter.ClassMetaModel.stochasticOption**, **starter.ClassMetaModel.stochasticRuns**, **starter.ClassMetaModel.stopper**, **starter.ClassMetaModel.strictCheckSBML**, **starter.ClassMetaModel.surfaceParameters**, **starter.ClassMetaModel.surfaceResolution**, **starter.ClassMetaModel.target**, **starter.ClassMetaModel.threshold**, and **starter.ClassMetaModel.velocities**.

7.28.4 Member Data Documentation

7.28.4.1 bandwidth

```
starter.ClassMetaModel.bandwidth
```

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.2 clusterInputData

```
starter.ClassMetaModel.clusterInputData
```

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.3 confidenceIntervals

```
starter.ClassMetaModel.confidenceIntervals
```

7.28.4.4 figureFormat

`starter.ClassMetaModel.figureFormat`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.5 fixedParameters

`starter.ClassMetaModel.fixedParameters`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.6 gaMutationRate

`starter.ClassMetaModel.gaMutationRate`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.7 gaPopulation

`starter.ClassMetaModel.gaPopulation`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.8 gaTranslocationRate

`starter.ClassMetaModel.gaTranslocationRate`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.9 graphics

`starter.ClassMetaModel.graphics`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.10 hybridPath

`starter.ClassMetaModel.hybridPath`

7.28.4.11 hybridScore

`starter.ClassMetaModel.hybridScore`

7.28.4.12 identifiabilityCriteria

`starter.ClassMetaModel.identifiabilityCriteria`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.13 identifiabilityOutputs

`starter.ClassMetaModel.identifiabilityOutputs`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.14 initialConcentrationsToVary

`starter.ClassMetaModel.initialConcentrationsToVary`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.15 initialConditionsSolutions

`starter.ClassMetaModel.initialConditionsSolutions`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.16 integrationMethod

`starter.ClassMetaModel.integrationMethod`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.17 integrationOption

`starter.ClassMetaModel.integrationOption`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.18 integrationTolerance

`starter.ClassMetaModel.integrationTolerance`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.19 lastIterationScore

`starter.ClassMetaModel.lastIterationScore`

7.28.4.20 modelFile

`starter.ClassMetaModel.modelFile`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.21 modelFormat

`starter.ClassMetaModel.modelFormat`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.22 oedResolution

`starter.ClassMetaModel.oedResolution`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.23 oedTargetSpecies

`starter.ClassMetaModel.oedTargetSpecies`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.24 oedTask

`starter.ClassMetaModel.oedTask`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.25 onlyLastState

`starter.ClassMetaModel.onlyLastState`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.26 optimisationMethod

`starter.ClassMetaModel.optimisationMethod`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.27 optionalOutputFormat

`starter.ClassMetaModel.optionalOutputFormat`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.28 parameters

`starter.ClassMetaModel.parameters`

Referenced by **sbmlWorker.ClassModelSBML.__checkNonConstant()**, **sbmlWorker.ClassModelSBML._↩ _putRateruleInTopology()**, **sbmlWorker.ClassModelSBML.__readParameters()**, **sampler.ClassSample.↩ calculateFitnessFunction()**, **sbmlWorker.ClassModelSBML.checkAssignmentRule()**, **sampler.Class↩ Sample.createRandomSetOfParameters()**, **sbmlWorker.ClassModelSBML.getDefaultValue()**, **starter.↩ ClassMetaModel.metamodelParser()**, **tagParser.ClassModelTags.readInput()**, **sbmlWorker.ClassModel↩ SBML.summary()**, and **tagParser.ClassModelTags.summary()**.

7.28.4.29 parametersToVary

`starter.ClassMetaModel.parametersToVary`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.30 plottingKeys

`starter.ClassMetaModel.plottingKeys`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.31 RKCustomButcherTableau

`starter.ClassMetaModel.RKCustomButcherTableau`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.32 RKDampedParameters

`starter.ClassMetaModel.RKDampedParameters`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.33 runningType

`starter.ClassMetaModel.runningType`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.34 runWithBackup

`starter.ClassMetaModel.runWithBackup`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.35 sampleMethod

```
starter.ClassMetaModel.sampleMethod
```

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.36 sampleSize

```
starter.ClassMetaModel.sampleSize
```

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.37 sensitivityParameters

```
starter.ClassMetaModel.sensitivityParameters
```

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.38 separatedGraphs

```
starter.ClassMetaModel.separatedGraphs
```

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.39 showingPlot

```
starter.ClassMetaModel.showingPlot
```

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.40 simulationNumber

```
starter.ClassMetaModel.simulationNumber
```

7.28.4.41 simulationTime

```
starter.ClassMetaModel.simulationTime
```

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.42 simulationTimeStep

```
starter.ClassMetaModel.simulationTimeStep
```

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.43 solutionsDirectory

`starter.ClassMetaModel.solutionsDirectory`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.44 statisticalOptimisation

`starter.ClassMetaModel.statisticalOptimisation`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.45 statisticalOptimisationSolutions

`starter.ClassMetaModel.statisticalOptimisationSolutions`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.46 stochasticMethod

`starter.ClassMetaModel.stochasticMethod`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.47 stochasticOption

`starter.ClassMetaModel.stochasticOption`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.48 stochasticRuns

`starter.ClassMetaModel.stochasticRuns`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.49 stopper

`starter.ClassMetaModel.stopper`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.50 strictCheckSBML

`starter.ClassMetaModel.strictCheckSBML`

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.51 surfaceParameters

```
starter.ClassMetaModel.surfaceParameters
```

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.52 surfaceResolution

```
starter.ClassMetaModel.surfaceResolution
```

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.53 target

```
starter.ClassMetaModel.target
```

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.54 threshold

```
starter.ClassMetaModel.threshold
```

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.28.4.55 velocities

```
starter.ClassMetaModel.velocities
```

Referenced by **starter.ClassMetaModel.metamodelParser()**.

The documentation for this class was generated from the following file:

- **starter.py**

7.29 sbmlWorker.ClassModelSBML Class Reference

Public Member Functions

- **__init__** (self)
- **canBeIntegrate** (self)
- **checkAssignmentRule** (self, rule)
- **checkCompatibilities** (self, metamodel)
- **getDefaultValue** (self, a)
- **isNonConstant** (self, nc)
- **parser** (self, metamodel)
- **summary** (self, outputfiles)

Public Attributes

- **algebraicNodes**
- **compartments**
- **constantNodes**
- **delayFunctions**
- **delays**
- **events**
- **functions**
- **initialConditions**
- **nodes**
- **nonConstantCompartments**
- **nonConstantParameters**
- **parameters**
- **reactions**
- **rules**
- **stoichiometryMatrix**
- **systemName**
- **topology**
- **topologyAST**
- **xlength**
- **ywidth**

Private Member Functions

- **__checkAlgebraicNodes** (self)
- **__checkAndPutRaterules** (self)
- **__checkNonConstant** (self)
- **__putRateruleInTopology** (self, rule)
- **__readCompartments** (self, sbmlModel)
- **__readConstraints** (self, sbmlModel)
- **__readEvents** (self, sbmlModel)
- **__readFunctionDefinitions** (self, sbmlModel)
- **__readInitialAssignments** (self, sbmlModel)
- **__readNodes** (self, sbmlModel)
- **__readParameters** (self, sbmlModel, metamodel)
- **__readRules** (self, sbmlModel)
- **__readTopology** (self, sbmlModel)
- **__searchDelayFunction** (self, sbmlModel)

7.29.1 Detailed Description

```
This is the classs with the model description and methods necessary to work with the model.
```

7.29.2 Constructor & Destructor Documentation

7.29.2.1 __init__()

```
sbmlWorker.ClassModelSBML.__init__ (
              self )
```

```
        The constructor.
```

7.29.3 Member Function Documentation

7.29.3.1 __checkAlgebraicNodes()

```
sbmlWorker.ClassModelSBML.__checkAlgebraicNodes (
              self )  [private]
```

This method searches if all the species defined with boundary condition true and constant false are used in rules or events. Otherwise, it converts this species like constant species.

References **sbmlWorker.ClassModelSBML.algebraicNodes**, **tagParser.ClassModelTags.algebraicNodes**, **sbmlWorker.ClassModelSBML.constantNodes**, **tagParser.ClassModelTags.constantNodes**, **sbmlWorker.ClassModelSBML.events**, **tagParser.ClassModelTags.events**, **sbmlWorker.ClassModelSBML.rules**, and **tagParser.ClassModelTags.rules**.

Referenced by **sbmlWorker.ClassModelSBML.parser()**.

7.29.3.2 __checkAndPutRaterules()

```
sbmlWorker.ClassModelSBML.__checkAndPutRaterules (
              self )  [private]
```

This method checks the rules and puts rate rules into the model topology.

References **sbmlWorker.ClassModelSBML.__putRateruleInTopology()**, **sbmlWorker.ClassModelSBML.checkAssignmentRule()**, **sbmlWorker.ClassModelSBML.rules**, and **tagParser.ClassModelTags.rules**.

Referenced by **sbmlWorker.ClassModelSBML.parser()**.

7.29.3.3 __checkNonConstant()

```
sbmlWorker.ClassModelSBML.__checkNonConstant (
              self )  [private]
```

This internal method checks all the nonConstant parameters and compartments.
If it finds that one is not used in the equations of the system,
this one is moved to constant.
A warning is showed because in this point we are modifing the information of SBML file,
and it can be an error of SBML file.

References **sbmlWorker.ClassModelSBML.compartments**, **tagParser.ClassModelTags.compartments**, **sbmlWorker.ClassModelSBML.isNonConstant()**, **sbmlWorker.ClassModelSBML.nonConstantCompartments**, **tagParser.ClassModelTags.nonConstantCompartments**, **sbmlWorker.ClassModelSBML.nonConstantParameters**, **tagParser.ClassModelTags.nonConstantParameters**, **sampler.ClassSample.parameters**, **sbmlWorker.ClassModelSBML.parameters**, **starter.ClassMetaModel.parameters**, and **tagParser.ClassModelTags.parameters**.

Referenced by **sbmlWorker.ClassModelSBML.parser()**.

7.29.3.4 __putRateruleInTopology()

```
sbmlWorker.ClassModelSBML.__putRateruleInTopology (
              self,
              rule )  [private]
```

This method introduces rate rules in the model topology.

References **sbmlWorker.ClassModelSBML.compartments**, **tagParser.ClassModelTags.compartments**, **sbmlWorker.ClassModelSBML.nodes**, **simulator.ClassEpithelium.nodes**, **tagParser.ClassModelTags.↩ nodes**, **sbmlWorker.ClassModelSBML.nonConstantCompartments**, **tagParser.ClassModelTags.non↩ ConstantCompartments**, **sbmlWorker.ClassModelSBML.nonConstantParameters**, **tagParser.Class↩ ModelTags.nonConstantParameters**, **sampler.ClassSample.parameters**, **sbmlWorker.ClassModelSBML.↩ parameters**, **starter.ClassMetaModel.parameters**, **tagParser.ClassModelTags.parameters**, **sbmlWorker.↩ ClassModelSBML.topology**, and **tagParser.ClassModelTags.topology**.

Referenced by **sbmlWorker.ClassModelSBML.__checkAndPutRaterules()**.

7.29.3.5 __readCompartments()

```
sbmlWorker.ClassModelSBML.__readCompartments (
              self,
              sbmlModel )  [private]
```

This method reads the compartments from a SBML model.
It discriminates between constant and non constant compartments.

References **sbmlWorker.ClassModelSBML.compartments**, **tagParser.ClassModelTags.compartments**, **sbmlWorker.ClassModelSBML.nonConstantCompartments**, and **tagParser.ClassModelTags.nonConstant↩ Compartments**.

Referenced by **sbmlWorker.ClassModelSBML.parser()**.

7.29.3.6 __readConstraints()

```
sbmlWorker.ClassModelSBML.__readConstraints (
              self,
              sbmlModel )  [private]
```

This method reads constraints from a SBML file

Referenced by **sbmlWorker.ClassModelSBML.parser()**.

7.29.3.7 __readEvents()

```
sbmlWorker.ClassModelSBML.__readEvents (
              self,
              sbmlModel )  [private]
```

This method reads the events from a SBML file.

References **sbmlWorker.ClassModelSBML.events**, **tagParser.ClassModelTags.events**, and **sbmlWorker.↩ getBooleanExpression()**.

Referenced by **sbmlWorker.ClassModelSBML.parser()**.

7.29.3.8 __readFunctionDefinitions()

```
sbmlWorker.ClassModelSBML.__readFunctionDefinitions (
              self,
              sbmlModel )  [private]
```

This method reads the user function definitions from the SBML file.

References **sbmlWorker.ClassModelSBML.functions**, and **tagParser.ClassModelTags.functions**.

Referenced by **sbmlWorker.ClassModelSBML.parser()**.

7.29.3.9 __readInitialAssignments()

```
sbmlWorker.ClassModelSBML.__readInitialAssignments (
              self,
              sbmlModel )  [private]
```

This method reads the initial Assignments from a SBML file.

Referenced by **sbmlWorker.ClassModelSBML.parser()**.

7.29.3.10 __readNodes()

```
sbmlWorker.ClassModelSBML.__readNodes (
              self,
              sbmlModel )  [private]
```

```
        This method reads of species from a SBML file.
        It sets the initial conditions of the model.
```

References **sbmlWorker.ClassModelSBML.algebraicNodes**, **tagParser.ClassModelTags.algebraicNodes**, **sbmlWorker.ClassModelSBML.compartments**, **tagParser.ClassModelTags.compartments**, **sbmlWorker.ClassModelSBML.constantNodes**, **tagParser.ClassModelTags.constantNodes**, **sbmlWorker.ClassModelSBML.initialConditions**, **tagParser.ClassModelTags.initialConditions**, **sbmlWorker.ClassModelSBML.nodes**, **simulator.ClassEpithelium.nodes**, **tagParser.ClassModelTags.nodes**, **sbmlWorker.ClassModelSBML.nonConstantCompartments**, and **tagParser.ClassModelTags.nonConstantCompartments**.

Referenced by **sbmlWorker.ClassModelSBML.parser()**.

7.29.3.11 __readParameters()

```
sbmlWorker.ClassModelSBML.__readParameters (
              self,
              sbmlModel,
              metamodel )  [private]
```

```
        This method reads the parameters from a SBML file.
```

References **sbmlWorker.ClassModelSBML.nonConstantParameters**, **tagParser.ClassModelTags.nonConstantParameters**, **sampler.ClassSample.parameters**, **sbmlWorker.ClassModelSBML.parameters**, **starter.ClassMetaModel.parameters**, and **tagParser.ClassModelTags.parameters**.

Referenced by **sbmlWorker.ClassModelSBML.parser()**.

7.29.3.12 __readRules()

```
sbmlWorker.ClassModelSBML.__readRules (
              self,
              sbmlModel )  [private]
```

```
        This method reads the rules from a SBML file.
```

References **sbmlWorker.ClassModelSBML.rules**, and **tagParser.ClassModelTags.rules**.

Referenced by **sbmlWorker.ClassModelSBML.parser()**.

7.29.3.13 __readTopology()

```
sbmlWorker.ClassModelSBML.__readTopology (
              self,
              sbmlModel )  [private]
```

```
        This method builds the network topology from SBML file.
```

References **sbmlWorker.ClassModelSBML.constantNodes**, **tagParser.ClassModelTags.constantNodes**, **sbmlWorker.divCompartment()**, **sbmlWorker.ClassModelSBML.nodes**, **simulator.ClassEpithelium.nodes**, **tagParser.ClassModelTags.nodes**, **sbmlWorker.ClassModelSBML.reactions**, **sbmlWorker.ClassModel↩ SBML.stoichiometryMatrix**, **sbmlWorker.ClassModelSBML.topology**, **tagParser.ClassModelTags.topology**, **sbmlWorker.ClassModelSBML.topologyAST**, and **sbmlWorker.translateLocalParametersNamesInFormula()**.

Referenced by **sbmlWorker.ClassModelSBML.parser()**.

7.29.3.14 __searchDelayFunction()

```
sbmlWorker.ClassModelSBML.__searchDelayFunction (
              self,
              sbmlModel )  [private]
```

```
        This internal method searchs delay functions in kinetic laws.
If delay is found, it will create a non constant parameter
and an assignment rule in order to handle this delay in integrator solvers.
```

References **sbmlWorker.ClassModelSBML.delayFunctions**, **tagParser.ClassModelTags.delayFunctions**, **sbmlWorker.ClassModelSBML.delays**, **sbmlWorker.ClassModelSBML.nonConstantParameters**, **tag↩ Parser.ClassModelTags.nonConstantParameters**, **sbmlWorker.ClassModelSBML.rules**, **tagParser.Class↩ ModelTags.rules**, **formulas.solveFormula()**, **sbmlWorker.ClassModelSBML.topology**, and **tagParser.↩ ClassModelTags.topology**.

Referenced by **sbmlWorker.ClassModelSBML.parser()**.

7.29.3.15 canBeIntegrate()

```
sbmlWorker.ClassModelSBML.canBeIntegrate (
              self )
```

This method searches if the model contains kinetic information. It means kinetic laws, rules or events. It returns True or False.

References **sbmlWorker.ClassModelSBML.events**, **tagParser.ClassModelTags.events**, **sbmlWorker.ClassModelSBML.rules**, **tagParser.ClassModelTags.rules**, **sbmlWorker.ClassModelSBML.topology**, and **tagParser.ClassModelTags.topology**.

Referenced by **sbmlWorker.ClassModelSBML.parser()**.

7.29.3.16 checkAssignmentRule()

```
sbmlWorker.ClassModelSBML.checkAssignmentRule (
              self,
              rule )
```

This method checks for the assignment rules.

References **sbmlWorker.ClassModelSBML.algebraicNodes**, **tagParser.ClassModelTags.algebraicNodes**, **sbmlWorker.ClassModelSBML.compartments**, **tagParser.ClassModelTags.compartments**, **sbmlWorker.ClassModelSBML.constantNodes**, **tagParser.ClassModelTags.constantNodes**, **sbmlWorker.ClassModelSBML.initialConditions**, **tagParser.ClassModelTags.initialConditions**, **sbmlWorker.ClassModelSBML.nodes**, **simulator.ClassEpithelium.nodes**, **tagParser.ClassModelTags.nodes**, **sbmlWorker.ClassModelSBML.nonConstantCompartments**, **tagParser.ClassModelTags.nonConstantCompartments**, **sbmlWorker.ClassModelSBML.nonConstantParameters**, **tagParser.ClassModelTags.nonConstantParameters**, **sampler.ClassSample.parameters**, **sbmlWorker.ClassModelSBML.parameters**, **starter.ClassMetaModel.parameters**, **tagParser.ClassModelTags.parameters**, **formulas.solveFormula()**, **sbmlWorker.ClassModelSBML.topology**, and **tagParser.ClassModelTags.topology**.

Referenced by **sbmlWorker.ClassModelSBML.__checkAndPutRaterules()**.

7.29.3.17 checkCompatibilities()

```
sbmlWorker.ClassModelSBML.checkCompatibilities (
              self,
              metamodel )
```

This method searches for possible problems between the model and the ByoDyn version.

References **sbmlWorker.ClassModelSBML.compartments**, **tagParser.ClassModelTags.compartments**, **sbmlWorker.ClassModelSBML.delayFunctions**, **tagParser.ClassModelTags.delayFunctions**, **sbmlWorker.ClassModelSBML.events**, **tagParser.ClassModelTags.events**, **sbmlWorker.ClassModelSBML.nodes**, **simulator.ClassEpithelium.nodes**, **tagParser.ClassModelTags.nodes**, **sbmlWorker.ClassModelSBML.rules**, and **tagParser.ClassModelTags.rules**.

7.29.3.18 getDefaultValue()

```
sbmlWorker.ClassModelSBML.getDefaultValue (
              self,
              a )
```

```
        This method gets the values of the non constant parameters,
        the non constant compartments and the algebraic nodes.
        This method is required from the simulatorOpenModelica for proper writing of the equations.
```

References **sbmlWorker.ClassModelSBML.algebraicNodes**, **tagParser.ClassModelTags.algebraicNodes**, **sbmlWorker.ClassModelSBML.initialConditions**, **tagParser.ClassModelTags.initialConditions**, **sbml↩ Worker.ClassModelSBML.nodes**, **simulator.ClassEpithelium.nodes**, **tagParser.ClassModelTags.nodes**, **sbmlWorker.ClassModelSBML.nonConstantCompartments**, **tagParser.ClassModelTags.nonConstant↩ Compartments**, **sbmlWorker.ClassModelSBML.nonConstantParameters**, **tagParser.ClassModelTags.↩ nonConstantParameters**, **sampler.ClassSample.parameters**, **sbmlWorker.ClassModelSBML.parameters**, **starter.ClassMetaModel.parameters**, and **tagParser.ClassModelTags.parameters**.

7.29.3.19 isNonConstant()

```
sbmlWorker.ClassModelSBML.isNonConstant (
              self,
              nc )
```

```
        This method checks if some nonConstant element (parameter or compartment)
        is modified by rules or events. It returns a boolean with the answer.
```

References **sbmlWorker.ClassModelSBML.events**, **tagParser.ClassModelTags.events**, **sbmlWorker.Class↩ ModelSBML.rules**, and **tagParser.ClassModelTags.rules**.

Referenced by **sbmlWorker.ClassModelSBML.__checkNonConstant()**.

7.29.3.20 parser()

```
sbmlWorker.ClassModelSBML.parser (
              self,
              metamodel )
```

```
        This method reads a SBML model and transfrom this into  the class ClassModelSBML.
```

References **sbmlWorker.ClassModelSBML.__checkAlgebraicNodes()**, **sbmlWorker.ClassModelSBML.↩ __checkAndPutRaterules()**, **sbmlWorker.ClassModelSBML.__checkNonConstant()**, **sbmlWorker.↩ ClassModelSBML.__readCompartments()**, **sbmlWorker.ClassModelSBML.__readConstraints()**, **sbml↩ Worker.ClassModelSBML.__readEvents()**, **sbmlWorker.ClassModelSBML.__readFunctionDefinitions()**, **sbmlWorker.ClassModelSBML.__readInitialAssignments()**, **sbmlWorker.ClassModelSBML.__read↩ Nodes()**, **sbmlWorker.ClassModelSBML.__readParameters()**, **sbmlWorker.ClassModelSBML.__read↩ Rules()**, **sbmlWorker.ClassModelSBML.__readTopology()**, **sbmlWorker.ClassModelSBML.__search↩ DelayFunction()**, **sbmlWorker.ClassModelSBML.canBeIntegrate()**, **sbmlWorker.ClassModelSBML.non↩ ConstantCompartments**, **tagParser.ClassModelTags.nonConstantCompartments**, **sbmlWorker.Class↩ ModelSBML.nonConstantParameters**, **tagParser.ClassModelTags.nonConstantParameters**, **sbmlWorker.↩ ClassModelSBML.rules**, **tagParser.ClassModelTags.rules**, **sbmlWorker.ClassModelSBML.systemName**, and **tagParser.ClassModelTags.systemName**.

7.29.3.21 summary()

```
sbmlWorker.ClassModelSBML.summary (
              self,
              outputfiles )
```

This method writes a short summary of the model description in an output file.

References **sbmlWorker.ClassModelSBML.constantNodes**, **tagParser.ClassModelTags.constantNodes**, **sbmlWorker.ClassModelSBML.events**, **tagParser.ClassModelTags.events**, **sbmlWorker.ClassModelSBML.nodes**, **simulator.ClassEpithelium.nodes**, **tagParser.ClassModelTags.nodes**, **sbmlWorker.ClassModelSBML.nonConstantParameters**, **tagParser.ClassModelTags.nonConstantParameters**, **sampler.ClassSample.parameters**, **sbmlWorker.ClassModelSBML.parameters**, **starter.ClassMetaModel.parameters**, **tagParser.ClassModelTags.parameters**, **sbmlWorker.ClassModelSBML.rules**, **tagParser.ClassModelTags.rules**, **sbmlWorker.ClassModelSBML.systemName**, and **tagParser.ClassModelTags.systemName**.

7.29.4 Member Data Documentation

7.29.4.1 algebraicNodes

```
sbmlWorker.ClassModelSBML.algebraicNodes
```

Referenced by **sbmlWorker.ClassModelSBML.__checkAlgebraicNodes()**, **sbmlWorker.ClassModelSBML.__readNodes()**, **sbmlWorker.ClassModelSBML.checkAssignmentRule()**, and **sbmlWorker.ClassModelSBML.getDefaultValue()**.

7.29.4.2 compartments

```
sbmlWorker.ClassModelSBML.compartments
```

Referenced by **sbmlWorker.ClassModelSBML.__checkNonConstant()**, **sbmlWorker.ClassModelSBML.__putRateruleInTopology()**, **sbmlWorker.ClassModelSBML.__readCompartments()**, **sbmlWorker.ClassModelSBML.__readNodes()**, **sbmlWorker.ClassModelSBML.checkAssignmentRule()**, and **sbmlWorker.ClassModelSBML.checkCompatibilities()**.

7.29.4.3 constantNodes

```
sbmlWorker.ClassModelSBML.constantNodes
```

Referenced by **sbmlWorker.ClassModelSBML.__checkAlgebraicNodes()**, **sbmlWorker.ClassModelSBML.__readNodes()**, **sbmlWorker.ClassModelSBML.__readTopology()**, **sbmlWorker.ClassModelSBML.checkAssignmentRule()**, and **sbmlWorker.ClassModelSBML.summary()**.

7.29.4.4 delayFunctions

```
sbmlWorker.ClassModelSBML.delayFunctions
```

Referenced by **sbmlWorker.ClassModelSBML.__searchDelayFunction()**, and **sbmlWorker.ClassModelSBML.checkCompatibilities()**.

7.29.4.5 delays

sbmlWorker.ClassModelSBML.delays

Referenced by **sbmlWorker.ClassModelSBML.__searchDelayFunction()**.

7.29.4.6 events

sbmlWorker.ClassModelSBML.events

Referenced by **sbmlWorker.ClassModelSBML.__checkAlgebraicNodes()**, **sbmlWorker.ClassModelSBML.__readEvents()**, **sbmlWorker.ClassModelSBML.canBeIntegrate()**, **sbmlWorker.ClassModelSBML.checkCompatibilities()**, **sbmlWorker.ClassModelSBML.isNonConstant()**, and **sbmlWorker.ClassModelSBML.summary()**.

7.29.4.7 functions

sbmlWorker.ClassModelSBML.functions

Referenced by **sbmlWorker.ClassModelSBML.__readFunctionDefinitions()**.

7.29.4.8 initialConditions

sbmlWorker.ClassModelSBML.initialConditions

Referenced by **sbmlWorker.ClassModelSBML.__readNodes()**, **sbmlWorker.ClassModelSBML.checkAssignmentRule()**, **sbmlWorker.ClassModelSBML.getDefaultValue()**, and **tagParser.ClassModelTags.readInput()**.

7.29.4.9 nodes

sbmlWorker.ClassModelSBML.nodes

Referenced by **sbmlWorker.ClassModelSBML.__putRateruleInTopology()**, **sbmlWorker.ClassModelSBML.__readNodes()**, **sbmlWorker.ClassModelSBML.__readTopology()**, **sbmlWorker.ClassModelSBML.checkAssignmentRule()**, **sbmlWorker.ClassModelSBML.checkCompatibilities()**, **sbmlWorker.ClassModelSBML.getDefaultValue()**, **tagParser.ClassModelTags.readInput()**, **sbmlWorker.ClassModelSBML.summary()**, and **tagParser.ClassModelTags.summary()**.

7.29.4.10 nonConstantCompartments

sbmlWorker.ClassModelSBML.nonConstantCompartments

Referenced by **sbmlWorker.ClassModelSBML.__checkNonConstant()**, **sbmlWorker.ClassModelSBML.__putRateruleInTopology()**, **sbmlWorker.ClassModelSBML.__readCompartments()**, **sbmlWorker.ClassModelSBML.__readNodes()**, **sbmlWorker.ClassModelSBML.checkAssignmentRule()**, **sbmlWorker.ClassModelSBML.getDefaultValue()**, and **sbmlWorker.ClassModelSBML.parser()**.

7.29.4.11 nonConstantParameters

`sbmlWorker.ClassModelSBML.nonConstantParameters`

Referenced by **sbmlWorker.ClassModelSBML.__checkNonConstant()**, **sbmlWorker.ClassModelSBML.__putRateruleInTopology()**, **sbmlWorker.ClassModelSBML.__readParameters()**, **sbmlWorker.ClassModelSBML.__searchDelayFunction()**, **sbmlWorker.ClassModelSBML.checkAssignmentRule()**, **sbmlWorker.ClassModelSBML.getDefaultValue()**, **sbmlWorker.ClassModelSBML.parser()**, and **sbmlWorker.ClassModelSBML.summary()**.

7.29.4.12 parameters

`sbmlWorker.ClassModelSBML.parameters`

Referenced by **sbmlWorker.ClassModelSBML.__checkNonConstant()**, **sbmlWorker.ClassModelSBML._putRateruleInTopology()**, **sbmlWorker.ClassModelSBML.__readParameters()**, **sampler.ClassSample.calculateFitnessFunction()**, **sbmlWorker.ClassModelSBML.checkAssignmentRule()**, **sampler.ClassSample.createRandomSetOfParameters()**, **sbmlWorker.ClassModelSBML.getDefaultValue()**, **starter.ClassMetaModel.metamodelParser()**, **tagParser.ClassModelTags.readInput()**, **sbmlWorker.ClassModelSBML.summary()**, and **tagParser.ClassModelTags.summary()**.

7.29.4.13 reactions

`sbmlWorker.ClassModelSBML.reactions`

Referenced by **sbmlWorker.ClassModelSBML.__readTopology()**.

7.29.4.14 rules

`sbmlWorker.ClassModelSBML.rules`

Referenced by **sbmlWorker.ClassModelSBML.__checkAlgebraicNodes()**, **sbmlWorker.ClassModelSBML._checkAndPutRaterules()**, **sbmlWorker.ClassModelSBML.__readRules()**, **sbmlWorker.ClassModelSBML.__searchDelayFunction()**, **sbmlWorker.ClassModelSBML.canBeIntegrate()**, **sbmlWorker.ClassModelSBML.checkCompatibilities()**, **sbmlWorker.ClassModelSBML.isNonConstant()**, **sbmlWorker.ClassModelSBML.parser()**, and **sbmlWorker.ClassModelSBML.summary()**.

7.29.4.15 stoichiometryMatrix

`sbmlWorker.ClassModelSBML.stoichiometryMatrix`

Referenced by **sbmlWorker.ClassModelSBML.__readTopology()**.

7.29.4.16 systemName

`sbmlWorker.ClassModelSBML.systemName`

Referenced by **sbmlWorker.ClassModelSBML.parser()**, **tagParser.ClassModelTags.readInput()**, **sbmlWorker.ClassModelSBML.summary()**, and **tagParser.ClassModelTags.summary()**.

7.29.4.17 topology

`sbmlWorker.ClassModelSBML.topology`

Referenced by **sbmlWorker.ClassModelSBML.__putRateruleInTopology()**, **sbmlWorker.ClassModelSBML.__readTopology()**, **sbmlWorker.ClassModelSBML.__searchDelayFunction()**, **sbmlWorker.ClassModelSBML.canBeIntegrate()**, **sbmlWorker.ClassModelSBML.checkAssignmentRule()**, and **tagParser.ClassModelTags.readInput()**.

7.29.4.18 topologyAST

`sbmlWorker.ClassModelSBML.topologyAST`

Referenced by **sbmlWorker.ClassModelSBML.__readTopology()**.

7.29.4.19 xlength

`sbmlWorker.ClassModelSBML.xlength`

Referenced by **tagParser.ClassModelTags.readInput()**.

7.29.4.20 ywidth

`sbmlWorker.ClassModelSBML.ywidth`

Referenced by **tagParser.ClassModelTags.readInput()**.

The documentation for this class was generated from the following file:

- **sbmlWorker.py**

7.30 tagParser.ClassModelTags Class Reference

Public Member Functions

- **__init__** (self)
- **readInput** (self, metamodel)
- **summary** (self, outputfiles)

Public Attributes

- **algebraicNodes**
- **compartments**
- **constantNodes**
- **delayFunctions**
- **events**
- **functions**
- **initialConditions**
- **nodes**
- **nonConstantCompartments**
- **nonConstantParameters**
- **parameters**
- **rules**
- **systemName**
- **topology**
- **xlength**
- **ywidth**

7.30.1 Detailed Description

Class for the model.

7.30.2 Constructor & Destructor Documentation

7.30.2.1 __init__()

```
tagParser.ClassModelTags.__init__ (
              self )
```

The constructor.

7.30.3 Member Function Documentation

7.30.3.1 readInput()

```
tagParser.ClassModelTags.readInput (
              self,
              metamodel )
```

This method gets the information of the input file and it sets the object.

References **sbmlWorker.ClassModelSBML.initialConditions**, **tagParser.ClassModelTags.initialConditions**, **sbmlWorker.ClassModelSBML.nodes**, **simulator.ClassEpithelium.nodes**, **tagParser.ClassModelTags.nodes**, **sampler.ClassSample.parameters**, **sbmlWorker.ClassModelSBML.parameters**, **starter.ClassMetaModel.parameters**, **tagParser.ClassModelTags.parameters**, **sbmlWorker.ClassModelSBML.systemName**, **tagParser.ClassModelTags.systemName**, **sbmlWorker.ClassModelSBML.topology**, **tagParser.ClassModelTags.topology**, **sbmlWorker.ClassModelSBML.xlength**, **tagParser.ClassModelTags.xlength**, **sbmlWorker.ClassModelSBML.ywidth**, and **tagParser.ClassModelTags.ywidth**.

7.30.3.2 summary()

```
tagParser.ClassModelTags.summary (
              self,
              outputfiles )
```

```
        This method writes a short summary of the model description in an output file.
```

References **sbmlWorker.ClassModelSBML.nodes**, **simulator.ClassEpithelium.nodes**, **tagParser.ClassModelTags.nodes**, **sampler.ClassSample.parameters**, **sbmlWorker.ClassModelSBML.parameters**, **starter.ClassMetaModel.parameters**, **tagParser.ClassModelTags.parameters**, **sbmlWorker.ClassModelSBML.systemName**, and **tagParser.ClassModelTags.systemName**.

7.30.4 Member Data Documentation

7.30.4.1 algebraicNodes

```
tagParser.ClassModelTags.algebraicNodes
```

Referenced by **sbmlWorker.ClassModelSBML.__checkAlgebraicNodes()**, **sbmlWorker.ClassModelSBML.__readNodes()**, **sbmlWorker.ClassModelSBML.checkAssignmentRule()**, and **sbmlWorker.ClassModelSBML.getDefaultValue()**.

7.30.4.2 compartments

```
tagParser.ClassModelTags.compartments
```

Referenced by **sbmlWorker.ClassModelSBML.__checkNonConstant()**, **sbmlWorker.ClassModelSBML.__putRateruleInTopology()**, **sbmlWorker.ClassModelSBML.__readCompartments()**, **sbmlWorker.ClassModelSBML.__readNodes()**, **sbmlWorker.ClassModelSBML.checkAssignmentRule()**, and **sbmlWorker.ClassModelSBML.checkCompatibilities()**.

7.30.4.3 constantNodes

```
tagParser.ClassModelTags.constantNodes
```

Referenced by **sbmlWorker.ClassModelSBML.__checkAlgebraicNodes()**, **sbmlWorker.ClassModelSBML.__readNodes()**, **sbmlWorker.ClassModelSBML.__readTopology()**, **sbmlWorker.ClassModelSBML.checkAssignmentRule()**, and **sbmlWorker.ClassModelSBML.summary()**.

7.30.4.4 delayFunctions

```
tagParser.ClassModelTags.delayFunctions
```

Referenced by **sbmlWorker.ClassModelSBML.__searchDelayFunction()**, and **sbmlWorker.ClassModelSBML.checkCompatibilities()**.

7.30.4.5 events

tagParser.ClassModelTags.events

Referenced by **sbmlWorker.ClassModelSBML.__checkAlgebraicNodes()**, **sbmlWorker.ClassModelSBML.__readEvents()**, **sbmlWorker.ClassModelSBML.canBeIntegrate()**, **sbmlWorker.ClassModelSBML.checkCompatibilities()**, **sbmlWorker.ClassModelSBML.isNonConstant()**, and **sbmlWorker.ClassModelSBML.summary()**.

7.30.4.6 functions

tagParser.ClassModelTags.functions

Referenced by **sbmlWorker.ClassModelSBML.__readFunctionDefinitions()**.

7.30.4.7 initialConditions

tagParser.ClassModelTags.initialConditions

Referenced by **sbmlWorker.ClassModelSBML.__readNodes()**, **sbmlWorker.ClassModelSBML.checkAssignmentRule()**, **sbmlWorker.ClassModelSBML.getDefaultValue()**, and **tagParser.ClassModelTags.readInput()**.

7.30.4.8 nodes

tagParser.ClassModelTags.nodes

Referenced by **sbmlWorker.ClassModelSBML.__putRateruleInTopology()**, **sbmlWorker.ClassModelSBML.__readNodes()**, **sbmlWorker.ClassModelSBML.__readTopology()**, **sbmlWorker.ClassModelSBML.checkAssignmentRule()**, **sbmlWorker.ClassModelSBML.checkCompatibilities()**, **sbmlWorker.ClassModelSBML.getDefaultValue()**, **tagParser.ClassModelTags.readInput()**, **sbmlWorker.ClassModelSBML.summary()**, and **tagParser.ClassModelTags.summary()**.

7.30.4.9 nonConstantCompartments

tagParser.ClassModelTags.nonConstantCompartments

Referenced by **sbmlWorker.ClassModelSBML.__checkNonConstant()**, **sbmlWorker.ClassModelSBML.__putRateruleInTopology()**, **sbmlWorker.ClassModelSBML.__readCompartments()**, **sbmlWorker.ClassModelSBML.__readNodes()**, **sbmlWorker.ClassModelSBML.checkAssignmentRule()**, **sbmlWorker.ClassModelSBML.getDefaultValue()**, and **sbmlWorker.ClassModelSBML.parser()**.

7.30.4.10 nonConstantParameters

tagParser.ClassModelTags.nonConstantParameters

Referenced by **sbmlWorker.ClassModelSBML.__checkNonConstant()**, **sbmlWorker.ClassModelSBML.__putRateruleInTopology()**, **sbmlWorker.ClassModelSBML.__readParameters()**, **sbmlWorker.ClassModelSBML.__searchDelayFunction()**, **sbmlWorker.ClassModelSBML.checkAssignmentRule()**, **sbmlWorker.ClassModelSBML.getDefaultValue()**, **sbmlWorker.ClassModelSBML.parser()**, and **sbmlWorker.ClassModelSBML.summary()**.

7.30.4.11 parameters

`tagParser.ClassModelTags.parameters`

Referenced by **sbmlWorker.ClassModelSBML.__checkNonConstant()**, **sbmlWorker.ClassModelSBML._↩ _putRateruleInTopology()**, **sbmlWorker.ClassModelSBML.__readParameters()**, **sampler.ClassSample.↩ calculateFitnessFunction()**, **sbmlWorker.ClassModelSBML.checkAssignmentRule()**, **sampler.Class↩ Sample.createRandomSetOfParameters()**, **sbmlWorker.ClassModelSBML.getDefaultValue()**, **starter.↩ ClassMetaModel.metamodelParser()**, **tagParser.ClassModelTags.readInput()**, **sbmlWorker.ClassModel↩ SBML.summary()**, and **tagParser.ClassModelTags.summary()**.

7.30.4.12 rules

`tagParser.ClassModelTags.rules`

Referenced by **sbmlWorker.ClassModelSBML.__checkAlgebraicNodes()**, **sbmlWorker.ClassModelSBML._↩ _checkAndPutRaterules()**, **sbmlWorker.ClassModelSBML.__readRules()**, **sbmlWorker.ClassModelSBML.↩ __searchDelayFunction()**, **sbmlWorker.ClassModelSBML.canBeIntegrate()**, **sbmlWorker.ClassModel↩ SBML.checkCompatibilities()**, **sbmlWorker.ClassModelSBML.isNonConstant()**, **sbmlWorker.ClassModel↩ SBML.parser()**, and **sbmlWorker.ClassModelSBML.summary()**.

7.30.4.13 systemName

`tagParser.ClassModelTags.systemName`

Referenced by **sbmlWorker.ClassModelSBML.parser()**, **tagParser.ClassModelTags.readInput()**, **sbml↩ Worker.ClassModelSBML.summary()**, and **tagParser.ClassModelTags.summary()**.

7.30.4.14 topology

`tagParser.ClassModelTags.topology`

Referenced by **sbmlWorker.ClassModelSBML.__putRateruleInTopology()**, **sbmlWorker.ClassModelSBML.↩ __readTopology()**, **sbmlWorker.ClassModelSBML.__searchDelayFunction()**, **sbmlWorker.ClassModel↩ SBML.canBeIntegrate()**, **sbmlWorker.ClassModelSBML.checkAssignmentRule()**, and **tagParser.Class↩ ModelTags.readInput()**.

7.30.4.15 xlength

`tagParser.ClassModelTags.xlength`

Referenced by **tagParser.ClassModelTags.readInput()**.

7.30.4.16 ywidth

`tagParser.ClassModelTags.ywidth`

Referenced by **tagParser.ClassModelTags.readInput()**.

The documentation for this class was generated from the following file:

- **tagParser.py**

7.31 testers.ClassNumberOfSimulationsStopperTest Class Reference

Inheritance diagram for testers.ClassNumberOfSimulationsStopperTest:

Collaboration diagram for testers.ClassNumberOfSimulationsStopperTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Additional Inherited Members

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.31.1 Detailed Description

```
Class for testing the results of an optimisation constrained by numberOfSimulations variable.
```

7.31.2 Member Function Documentation

7.31.2.1 analyseResults()

```
testers.ClassNumberOfSimulationsStopperTest.analyseResults (
              self )
```

```
        This method analyses the result of an optimisation constrained by numberOfSimulations variable.
```

References **testers.ClassTest.obtainedResult**, **testers.ClassFitnessFunctionSurfaceTest.obtained↩Result**, **testers.ClassOEDTest.obtainedResult**, **testers.ClassSciPyTest.obtainedResult**, **testers.Class↩SensitivityTest.obtainedResult**, **testers.ClassTagFormatTest.obtainedResult**, **testers.ClassTrajectories↩ReconstructionTest.obtainedResult**, and **testers.ClassWithoutGraphicsTest.obtainedResult**.

Referenced by **testers.ClassTest.run()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.32 testers.ClassOEDTest Class Reference

Inheritance diagram for testers.ClassOEDTest:

Collaboration diagram for testers.ClassOEDTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Public Attributes

- **expectedResult**
- **obtainedResult**

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.32.1 Detailed Description

```
Class for testing the results of the optimal experimental design.
```

7.32.2 Member Function Documentation

7.32.2.1 analyseResults()

```
testers.ClassOEDTest.analyseResults (
              self )

        This method analyses the results of the optimal experimental design.
```

References **testers.ClassTest.expectedResult**, **testers.ClassOEDTest.expectedResult**, **testers.Class↩ SciPyTest.expectedResult**, **testers.ClassTrajectoriesReconstructionTest.expectedResult**, **testers.Class↩ Test.obtainedResult**, **testers.ClassFitnessFunctionSurfaceTest.obtainedResult**, **testers.ClassOEDTest.↩ obtainedResult**, **testers.ClassSciPyTest.obtainedResult**, **testers.ClassSensitivityTest.obtainedResult**, **testers.ClassTagFormatTest.obtainedResult**, **testers.ClassTrajectoriesReconstructionTest.obtainedResult**, and **testers.ClassWithoutGraphicsTest.obtainedResult**.

Referenced by **testers.ClassTest.run()**.

7.32.3 Member Data Documentation

7.32.3.1 expectedResult

`testers.ClassOEDTest.expectedResult`

Referenced by **testers.ClassIdentifiabilityTest.analyseResults()**, **testers.ClassOEDTest.analyseResults()**, **testers.ClassSciPyTest.analyseResults()**, **testers.ClassSensitivityTest.analyseResults()**, **testers.Class↩ TrajectoriesReconstructionTest.analyseResults()**, and **testers.ClassTest.dataFilesChecker()**.

7.32.3.2 obtainedResult

`testers.ClassOEDTest.obtainedResult`

Referenced by **testers.ClassCluster2DTest.analyseResults()**, **testers.ClassCluster3DTest.analyseResults()**, **testers.ClassFigureFormatTest.analyseResults()**, **testers.ClassFitnessFunctionCalculationTest.analyse↩ Results()**, **testers.ClassFitnessFunctionSurfaceTest.analyseResults()**, **testers.ClassGeneticAlgorithm↩ Test.analyseResults()**, **testers.ClassHybridOnePhaseTest.analyseResults()**, **testers.ClassHybridTwo↩ PhasesTest.analyseResults()**, **testers.ClassIdentifiabilityTest.analyseResults()**, **testers.ClassLocal↩ SearchOptimisationTest.analyseResults()**, **testers.ClassNumberOfSimulationsStopperTest.analyse↩ Results()**, **testers.ClassOEDTest.analyseResults()**, **testers.ClassOptionalOutputFormatTest.analyse↩ Results()**, **testers.ClassPlotKeysTest.analyseResults()**, **testers.ClassRandomSearchOptimisationTest.↩ analyseResults()**, **testers.ClassSciPyTest.analyseResults()**, **testers.ClassScoreStopperTest.analyse↩ Results()**, **testers.ClassSensitivityTest.analyseResults()**, **testers.ClassStochasticGeneralTest.analyse↩ Results()**, **testers.ClassStochasticLastStateTest.analyseResults()**, **testers.ClassStochasticSeparated↩ GraphsTest.analyseResults()**, **testers.ClassStochasticSingleFigureTest.analyseResults()**, **testers.↩ ClassTagFormatTest.analyseResults()**, **testers.ClassTrajectoriesReconstructionTest.analyseResults()**, and **testers.ClassTest.dataFilesChecker()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.33 errorMessages.ClassOptimalExperimentalDesignException Class Reference

Inheritance diagram for errorMessages.ClassOptimalExperimentalDesignException:

Collaboration diagram for errorMessages.ClassOptimalExperimentalDesignException:

Public Member Functions

- **printExceptionInfo** (self)

Public Member Functions inherited from errorMessages.ClassByoDynException

- **__init__** (self, **errorString**)

Additional Inherited Members

Public Attributes inherited from errorMessages.ClassByoDynException

- **errorString**

7.33.1 Detailed Description

```
This class deals with the specific errors of the optimalExperimentalDesign module.
```

7.33.2 Member Function Documentation

7.33.2.1 printExceptionInfo()

```
errorMessages.ClassOptimalExperimentalDesignException.printExceptionInfo (
              self )

                This method prints the specific error string corresponding to the module and then the specific
```

Reimplemented from **errorMessages.ClassByoDynException** (p. **??**).

The documentation for this class was generated from the following file:

- **errorMessages.py**

7.34 errorMessages.ClassOptimiserException Class Reference

Inheritance diagram for errorMessages.ClassOptimiserException:

Collaboration diagram for errorMessages.ClassOptimiserException:

Public Member Functions

- **printExceptionInfo** (self)

Public Member Functions inherited from errorMessages.ClassByoDynException

- **__init__** (self, **errorString**)

Additional Inherited Members

Public Attributes inherited from errorMessages.ClassByoDynException

- **errorString**

7.34.1 Detailed Description

```
This class deals with the specific errors of the optimiser module.
```

7.34.2 Member Function Documentation

7.34.2.1 printExceptionInfo()

```
errorMessages.ClassOptimiserException.printExceptionInfo (
             self )

               This method prints the specific error string corresponding to the module and then the specific
```

Reimplemented from **errorMessages.ClassByoDynException** (p. **??**).

The documentation for this class was generated from the following file:

- **errorMessages.py**

7.35 testers.ClassOptionalOutputFormatTest Class Reference

Inheritance diagram for testers.ClassOptionalOutputFormatTest:

Collaboration diagram for testers.ClassOptionalOutputFormatTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Additional Inherited Members

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.35.1 Detailed Description

```
Class for testing the results of the optionalOutputFormat variable.
```

7.35.2 Member Function Documentation

7.35.2.1 analyseResults()

```
testers.ClassOptionalOutputFormatTest.analyseResults (
              self )
```

```
        This method analyses the results of a simulation with the output formatted in comma separated value.
```

References **testers.ClassTest.obtainedResult**, **testers.ClassFitnessFunctionSurfaceTest.obtained↩ Result**, **testers.ClassOEDTest.obtainedResult**, **testers.ClassSciPyTest.obtainedResult**, **testers.Class↩ SensitivityTest.obtainedResult**, **testers.ClassTagFormatTest.obtainedResult**, **testers.ClassTrajectories↩ ReconstructionTest.obtainedResult**, and **testers.ClassWithoutGraphicsTest.obtainedResult**.

Referenced by **testers.ClassTest.run()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.36 pca.ClassPCA Class Reference

Public Member Functions

- **__init__** (self)
- **setCOV** (self)
- **setDataSet** (self, dataMatrix)
- **setDeviationsFromMean** (self)
- **setEigen** (self)

Public Attributes

- **B**
- **COV**
- **eigenList**
- **eigenValues**
- **eigenVectors**
- **X**

7.36.1 Detailed Description

```
Class for the Principal Component Analysis
```

7.36.2 Constructor & Destructor Documentation

7.36.2.1 __init__()

```
pca.ClassPCA.__init__ (
              self )
```

This is the constructor.

7.36.3 Member Function Documentation

7.36.3.1 setCOV()

```
pca.ClassPCA.setCOV (
              self )
```

References **pca.ClassPCA.B**, **identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.COV**, and **pca.ClassPCA.COV**.

Referenced by **identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.setCorrelationMatrix()**.

7.36.3.2 setDataSet()

```
pca.ClassPCA.setDataSet (
              self,
              dataMatrix )
```

References **pca.ClassPCA.X**.

7.36.3.3 setDeviationsFromMean()

```
pca.ClassPCA.setDeviationsFromMean (
              self )
```

References **pca.ClassPCA.B**, and **pca.ClassPCA.X**.

7.36.3.4 setEigen()

```
pca.ClassPCA.setEigen (
              self )
```

References **identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.COV**, **pca.ClassPCA.COV**, **pca.ClassPCA.eigenList**, **matrixWorker.ClassMatrix.eigenValues**, **pca.ClassPCA.eigenValues**, **matrixWorker.ClassMatrix.eigenVectors**, and **pca.ClassPCA.eigenVectors**.

7.36.4 Member Data Documentation

7.36.4.1 B

`pca.ClassPCA.B`

Referenced by **pca.ClassPCA.setCOV()**, and **pca.ClassPCA.setDeviationsFromMean()**.

7.36.4.2 COV

`pca.ClassPCA.COV`

Referenced by **identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer.confidenceIntervals()**, **identifiability↩ Analyzer.ClassIdentifiabilityAnalyzer.setCorrelationMatrix()**, **pca.ClassPCA.setCOV()**, and **pca.Class↩ PCA.setEigen()**.

7.36.4.3 eigenList

`pca.ClassPCA.eigenList`

Referenced by **pca.ClassPCA.setEigen()**.

7.36.4.4 eigenValues

`pca.ClassPCA.eigenValues`

Referenced by **matrixWorker.ClassMatrix.getMaxEigenValue()**, **matrixWorker.ClassMatrix.getMinEigen↩ Value()**, **pca.ClassPCA.setEigen()**, and **matrixWorker.ClassMatrix.setEigenValues()**.

7.36.4.5 eigenVectors

`pca.ClassPCA.eigenVectors`

Referenced by **pca.ClassPCA.setEigen()**, and **matrixWorker.ClassMatrix.setEigenValues()**.

7.36.4.6 X

`pca.ClassPCA.X`

Referenced by **pca.ClassPCA.setDataSet()**, and **pca.ClassPCA.setDeviationsFromMean()**.

The documentation for this class was generated from the following file:

- **pca.py**

7.37 testers.ClassPlotKeysTest Class Reference

Inheritance diagram for testers.ClassPlotKeysTest:

Collaboration diagram for testers.ClassPlotKeysTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Additional Inherited Members

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.37.1 Detailed Description

```
Class for testing the results of the plotKeys variable.
```

7.37.2 Member Function Documentation

7.37.2.1 analyseResults()

```
testers.ClassPlotKeysTest.analyseResults (
              self )

        This method analyses the results of a simulation without plot keys on the gnuplot figure.
```

References **testers.ClassTest.obtainedResult**, **testers.ClassFitnessFunctionSurfaceTest.obtained↩Result**, **testers.ClassOEDTest.obtainedResult**, **testers.ClassSciPyTest.obtainedResult**, **testers.Class↩SensitivityTest.obtainedResult**, **testers.ClassTagFormatTest.obtainedResult**, **testers.ClassTrajectories↩ReconstructionTest.obtainedResult**, and **testers.ClassWithoutGraphicsTest.obtainedResult**.

Referenced by **testers.ClassTest.run()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.38 testers.ClassRandomSearchOptimisationTest Class Reference

Inheritance diagram for testers.ClassRandomSearchOptimisationTest:

Collaboration diagram for testers.ClassRandomSearchOptimisationTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Additional Inherited Members

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.38.1 Detailed Description

```
Class for testing the results of the optimisations.
```

7.38.2 Member Function Documentation

7.38.2.1 analyseResults()

```
testers.ClassRandomSearchOptimisationTest.analyseResults (
              self )

        This method analyses the results of the optimisation.
        The best optimisation parameters should be stored on the xml file.
```

References **testers.ClassTest.obtainedResult**, **testers.ClassFitnessFunctionSurfaceTest.obtained↩Result**, **testers.ClassOEDTest.obtainedResult**, **testers.ClassSciPyTest.obtainedResult**, **testers.Class↩SensitivityTest.obtainedResult**, **testers.ClassTagFormatTest.obtainedResult**, **testers.ClassTrajectories↩ReconstructionTest.obtainedResult**, and **testers.ClassWithoutGraphicsTest.obtainedResult**.

Referenced by **testers.ClassTest.run()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.39 sbmlWorker.ClassReaction Class Reference

Public Member Functions

- **__init__** (self)

Public Attributes

- **id**
- **kineticLaw**
- **products**
- **reactants**

7.39.1 Detailed Description

```
Class for the reactions.
Required for the stochastic simulations.
```

7.39.2 Constructor & Destructor Documentation

7.39.2.1 __init__()

```
sbmlWorker.ClassReaction.__init__ (
              self )
```

```
        This is the constructor.
```

7.39.3 Member Data Documentation

7.39.3.1 id

```
sbmlWorker.ClassReaction.id
```

Referenced by **testers.ClassTest.__execute()**.

7.39.3.2 kineticLaw

```
sbmlWorker.ClassReaction.kineticLaw
```

7.39.3.3 products

```
sbmlWorker.ClassReaction.products
```

7.39.3.4 reactants

```
sbmlWorker.ClassReaction.reactants
```

The documentation for this class was generated from the following file:

- **sbmlWorker.py**

7.40 sbmlWorker.ClassRules Class Reference

Public Member Functions

- **__init__** (self)

Public Attributes

- **id**
- **math**
- **mathAST**
- **type**
- **variable**

7.40.1 Detailed Description

```
Class for SBML rules.
```

7.40.2 Constructor & Destructor Documentation

7.40.2.1 __init__()

```
sbmlWorker.ClassRules.__init__ (
              self )

        This is the constructer.
```

7.40.3 Member Data Documentation

7.40.3.1 id

```
sbmlWorker.ClassRules.id
```

Referenced by **testers.ClassTest.__execute()**.

7.40.3.2 math

```
sbmlWorker.ClassRules.math
```

7.40.3.3 mathAST

```
sbmlWorker.ClassRules.mathAST
```

7.40.3.4 type

```
sbmlWorker.ClassRules.type
```

7.40.3.5 variable

```
sbmlWorker.ClassRules.variable
```

The documentation for this class was generated from the following file:

- **sbmlWorker.py**

7.41 sampler.ClassSample Class Reference

Public Member Functions

- **__init__** (self)
- **calculateFitnessFunction** (self, metamodel, model, outputfiles)
- **createRandomSetOfParameters** (self, metamodel)

Public Attributes

- **fitnessFunction**
- **parameters**

7.41.1 Detailed Description

```
Class for the sample object.
```

7.41.2 Constructor & Destructor Documentation

7.41.2.1 __init__()

```
sampler.ClassSample.__init__ (
              self )
```

```
The constructor.
```

7.41.3 Member Function Documentation

7.41.3.1 calculateFitnessFunction()

```
sampler.ClassSample.calculateFitnessFunction (
              self,
              metamodel,
              model,
              outputfiles )
```

```
This method calculate the fitness function for a set of parameters.
```

References **sampler.ClassSample.fitnessFunction**, **sampler.ClassSample.parameters**, **sbmlWorker.Class↩ ModelSBML.parameters**, **starter.ClassMetaModel.parameters**, **tagParser.ClassModelTags.parameters**, and **optimiser.scoreObtainer()**.

7.41.3.2 createRandomSetOfParameters()

```
sampler.ClassSample.createRandomSetOfParameters (
              self,
              metamodel )
```

```
This method creates a random set of parameters.
The parameters to explore and its range is indicated by the parametersToVary variable in the runner file.
```

References **optimiser.linearVariation()**, **optimiser.logarithmicVariation()**, **sampler.ClassSample.parameters**, **sbmlWorker.ClassModelSBML.parameters**, **starter.ClassMetaModel.parameters**, and **tagParser.Class↩ ModelTags.parameters**.

7.41.4 Member Data Documentation

7.41.4.1 fitnessFunction

```
sampler.ClassSample.fitnessFunction
```

Referenced by **sampler.ClassSample.calculateFitnessFunction()**.

7.41.4.2 parameters

```
sampler.ClassSample.parameters
```

Referenced by **sbmlWorker.ClassModelSBML.__checkNonConstant()**, **sbmlWorker.ClassModelSBML._↩ _putRateruleInTopology()**, **sbmlWorker.ClassModelSBML.__readParameters()**, **sampler.ClassSample.↩ calculateFitnessFunction()**, **sbmlWorker.ClassModelSBML.checkAssignmentRule()**, **sampler.Class↩ Sample.createRandomSetOfParameters()**, **sbmlWorker.ClassModelSBML.getDefaultValue()**, **starter.↩ ClassMetaModel.metamodelParser()**, **tagParser.ClassModelTags.readInput()**, **sbmlWorker.ClassModel↩ SBML.summary()**, and **tagParser.ClassModelTags.summary()**.

The documentation for this class was generated from the following file:

- **sampler.py**

7.42 errorMessages.ClassSamplerException Class Reference

Inheritance diagram for errorMessages.ClassSamplerException:

Collaboration diagram for errorMessages.ClassSamplerException:

Public Member Functions

- **printExceptionInfo** (self)

Public Member Functions inherited from errorMessages.ClassByoDynException

- **__init__** (self, **errorString**)

Additional Inherited Members

Public Attributes inherited from errorMessages.ClassByoDynException

- **errorString**

7.42.1 Detailed Description

```
This class deals with the specific errors of the sampler module.
```

7.42.2 Member Function Documentation

7.42.2.1 printExceptionInfo()

```
errorMessages.ClassSamplerException.printExceptionInfo (
              self )
```

```
This method prints the specific error string corresponding to the module and then the specific
```

Reimplemented from **errorMessages.ClassByoDynException** (p. **??**).

The documentation for this class was generated from the following file:

- **errorMessages.py**

7.43 sampler.ClassSamplerMonteCarlo Class Reference

Public Member Functions

- **__init__** (self, metamodel)
- **sampling** (self, metamodel, model, outputfiles)
- **setT** (self, metamodel, model, outputfiles)

Public Attributes

- **acceptedSample**
- **nonAcceptedSample**
- **sampleSize**
- **T**

Private Member Functions

- **__lastFF** (self)

7.43.1 Detailed Description

Class for the Monte Carlo sampler. The method implemented is based in Metropolis-Hastings algorithm.

7.43.2 Constructor & Destructor Documentation

7.43.2.1 __init__()

```
sampler.ClassSamplerMonteCarlo.__init__ (
              self,
              metamodel )
```

The constructor.

7.43.3 Member Function Documentation

7.43.3.1 __lastFF()

```
sampler.ClassSamplerMonteCarlo.__lastFF (
              self )  [private]
```

This private method returns the fitness function of the last sampled element.

References **sampler.ClassSamplerMonteCarlo.acceptedSample**.

7.43.3.2 sampling()

```
sampler.ClassSamplerMonteCarlo.sampling (
              self,
              metamodel,
              model,
              outputfiles )
```

This method sampling the fitness function surface using a Monte Carlo Metropolis-Hastings algorithm.
It creates a list of accepted sampled elements and not accepted sampled elements.

7.43.3.3 setT()

```
sampler.ClassSamplerMonteCarlo.setT (
              self,
              metamodel,
              model,
              outputfiles )
```

This method sets the class variable T from a random exploration of Fitness Function surface.

7.43.4 Member Data Documentation

7.43.4.1 acceptedSample

```
sampler.ClassSamplerMonteCarlo.acceptedSample
```

Referenced by **sampler.ClassSamplerMonteCarlo.__lastFF()**.

7.43.4.2 nonAcceptedSample

```
sampler.ClassSamplerMonteCarlo.nonAcceptedSample
```

7.43.4.3 sampleSize

```
sampler.ClassSamplerMonteCarlo.sampleSize
```

Referenced by **starter.ClassMetaModel.metamodelParser()**.

7.43.4.4 T

```
sampler.ClassSamplerMonteCarlo.T
```

The documentation for this class was generated from the following file:

- **sampler.py**

7.44 errorMessages.ClassSBMLWorkerException Class Reference

Inheritance diagram for errorMessages.ClassSBMLWorkerException:

Collaboration diagram for errorMessages.ClassSBMLWorkerException:

Public Member Functions

- **printExceptionInfo** (self)

Public Member Functions inherited from errorMessages.ClassByoDynException

- **__init__** (self, **errorString**)

Additional Inherited Members

Public Attributes inherited from errorMessages.ClassByoDynException

- **errorString**

7.44.1 Detailed Description

```
This class deals with the specific errors of the sbmlWorker module.
```

7.44.2 Member Function Documentation

7.44.2.1 printExceptionInfo()

```
errorMessages.ClassSBMLWorkerException.printExceptionInfo (
              self )

                This method prints the specific error string corresponding to the module and then the specific
```

Reimplemented from **errorMessages.ClassByoDynException** (p. **??**).

The documentation for this class was generated from the following file:

- **errorMessages.py**

7.45 testers.ClassSciPyTest Class Reference

Inheritance diagram for testers.ClassSciPyTest:

Collaboration diagram for testers.ClassSciPyTest:

Public Member Functions

- **__init__** (self)
- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Public Attributes

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.45.1 Detailed Description

```
Class for testing SciPy.
```

7.45.2 Constructor & Destructor Documentation

7.45.2.1 __init__()

```
testers.ClassSciPyTest.__init__ (
              self )

        The constructor.
```

Reimplemented from **testers.ClassTest** (p. 180).

7.45.3 Member Function Documentation

7.45.3.1 analyseResults()

```
testers.ClassSciPyTest.analyseResults (
              self )

        This method analyses the results of the SciPy test.
```

References **testers.ClassTest.dataFilesChecker()**, **testers.ClassTest.expectedResult**, **testers.ClassOEDTest.expectedResult**, **testers.ClassSciPyTest.expectedResult**, **testers.ClassTrajectoriesReconstructionTest.expectedResult**, **testers.ClassTest.obtainedResult**, **testers.ClassFitnessFunctionSurfaceTest.obtainedResult**, **testers.ClassOEDTest.obtainedResult**, **testers.ClassSciPyTest.obtainedResult**, **testers.ClassSensitivityTest.obtainedResult**, **testers.ClassTagFormatTest.obtainedResult**, **testers.ClassTrajectoriesReconstructionTest.obtainedResult**, **testers.ClassWithoutGraphicsTest.obtainedResult**, **testers.ClassTest.setOutputName()**, and **testers.ClassTest.setResultName()**.

Referenced by **testers.ClassTest.run()**.

7.45.4 Member Data Documentation

7.45.4.1 expectedResult

testers.ClassSciPyTest.expectedResult

Referenced by **testers.ClassIdentifiabilityTest.analyseResults()**, **testers.ClassOEDTest.analyseResults()**, **testers.ClassSciPyTest.analyseResults()**, **testers.ClassSensitivityTest.analyseResults()**, **testers.Class↩ TrajectoriesReconstructionTest.analyseResults()**, and **testers.ClassTest.dataFilesChecker()**.

7.45.4.2 id

testers.ClassSciPyTest.id

Referenced by **testers.ClassTest.__execute()**.

7.45.4.3 obtainedResult

testers.ClassSciPyTest.obtainedResult

Referenced by **testers.ClassCluster2DTest.analyseResults()**, **testers.ClassCluster3DTest.analyseResults()**, **testers.ClassFigureFormatTest.analyseResults()**, **testers.ClassFitnessFunctionCalculationTest.analyse↩ Results()**, **testers.ClassFitnessFunctionSurfaceTest.analyseResults()**, **testers.ClassGeneticAlgorithm↩ Test.analyseResults()**, **testers.ClassHybridOnePhaseTest.analyseResults()**, **testers.ClassHybridTwo↩ PhasesTest.analyseResults()**, **testers.ClassIdentifiabilityTest.analyseResults()**, **testers.ClassLocal↩ SearchOptimisationTest.analyseResults()**, **testers.ClassNumberOfSimulationsStopperTest.analyse↩ Results()**, **testers.ClassOEDTest.analyseResults()**, **testers.ClassOptionalOutputFormatTest.analyse↩ Results()**, **testers.ClassPlotKeysTest.analyseResults()**, **testers.ClassRandomSearchOptimisationTest.↩ analyseResults()**, **testers.ClassSciPyTest.analyseResults()**, **testers.ClassScoreStopperTest.analyse↩ Results()**, **testers.ClassSensitivityTest.analyseResults()**, **testers.ClassStochasticGeneralTest.analyse↩ Results()**, **testers.ClassStochasticLastStateTest.analyseResults()**, **testers.ClassStochasticSeparated↩ GraphsTest.analyseResults()**, **testers.ClassStochasticSingleFigureTest.analyseResults()**, **testers.↩ ClassTagFormatTest.analyseResults()**, **testers.ClassTrajectoriesReconstructionTest.analyseResults()**, and **testers.ClassTest.dataFilesChecker()**.

7.45.4.4 runner

testers.ClassSciPyTest.runner

Referenced by **testers.ClassTest.__execute()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.46 testers.ClassScoreStopperTest Class Reference

Inheritance diagram for testers.ClassScoreStopperTest:

Collaboration diagram for testers.ClassScoreStopperTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Additional Inherited Members

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.46.1 Detailed Description

```
Class for testing the results of an optimisation constrained by score stopper.
```

7.46.2 Member Function Documentation

7.46.2.1 analyseResults()

```
testers.ClassScoreStopperTest.analyseResults (
              self )
```

```
        This method analyses the results of the optimisation constrained by score stopper.
```

References **testers.ClassTest.obtainedResult**, **testers.ClassFitnessFunctionSurfaceTest.obtained↩Result**, **testers.ClassOEDTest.obtainedResult**, **testers.ClassSciPyTest.obtainedResult**, **testers.Class↩SensitivityTest.obtainedResult**, **testers.ClassTagFormatTest.obtainedResult**, **testers.ClassTrajectories↩ReconstructionTest.obtainedResult**, and **testers.ClassWithoutGraphicsTest.obtainedResult**.

Referenced by **testers.ClassTest.run()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.47 sensitivityAnalyzer.ClassSensitivityAnalyzer Class Reference

Public Member Functions

- **__init__** (self, parameters)
- **calculateOverallSens** (self)
- **calculateSens** (self, model, metamodel, outputfiles)
- **plotOSTimeCourse** (self, metamodel, outputfiles)
- **writeOSTable** (self, outputfiles)

Public Attributes

- **identifiabilityCoefficients**
- **OS**
- **OSCoefficients**
- **parametersToStudy**
- **restrictedTimePoints**
- **RSCoefficients**
- **RSMatrix**

Private Member Functions

- **__obtainSensitivity** (self, model, metamodel, originalSimulation, parameterSimulation, h, parameterValue)

7.47.1 Detailed Description

```
Class for the sensitivity analysis.
```

7.47.2 Constructor & Destructor Documentation

7.47.2.1 __init__()

```
sensitivityAnalyzer.ClassSensitivityAnalyzer.__init__ (
              self,
              parameters )

        This is the constructor.
```

7.47.3 Member Function Documentation

7.47.3.1 __obtainSensitivity()

```
sensitivityAnalyzer.ClassSensitivityAnalyzer.__obtainSensitivity (
              self,
              model,
              metamodel,
              originalSimulation,
              parameterSimulation,
              h,
              parameterValue )  [private]

        This method calculates the sensitivity given the dynamics of both the system and an infinitesimally ch
        It returns trajectories.
```

References **sensitivityAnalyzer.getNodeName()**, **sensitivityAnalyzer.getPositionNode()**, and **sensitivity↩ Analyzer.ClassSensitivityAnalyzer.restrictedTimePoints**.

Referenced by **sensitivityAnalyzer.ClassSensitivityAnalyzer.calculateSens()**.

7.47.3.2 calculateOverallSens()

```
sensitivityAnalyzer.ClassSensitivityAnalyzer.calculateOverallSens (
              self )

        This method calculates single value sensitivities from the trajectories.
```

References **sensitivityAnalyzer.ClassSensitivityAnalyzer.OS**, **sensitivityAnalyzer.ClassSensitivity↩ Analyzer.OSCoefficients**, and **sensitivityAnalyzer.ClassSensitivityAnalyzer.RSCoefficients**.

7.47.3.3 calculateSens()

```
sensitivityAnalyzer.ClassSensitivityAnalyzer.calculateSens (
              self,
              model,
              metamodel,
              outputfiles )

        This method runs the simulation for a given parameter values,
        change the parameters infinitesimally and re-runs the simulation.
        With this data, the sensitivity is calculated.
```

References **sensitivityAnalyzer.ClassSensitivityAnalyzer.__obtainSensitivity()**, **sensitivityAnalyzer.↩ ClassSensitivityAnalyzer.identifiabilityCoefficients**, **simulator.obtainSimulationValues()**, **sensitivity↩ Analyzer.ClassSensitivityAnalyzer.parametersToStudy**, **sensitivityAnalyzer.ClassSensitivityAnalyzer.↩ RSCoefficients**, and **sensitivityAnalyzer.ClassSensitivityAnalyzer.RSMatrix**.

7.47.3.4 plotOSTimeCourse()

```
sensitivityAnalyzer.ClassSensitivityAnalyzer.plotOSTimeCourse (
              self,
              metamodel,
              outputfiles )

This method creates the sensitivity time course plot
```

References **sensitivityAnalyzer.ClassSensitivityAnalyzer.OSCoefficients**.

7.47.3.5 writeOSTable()

```
sensitivityAnalyzer.ClassSensitivityAnalyzer.writeOSTable (
              self,
              outputfiles )

This method writes the global sensitivity values in output file
```

References **sensitivityAnalyzer.ClassSensitivityAnalyzer.OS**.

7.47.4 Member Data Documentation

7.47.4.1 identifiabilityCoefficients

```
sensitivityAnalyzer.ClassSensitivityAnalyzer.identifiabilityCoefficients
```

Referenced by **sensitivityAnalyzer.ClassSensitivityAnalyzer.calculateSens()**.

7.47.4.2 OS

```
sensitivityAnalyzer.ClassSensitivityAnalyzer.OS
```

Referenced by **sensitivityAnalyzer.ClassSensitivityAnalyzer.calculateOverallSens()**, and **sensitivity↩ Analyzer.ClassSensitivityAnalyzer.writeOSTable()**.

7.47.4.3 OSCoefficients

```
sensitivityAnalyzer.ClassSensitivityAnalyzer.OSCoefficients
```

Referenced by **sensitivityAnalyzer.ClassSensitivityAnalyzer.calculateOverallSens()**, and **sensitivity↩ Analyzer.ClassSensitivityAnalyzer.plotOSTimeCourse()**.

7.47.4.4 parametersToStudy

```
sensitivityAnalyzer.ClassSensitivityAnalyzer.parametersToStudy
```

Referenced by **sensitivityAnalyzer.ClassSensitivityAnalyzer.calculateSens()**.

7.47.4.5 restrictedTimePoints

```
sensitivityAnalyzer.ClassSensitivityAnalyzer.restrictedTimePoints
```

Referenced by **sensitivityAnalyzer.ClassSensitivityAnalyzer.__obtainSensitivity()**.

7.47.4.6 RSCoefficients

```
sensitivityAnalyzer.ClassSensitivityAnalyzer.RSCoefficients
```

Referenced by **sensitivityAnalyzer.ClassSensitivityAnalyzer.calculateOverallSens()**, and **sensitivity↩ Analyzer.ClassSensitivityAnalyzer.calculateSens()**.

7.47.4.7 RSMatrix

```
sensitivityAnalyzer.ClassSensitivityAnalyzer.RSMatrix
```

Referenced by **sensitivityAnalyzer.ClassSensitivityAnalyzer.calculateSens()**.

The documentation for this class was generated from the following file:

- **sensitivityAnalyzer.py**

7.48 errorMessages.ClassSensitivityAnalyzerException Class Reference

Inheritance diagram for errorMessages.ClassSensitivityAnalyzerException:

Collaboration diagram for errorMessages.ClassSensitivityAnalyzerException:

Public Member Functions

- **printExceptionInfo** (self)

Public Member Functions inherited from errorMessages.ClassByoDynException

- **__init__** (self, **errorString**)

Additional Inherited Members

Public Attributes inherited from errorMessages.ClassByoDynException

- **errorString**

7.48.1 Detailed Description

```
This class deals with the specific errors of the sensitivityAnalyzer module.
```

7.48.2 Member Function Documentation

7.48.2.1 printExceptionInfo()

```
errorMessages.ClassSensitivityAnalyzerException.printExceptionInfo (
              self )
```

```
This method prints the specific error string corresponding to the module and then the specific
```

Reimplemented from **errorMessages.ClassByoDynException** (p. **??**).

The documentation for this class was generated from the following file:

- **errorMessages.py**

7.49 testers.ClassSensitivityTest Class Reference

Inheritance diagram for testers.ClassSensitivityTest:

Collaboration diagram for testers.ClassSensitivityTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Public Attributes

- **obtainedResult**

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.49.1 Detailed Description

```
Class for testing the results of the sensitivity.
```

7.49.2 Member Function Documentation

7.49.2.1 analyseResults()

```
testers.ClassSensitivityTest.analyseResults (
              self )
```

```
        This method analyses the results of the sensitivity.
```

References **testers.ClassTest.expectedResult**, **testers.ClassOEDTest.expectedResult**, **testers.ClassSciPyTest.expectedResult**, **testers.ClassTrajectoriesReconstructionTest.expectedResult**, **testers.ClassTest.obtainedResult**, **testers.ClassFitnessFunctionSurfaceTest.obtainedResult**, **testers.ClassOEDTest.obtainedResult**, **testers.ClassSciPyTest.obtainedResult**, **testers.ClassSensitivityTest.obtainedResult**, **testers.ClassTagFormatTest.obtainedResult**, **testers.ClassTrajectoriesReconstructionTest.obtainedResult**, and **testers.ClassWithoutGraphicsTest.obtainedResult**.

Referenced by **testers.ClassTest.run()**.

7.49.3 Member Data Documentation

7.49.3.1 obtainedResult

`testers.ClassSensitivityTest.obtainedResult`

Referenced by **testers.ClassCluster2DTest.analyseResults()**, **testers.ClassCluster3DTest.analyseResults()**, **testers.ClassFigureFormatTest.analyseResults()**, **testers.ClassFitnessFunctionCalculationTest.analyseResults()**, **testers.ClassFitnessFunctionSurfaceTest.analyseResults()**, **testers.ClassGeneticAlgorithmTest.analyseResults()**, **testers.ClassHybridOnePhaseTest.analyseResults()**, **testers.ClassHybridTwoPhasesTest.analyseResults()**, **testers.ClassIdentifiabilityTest.analyseResults()**, **testers.ClassLocalSearchOptimisationTest.analyseResults()**, **testers.ClassNumberOfSimulationsStopperTest.analyseResults()**, **testers.ClassOEDTest.analyseResults()**, **testers.ClassOptionalOutputFormatTest.analyseResults()**, **testers.ClassPlotKeysTest.analyseResults()**, **testers.ClassRandomSearchOptimisationTest.analyseResults()**, **testers.ClassSciPyTest.analyseResults()**, **testers.ClassScoreStopperTest.analyseResults()**, **testers.ClassSensitivityTest.analyseResults()**, **testers.ClassStochasticGeneralTest.analyseResults()**, **testers.ClassStochasticLastStateTest.analyseResults()**, **testers.ClassStochasticSeparatedGraphsTest.analyseResults()**, **testers.ClassStochasticSingleFigureTest.analyseResults()**, **testers.ClassTagFormatTest.analyseResults()**, **testers.ClassTrajectoriesReconstructionTest.analyseResults()**, and **testers.ClassTest.dataFilesChecker()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.50 testers.ClassSeparatedGraphsTest Class Reference

Inheritance diagram for testers.ClassSeparatedGraphsTest:

Collaboration diagram for testers.ClassSeparatedGraphsTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Additional Inherited Members

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.50.1 Detailed Description

```
Class for testing the results of a simulation with the option of a single file for each of the node trajectori
```

7.50.2 Member Function Documentation

7.50.2.1 analyseResults()

```
testers.ClassSeparatedGraphsTest.analyseResults (
              self )
```

```
        This method analyses the results of a simulation asking for separated graphs.
```

Referenced by **testers.ClassTest.run()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.51 testers.ClassSimulationMethodsTest Class Reference

Inheritance diagram for testers.ClassSimulationMethodsTest:

Collaboration diagram for testers.ClassSimulationMethodsTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Additional Inherited Members

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.51.1 Detailed Description

```
Class for testing the several methods of integration available from Scipy.
```

7.51.2 Member Function Documentation

7.51.2.1 analyseResults()

```
testers.ClassSimulationMethodsTest.analyseResults (
              self )
```

```
        This method analyses the results of the simulations due to different integration methods.
```

References **testers.ClassTest.dataFilesChecker()**.

Referenced by **testers.ClassTest.run()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.52 simulatorEuler.ClassSimulatorEuler Class Reference

Inheritance diagram for simulatorEuler.ClassSimulatorEuler:

Public Member Functions

- **__init__** (self)
- **integrate** (self, f, x_0, time, dt)
- **run** (self, metamodel, model, outputfiles)

Private Member Functions

- **__getVariables** (self, model)
- **__loadIntegrationFunction** (self, outputfiles)
- **__plotSimulation** (self, x, t, model, metamodel, outputfiles)
- **__storeResults** (self, x, t, model, metamodel, outputfiles)
- **__writeTopologyInput** (self, model, outputfiles, initialVariablesValues, variables)

7.52.1 Detailed Description

```
Class for the Euler simulator.
```

7.52.2 Constructor & Destructor Documentation

7.52.2.1 __init__()

```
simulatorEuler.ClassSimulatorEuler.__init__ (
              self )
```

The constructor.

7.52.3 Member Function Documentation

7.52.3.1 __getVariables()

```
simulatorEuler.ClassSimulatorEuler.__getVariables (
              self,
              model )  [private]
```

This private method sets the initial conditions of the system.

Referenced by **simulatorEuler.ClassSimulatorEuler.run()**, and **simulatorStochastic.ClassSimulator↩ Stochastic.run()**.

7.52.3.2 __loadIntegrationFunction()

```
simulatorEuler.ClassSimulatorEuler.__loadIntegrationFunction (
              self,
              outputfiles )  [private]
```

This private method loads the system of equations, that is the topology of the system.

Referenced by **simulatorEuler.ClassSimulatorEuler.run()**, and **simulatorStochastic.ClassSimulator↩ Stochastic.run()**.

7.52.3.3 __plotSimulation()

```
simulatorEuler.ClassSimulatorEuler.__plotSimulation (
              self,
              x,
              t,
              model,
              metamodel,
              outputfiles )  [private]
```

This private method creates the graph of the trajectories.

Referenced by **simulatorEuler.ClassSimulatorEuler.run()**, and **simulatorStochastic.ClassSimulator↩ Stochastic.run()**.

7.52.3.4 __storeResults()

```
simulatorEuler.ClassSimulatorEuler.__storeResults (
              self,
              x,
              t,
              model,
              metamodel,
              outputfiles )  [private]
```

This method stores the results of the simulation in a file in the output directory.

Referenced by **simulatorEuler.ClassSimulatorEuler.run()**, and **simulatorStochastic.ClassSimulator↩ Stochastic.run()**.

7.52.3.5 __writeTopologyInput()

```
simulatorEuler.ClassSimulatorEuler.__writeTopologyInput (
              self,
              model,
              outputfiles,
              initialVariablesValues,
              variables )  [private]
```

This private method writes a file with the topology of the system.

Referenced by **simulatorEuler.ClassSimulatorEuler.run()**.

7.52.3.6 integrate()

```
simulatorEuler.ClassSimulatorEuler.integrate (
              self,
              f,
              x_0,
              time,
              dt )
```

This method integrates the system of equations.

Reimplemented in **simulatorRungeKutta.ClassSimulatorRungeKutta** (p. 163).

Referenced by **simulatorEuler.ClassSimulatorEuler.run()**.

7.52.3.7 run()

```
simulatorEuler.ClassSimulatorEuler.run (
              self,
              metamodel,
              model,
              outputfiles )


        This method directs the simulation.
        It creates the initial conditions,
        loads the system of equations,
        integrates the system and
        plots the results.
```

References **simulatorEuler.ClassSimulatorEuler.__getVariables()**, **simulatorStochastic.ClassSimulator↩ Stochastic.__getVariables()**, **simulatorEuler.ClassSimulatorEuler.__loadIntegrationFunction()**, **simulator↩ Stochastic.ClassSimulatorStochastic.__loadIntegrationFunction()**, **simulatorStochastic.ClassSimulator↩ Stochastic.__plotSimulation()**, **simulatorEuler.ClassSimulatorEuler.__plotSimulation()**, **simulator↩ Euler.ClassSimulatorEuler.__storeResults()**, **simulatorStochastic.ClassSimulatorStochastic.__store↩ Results()**, **simulatorEuler.ClassSimulatorEuler.__writeTopologyInput()**, **simulatorEuler.ClassSimulator↩ Euler.integrate()**, and **simulatorRungeKutta.ClassSimulatorRungeKutta.integrate()**.

The documentation for this class was generated from the following file:

- **simulatorEuler.py**

7.53 errorMessages.ClassSimulatorEulerException Class Reference

Inheritance diagram for errorMessages.ClassSimulatorEulerException:

Collaboration diagram for errorMessages.ClassSimulatorEulerException:

Public Member Functions

- **printExceptionInfo** (self)

Public Member Functions inherited from errorMessages.ClassByoDynException

- **__init__** (self, **errorString**)

Additional Inherited Members

Public Attributes inherited from errorMessages.ClassByoDynException

- **errorString**

7.53.1 Detailed Description

```
        This class deals with the specific errors of the simulatorEuler module.
```

7.53.2 Member Function Documentation

7.53.2.1 printExceptionInfo()

```
errorMessages.ClassSimulatorEulerException.printExceptionInfo (
              self )
```

```
This method prints the specific error string corresponding to the module and then the specific
```

Reimplemented from **errorMessages.ClassByoDynException** (p. **??**).

The documentation for this class was generated from the following file:

- **errorMessages.py**

7.54 errorMessages.ClassSimulatorException Class Reference

Inheritance diagram for errorMessages.ClassSimulatorException:

Collaboration diagram for errorMessages.ClassSimulatorException:

Public Member Functions

- **printExceptionInfo** (self)

Public Member Functions inherited from errorMessages.ClassByoDynException

- **__init__** (self, **errorString**)

Additional Inherited Members

Public Attributes inherited from errorMessages.ClassByoDynException

- **errorString**

7.54.1 Detailed Description

```
This class deals with the specific errors of the simulator module.
```

7.54.2 Member Function Documentation

7.54.2.1 printExceptionInfo()

```
errorMessages.ClassSimulatorException.printExceptionInfo (
              self )

                This method prints the specific error string corresponding to the module and then the specific
```

Reimplemented from **errorMessages.ClassByoDynException** (p. **??**).

The documentation for this class was generated from the following file:

- **errorMessages.py**

7.55 simulatorOpenModelica.ClassSimulatorOpenModelica Class Reference

Public Member Functions

- **__init__** (self)
- **callSolver** (self, outputfiles)
- **createInput** (self, metamodel, model, outputfiles)
- **createOutputs** (self, model, outputfiles, metamodel)

Private Member Functions

- **__writeModel** (self, model, metamodel, outputfiles)
- **__writeRunner** (self, metamodel, model, outputfiles)

7.55.1 Detailed Description

```
Class for the OpenModelica simulator.
```

7.55.2 Constructor & Destructor Documentation

7.55.2.1 __init__()

```
simulatorOpenModelica.ClassSimulatorOpenModelica.__init__ (
              self )
```

```
The constructor.
```

7.55.3 Member Function Documentation

7.55.3.1 __writeModel()

```
simulatorOpenModelica.ClassSimulatorOpenModelica.__writeModel (
              self,
              model,
              metamodel,
              outputfiles )  [private]
```

This method writes the openModelica model file.

References **formulas.writeOpenModelicaFormula()**.

Referenced by **simulatorOpenModelica.ClassSimulatorOpenModelica.createInput()**.

7.55.3.2 __writeRunner()

```
simulatorOpenModelica.ClassSimulatorOpenModelica.__writeRunner (
              self,
              metamodel,
              model,
              outputfiles )  [private]
```

This method writes the openModelica options file.

Referenced by **simulatorOpenModelica.ClassSimulatorOpenModelica.createInput()**.

7.55.3.3 callSolver()

```
simulatorOpenModelica.ClassSimulatorOpenModelica.callSolver (
              self,
              outputfiles )
```

This method calls openModelica to simulate the model

7.55.3.4 createInput()

```
simulatorOpenModelica.ClassSimulatorOpenModelica.createInput (
              self,
              metamodel,
              model,
              outputfiles )
```

This method creates the input files for the openModelica simulator.

References **simulatorOpenModelica.ClassSimulatorOpenModelica.__writeModel()**, and **simulatorOpen↩ Modelica.ClassSimulatorOpenModelica.__writeRunner()**.

7.55.3.5 createOutputs()

```
simulatorOpenModelica.ClassSimulatorOpenModelica.createOutputs (
              self,
              model,
              outputfiles,
              metamodel )

This method converts the format of the openModelica output file into a more adequate for ByoDyn.
```

The documentation for this class was generated from the following file:

- **simulatorOpenModelica.py**

7.56 simulatorRungeKutta.ClassSimulatorRungeKutta Class Reference

Inheritance diagram for simulatorRungeKutta.ClassSimulatorRungeKutta:

Collaboration diagram for simulatorRungeKutta.ClassSimulatorRungeKutta:

Public Member Functions

- **integrate** (self, f, x_0, time, dt)

Public Member Functions inherited from simulatorEuler.ClassSimulatorEuler

- **__init__** (self)
- **run** (self, metamodel, model, outputfiles)

7.56.1 Detailed Description

```
Class for the Runge-Kutta simulator.
It heritages from ClassSimulatorEuler.
```

7.56.2 Member Function Documentation

7.56.2.1 integrate()

```
simulatorRungeKutta.ClassSimulatorRungeKutta.integrate (
              self,
              f,
              x_0,
              time,
              dt )

        This method directs the integration
```

Reimplemented from **simulatorEuler.ClassSimulatorEuler** (p. 158).

Referenced by **simulatorEuler.ClassSimulatorEuler.run()**.

The documentation for this class was generated from the following file:

- **simulatorRungeKutta.py**

7.57 simulatorStochastic.ClassSimulatorStochastic Class Reference

Public Member Functions

- **__init__** (self)
- **run** (self, metamodel, model, outputfiles)
- **simulate** (self, **propensities**, stoichiometryMatrix, x_0, time, dt, **option**, method=None)

Static Public Attributes

- int **cellIndex** = 0
- int **equationNumber** = 0
- **file** = open(outputfiles.integrationInput, 'w')
- str **option** = 'python'
- **output**
- list **propensities** = [reaction.propensity for reaction in model.reactions]
- **squareRootDefinitions** = re.findall('root\(2, [\w\[\]()/\+\-*\s\d\.\^\,]*\)', function.output)

Private Member Functions

- **__getVariables** (self, model)
- **__loadIntegrationFunction** (self, outputfiles)
- **__plotSimulation** (self, model, metamodel, outputfiles, **run**)
- **__storeResults** (self, x, t, model, metamodel, outputfiles, **run**)
- **__writePropensitiesInput** (self, model, outputfiles, initialVariablesValues, variables)

7.57.1 Detailed Description

```
Class for the Stochastic simulators.
```

7.57.2 Constructor & Destructor Documentation

7.57.2.1 __init__()

```
simulatorStochastic.ClassSimulatorStochastic.__init__ (
              self )
```

```
The constructor.
```

7.57.3 Member Function Documentation

7.57.3.1 __getVariables()

```
simulatorStochastic.ClassSimulatorStochastic.__getVariables (
              self,
              model )  [private]
```

```
This private method sets the initial conditions of the system.
```

Referenced by **simulatorEuler.ClassSimulatorEuler.run()**, and **simulatorStochastic.ClassSimulator↩ Stochastic.run()**.

7.57.3.2 __loadIntegrationFunction()

```
simulatorStochastic.ClassSimulatorStochastic.__loadIntegrationFunction (
              self,
              outputfiles )  [private]
```

This private method loads the system of equations, that is the topology of the system.

Referenced by **simulatorEuler.ClassSimulatorEuler.run()**, and **simulatorStochastic.ClassSimulator↩ Stochastic.run()**.

7.57.3.3 __plotSimulation()

```
simulatorStochastic.ClassSimulatorStochastic.__plotSimulation (
              self,
              model,
              metamodel,
              outputfiles,
              run )  [private]
```

This private method creates the graph of the trajectories.

Referenced by **simulatorEuler.ClassSimulatorEuler.run()**, and **simulatorStochastic.ClassSimulator↩ Stochastic.run()**.

7.57.3.4 __storeResults()

```
simulatorStochastic.ClassSimulatorStochastic.__storeResults (
              self,
              x,
              t,
              model,
              metamodel,
              outputfiles,
              run )  [private]
```

This method stores the results of the simulation in a file in the output directory.

Referenced by **simulatorEuler.ClassSimulatorEuler.run()**, and **simulatorStochastic.ClassSimulator↩ Stochastic.run()**.

7.57.3.5 __writePropensitiesInput()

```
simulatorStochastic.ClassSimulatorStochastic.__writePropensitiesInput (
              self,
              model,
              outputfiles,
              initialVariablesValues,
              variables )  [private]
```

This private method writes a file with the topology of the system.

Referenced by **simulatorStochastic.ClassSimulatorStochastic.run()**.

7.57.3.6 run()

```
simulatorStochastic.ClassSimulatorStochastic.run (
              self,
              metamodel,
              model,
              outputfiles )
```

```
This method directs the simulation.
It creates the initial conditions,
loads the system of equations,
simulates the system and
plots the results.
```

References **simulatorEuler.ClassSimulatorEuler.__getVariables()**, **simulatorStochastic.ClassSimulator↩ Stochastic.__getVariables()**, **simulatorEuler.ClassSimulatorEuler.__loadIntegrationFunction()**, **simulator↩ Stochastic.ClassSimulatorStochastic.__loadIntegrationFunction()**, **simulatorStochastic.ClassSimulator↩ Stochastic.__plotSimulation()**, **simulatorEuler.ClassSimulatorEuler.__plotSimulation()**, **simulatorEuler.↩ ClassSimulatorEuler.__storeResults()**, **simulatorStochastic.ClassSimulatorStochastic.__storeResults()**, **simulatorStochastic.ClassSimulatorStochastic.__writePropensitiesInput()**, and **simulatorStochastic.↩ ClassSimulatorStochastic.simulate()**.

7.57.3.7 simulate()

```
simulatorStochastic.ClassSimulatorStochastic.simulate (
              self,
              propensities,
              stoichiometryMatrix,
              x_0,
              time,
              dt,
              option,
              method = None )
```

```
This method simulate the system of equations.
method     describes which method to use
pyfile     python source file where the butcher tableau is found
```

Referenced by **simulatorStochastic.ClassSimulatorStochastic.run()**.

7.57.4 Member Data Documentation

7.57.4.1 cellIndex

```
int simulatorStochastic.ClassSimulatorStochastic.cellIndex = 0  [static]
```

7.57.4.2 equationNumber

```
int simulatorStochastic.ClassSimulatorStochastic.equationNumber = 0  [static]
```

7.57.4.3 file

```
simulatorStochastic.ClassSimulatorStochastic.file = open(outputfiles.integrationInput, 'w')
[static]
```

7.57.4.4 option

```
str simulatorStochastic.ClassSimulatorStochastic.option = 'python'  [static]
```

7.57.4.5 output

```
simulatorStochastic.ClassSimulatorStochastic.output  [static]
```

7.57.4.6 propensities

```
list simulatorStochastic.ClassSimulatorStochastic.propensities = [reaction.propensity for
reaction in model.reactions]  [static]
```

7.57.4.7 squareRootDefinitions

```
simulatorStochastic.ClassSimulatorStochastic.squareRootDefinitions = re.findall('root\(2,
[\w\[\]()/\+\-\*\s\d\.\^\,]*\)', function.output)  [static]
```

The documentation for this class was generated from the following file:

- **simulatorStochastic.py**

7.58 errorMessages.ClassSimulatorStochasticException Class Reference

Inheritance diagram for errorMessages.ClassSimulatorStochasticException:

Collaboration diagram for errorMessages.ClassSimulatorStochasticException:

Public Member Functions

- **printExceptionInfo** (self)

Public Member Functions inherited from errorMessages.ClassByoDynException

- **__init__** (self, **errorString**)

Additional Inherited Members

Public Attributes inherited from errorMessages.ClassByoDynException

- **errorString**

7.58.1 Detailed Description

```
This class deals with the specific errors of the simulatorStochastic module.
```

7.58.2 Member Function Documentation

7.58.2.1 printExceptionInfo()

```
errorMessages.ClassSimulatorStochasticException.printExceptionInfo (
              self )

                This method prints the specific error string corresponding to the module and then the specific
```

Reimplemented from **errorMessages.ClassByoDynException** (p. **??**).

The documentation for this class was generated from the following file:

- **errorMessages.py**

7.59 simulatorXPP.ClassSimulatorXPP Class Reference

Public Member Functions

- **__init__** (self)
- **callSolver** (self, outputfiles)
- **createInput** (self, metamodel, model, outputfiles)
- **createOutputs** (self, model, outputfiles, metamodel)

Private Member Functions

- **__checkNames** (self, model)

7.59.1 Detailed Description

```
Class for the XPP-AUT simulator.
```

7.59.2 Constructor & Destructor Documentation

7.59.2.1 __init__()

```
simulatorXPP.ClassSimulatorXPP.__init__ (
              self )
```

The constructor.

7.59.3 Member Function Documentation

7.59.3.1 __checkNames()

```
simulatorXPP.ClassSimulatorXPP.__checkNames (
              self,
              model )  [private]
```

XPPAUT converts all the strings into capittal letters, so we need to check for repeated nodes.

Referenced by **simulatorXPP.ClassSimulatorXPP.createInput()**.

7.59.3.2 callSolver()

```
simulatorXPP.ClassSimulatorXPP.callSolver (
              self,
              outputfiles )
```

This method calls xppaut to simulate the model

7.59.3.3 createInput()

```
simulatorXPP.ClassSimulatorXPP.createInput (
              self,
              metamodel,
              model,
              outputfiles )
```

This method creates the input file for the XPPAUT simulator.

References **simulatorXPP.ClassSimulatorXPP.__checkNames()**, and **formulas.writeXPPFormula()**.

7.59.3.4 createOutputs()

```
simulatorXPP.ClassSimulatorXPP.createOutputs (
              self,
              model,
              outputfiles,
              metamodel )
```

```
This method converts the format of the XPP output file into a more adequate for ByoDyn.
```

The documentation for this class was generated from the following file:

- **simulatorXPP.py**

7.60 errorMessages.ClassSimulatorXPPException Class Reference

Inheritance diagram for errorMessages.ClassSimulatorXPPException:

Collaboration diagram for errorMessages.ClassSimulatorXPPException:

Public Member Functions

- **printExceptionInfo** (self)

Public Member Functions inherited from errorMessages.ClassByoDynException

- **__init__** (self, **errorString**)

Additional Inherited Members

Public Attributes inherited from errorMessages.ClassByoDynException

- **errorString**

7.60.1 Detailed Description

```
This class deals with the specific errors of the simulator module.
```

7.60.2 Member Function Documentation

7.60.2.1 printExceptionInfo()

```
errorMessages.ClassSimulatorXPPException.printExceptionInfo (
              self )
```

```
This method prints the specific error string corresponding to the module and then the specific
```

Reimplemented from **errorMessages.ClassByoDynException** (p. **??**).

The documentation for this class was generated from the following file:

- **errorMessages.py**

7.61 errorMessages.ClassStarterException Class Reference

Inheritance diagram for errorMessages.ClassStarterException:

Collaboration diagram for errorMessages.ClassStarterException:

Public Member Functions

- **printExceptionInfo** (self)

Public Member Functions inherited from errorMessages.ClassByoDynException

- **__init__** (self, **errorString**)

Additional Inherited Members

Public Attributes inherited from errorMessages.ClassByoDynException

- **errorString**

7.61.1 Detailed Description

```
This class deals with the specific errors of the starter module.
```

7.61.2 Member Function Documentation

7.61.2.1 printExceptionInfo()

```
errorMessages.ClassStarterException.printExceptionInfo (
              self )

                This method prints the specific error string corresponding to the module and then the specific
```

Reimplemented from **errorMessages.ClassByoDynException** (p. **??**).

The documentation for this class was generated from the following file:

- **errorMessages.py**

7.62 testers.ClassStochasticGeneralTest Class Reference

Inheritance diagram for testers.ClassStochasticGeneralTest:

Collaboration diagram for testers.ClassStochasticGeneralTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Additional Inherited Members

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.62.1 Detailed Description

```
Class for testing the general issues for the stochastic simulation.
```

7.62.2 Member Function Documentation

7.62.2.1 analyseResults()

```
testers.ClassStochasticGeneralTest.analyseResults (
              self )
```

```
        This method analyses the results of the stochastic simulation in a general form.
```

References **testers.ClassTest.obtainedResult**, **testers.ClassFitnessFunctionSurfaceTest.obtained↩ Result**, **testers.ClassOEDTest.obtainedResult**, **testers.ClassSciPyTest.obtainedResult**, **testers.Class↩ SensitivityTest.obtainedResult**, **testers.ClassTagFormatTest.obtainedResult**, **testers.ClassTrajectories↩ ReconstructionTest.obtainedResult**, and **testers.ClassWithoutGraphicsTest.obtainedResult**.

Referenced by **testers.ClassTest.run()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.63 testers.ClassStochasticLastStateTest Class Reference

Inheritance diagram for testers.ClassStochasticLastStateTest:

Collaboration diagram for testers.ClassStochasticLastStateTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Additional Inherited Members

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.63.1 Detailed Description

```
Class for testing the results of stochastic histograms.
```

7.63.2 Member Function Documentation

7.63.2.1 analyseResults()

```
testers.ClassStochasticLastStateTest.analyseResults (
              self )
```

```
        This method analyses the histograms resulting from stochastic simulations.
```

References **testers.ClassTest.obtainedResult**, **testers.ClassFitnessFunctionSurfaceTest.obtained↩ Result**, **testers.ClassOEDTest.obtainedResult**, **testers.ClassSciPyTest.obtainedResult**, **testers.Class↩ SensitivityTest.obtainedResult**, **testers.ClassTagFormatTest.obtainedResult**, **testers.ClassTrajectories↩ ReconstructionTest.obtainedResult**, and **testers.ClassWithoutGraphicsTest.obtainedResult**.

Referenced by **testers.ClassTest.run()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.64 testers.ClassStochasticSeparatedGraphsTest Class Reference

Inheritance diagram for testers.ClassStochasticSeparatedGraphsTest:

Collaboration diagram for testers.ClassStochasticSeparatedGraphsTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Additional Inherited Members

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.64.1 Detailed Description

```
Class for testing the results of stochastic simulations rendering separated graphs.
```

7.64.2 Member Function Documentation

7.64.2.1 analyseResults()

```
testers.ClassStochasticSeparatedGraphsTest.analyseResults (
              self )
```

```
        This method for testing the results of stochastic simulaitons rendering separated graphs.
```

References **testers.ClassTest.obtainedResult**, **testers.ClassFitnessFunctionSurfaceTest.obtained↩ Result**, **testers.ClassOEDTest.obtainedResult**, **testers.ClassSciPyTest.obtainedResult**, **testers.Class↩ SensitivityTest.obtainedResult**, **testers.ClassTagFormatTest.obtainedResult**, **testers.ClassTrajectories↩ ReconstructionTest.obtainedResult**, and **testers.ClassWithoutGraphicsTest.obtainedResult**.

Referenced by **testers.ClassTest.run()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.65 testers.ClassStochasticSingleFigureTest Class Reference

Inheritance diagram for testers.ClassStochasticSingleFigureTest:

Collaboration diagram for testers.ClassStochasticSingleFigureTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Additional Inherited Members

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.65.1 Detailed Description

```
Class for testing the results of stochastic simulations creating a single figure.
```

7.65.2 Member Function Documentation

7.65.2.1 analyseResults()

```
testers.ClassStochasticSingleFigureTest.analyseResults (
              self )
```

```
        This method analyses the results of the stochastic simulation creating a single figure.
```

References **testers.ClassTest.obtainedResult**, **testers.ClassFitnessFunctionSurfaceTest.obtained↩Result**, **testers.ClassOEDTest.obtainedResult**, **testers.ClassSciPyTest.obtainedResult**, **testers.Class↩SensitivityTest.obtainedResult**, **testers.ClassTagFormatTest.obtainedResult**, **testers.ClassTrajectories↩ReconstructionTest.obtainedResult**, and **testers.ClassWithoutGraphicsTest.obtainedResult**.

Referenced by **testers.ClassTest.run()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.66 errorMessages.ClassSurfaceException Class Reference

Inheritance diagram for errorMessages.ClassSurfaceException:

Collaboration diagram for errorMessages.ClassSurfaceException:

Public Member Functions

- **printExceptionInfo** (self)

Public Member Functions inherited from errorMessages.ClassByoDynException

- **__init__** (self, **errorString**)

Additional Inherited Members

Public Attributes inherited from errorMessages.ClassByoDynException

- **errorString**

7.66.1 Detailed Description

```
This class deals with the specific errors of the surface module.
```

7.66.2 Member Function Documentation

7.66.2.1 printExceptionInfo()

```
errorMessages.ClassSurfaceException.printExceptionInfo (
              self )
```

```
This method prints the specific error string corresponding to the module and then the specific
```

Reimplemented from **errorMessages.ClassByoDynException** (p. **??**).

The documentation for this class was generated from the following file:

- **errorMessages.py**

7.67 testers.ClassTagFormatTest Class Reference

Inheritance diagram for testers.ClassTagFormatTest:

Collaboration diagram for testers.ClassTagFormatTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Public Attributes

- **obtainedResult**

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.67.1 Detailed Description

```
Class for testing the results of a multicellular simulation using an input model on tag format.
```

7.67.2 Member Function Documentation

7.67.2.1 analyseResults()

```
testers.ClassTagFormatTest.analyseResults (
              self )
```

```
        This method analyses the results of the simulation of a multicellular tag format model.
```

References **testers.ClassTest.obtainedResult**, **testers.ClassFitnessFunctionSurfaceTest.obtainedResult**, **testers.ClassOEDTest.obtainedResult**, **testers.ClassSciPyTest.obtainedResult**, **testers.ClassSensitivityTest.obtainedResult**, **testers.ClassTagFormatTest.obtainedResult**, **testers.ClassTrajectoriesReconstructionTest.obtainedResult**, and **testers.ClassWithoutGraphicsTest.obtainedResult**.

Referenced by **testers.ClassTest.run()**.

7.67.3 Member Data Documentation

7.67.3.1 obtainedResult

```
testers.ClassTagFormatTest.obtainedResult
```

Referenced by **testers.ClassCluster2DTest.analyseResults()**, **testers.ClassCluster3DTest.analyseResults()**, **testers.ClassFigureFormatTest.analyseResults()**, **testers.ClassFitnessFunctionCalculationTest.analyse↩ Results()**, **testers.ClassFitnessFunctionSurfaceTest.analyseResults()**, **testers.ClassGeneticAlgorithm↩ Test.analyseResults()**, **testers.ClassHybridOnePhaseTest.analyseResults()**, **testers.ClassHybridTwo↩ PhasesTest.analyseResults()**, **testers.ClassIdentifiabilityTest.analyseResults()**, **testers.ClassLocal↩ SearchOptimisationTest.analyseResults()**, **testers.ClassNumberOfSimulationsStopperTest.analyse↩ Results()**, **testers.ClassOEDTest.analyseResults()**, **testers.ClassOptionalOutputFormatTest.analyse↩ Results()**, **testers.ClassPlotKeysTest.analyseResults()**, **testers.ClassRandomSearchOptimisationTest.↩ analyseResults()**, **testers.ClassSciPyTest.analyseResults()**, **testers.ClassScoreStopperTest.analyse↩ Results()**, **testers.ClassSensitivityTest.analyseResults()**, **testers.ClassStochasticGeneralTest.analyse↩ Results()**, **testers.ClassStochasticLastStateTest.analyseResults()**, **testers.ClassStochasticSeparated↩ GraphsTest.analyseResults()**, **testers.ClassStochasticSingleFigureTest.analyseResults()**, **testers.↩ ClassTagFormatTest.analyseResults()**, **testers.ClassTrajectoriesReconstructionTest.analyseResults()**, and **testers.ClassTest.dataFilesChecker()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.68 testers.ClassTest Class Reference

Inheritance diagram for testers.ClassTest:

Public Member Functions

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Public Attributes

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

Private Member Functions

- **__execute** (self)

7.68.1 Detailed Description

```
Class for the tests.
```

7.68.2 Constructor & Destructor Documentation

7.68.2.1 __init__()

```
testers.ClassTest.__init__ (
              self )
```

```
        The constructor.
```

Reimplemented in **testers.ClassExportingTest** (p. 80), and **testers.ClassSciPyTest** (p. 146).

7.68.3 Member Function Documentation

7.68.3.1 __execute()

```
testers.ClassTest.__execute (
              self )  [private]
```

```
        This method executes ByoDyn with the specific test.
```

References **testers.ClassTest.id**, **testers.ClassExportingTest.id**, **testers.ClassSciPyTest.id**, **sbmlWorker.ClassRules.id**, **sbmlWorker.ClassEvent.id**, **sbmlWorker.ClassReaction.id**, **sbmlWorker.ClassFunction.id**, **central.main()**, **testers.ClassTest.runner**, **testers.ClassExportingTest.runner**, and **testers.ClassSciPyTest.runner**.

Referenced by **testers.ClassTest.run()**.

7.68.3.2 dataFilesChecker()

```
testers.ClassTest.dataFilesChecker (
              self )
```

```
        This method compares if the numerical outputs of ByoDyn differ on 1 per cent.
```

References **testers.ClassTest.expectedResult**, **testers.ClassOEDTest.expectedResult**, **testers.ClassSciPyTest.expectedResult**, **testers.ClassTrajectoriesReconstructionTest.expectedResult**, **testers.ClassTest.obtainedResult**, **testers.ClassFitnessFunctionSurfaceTest.obtainedResult**, **testers.ClassOEDTest.obtainedResult**, **testers.ClassSciPyTest.obtainedResult**, **testers.ClassSensitivityTest.obtainedResult**, **testers.ClassTagFormatTest.obtainedResult**, **testers.ClassTrajectoriesReconstructionTest.obtainedResult**, and **testers.ClassWithoutGraphicsTest.obtainedResult**.

Referenced by **testers.ClassSciPyTest.analyseResults()**, and **testers.ClassSimulationMethodsTest.analyseResults()**.

7.68.3.3 run()

```
testers.ClassTest.run (
              self )

        This method evaluates if the test ran correctly'
```

References **testers.ClassTest.__execute()**, **testers.ClassCluster2DTest.analyseResults()**, **testers.Class↩ Cluster3DTest.analyseResults()**, **testers.ClassExportingTest.analyseResults()**, **testers.ClassFigure↩ FormatTest.analyseResults()**, **testers.ClassFitnessFunctionCalculationTest.analyseResults()**, **testers.↩ ClassFitnessFunctionSurfaceTest.analyseResults()**, **testers.ClassGeneticAlgorithmTest.analyseResults()**, **testers.ClassHybridOnePhaseTest.analyseResults()**, **testers.ClassHybridTwoPhasesTest.analyseResults()**, **testers.ClassIdentifiabilityTest.analyseResults()**, **testers.ClassLocalSearchOptimisationTest.analyse↩ Results()**, **testers.ClassNumberOfSimulationsStopperTest.analyseResults()**, **testers.ClassOEDTest.↩ analyseResults()**, **testers.ClassOptionalOutputFormatTest.analyseResults()**, **testers.ClassPlotKeys↩ Test.analyseResults()**, **testers.ClassRandomSearchOptimisationTest.analyseResults()**, **testers.Class↩ SciPyTest.analyseResults()**, **testers.ClassScoreStopperTest.analyseResults()**, **testers.ClassSensitivity↩ Test.analyseResults()**, **testers.ClassSeparatedGraphsTest.analyseResults()**, **testers.ClassSimulation↩ MethodsTest.analyseResults()**, **testers.ClassStochasticGeneralTest.analyseResults()**, **testers.Class↩ StochasticLastStateTest.analyseResults()**, **testers.ClassStochasticSeparatedGraphsTest.analyseResults()**, **testers.ClassStochasticSingleFigureTest.analyseResults()**, **testers.ClassTagFormatTest.analyseResults()**, **testers.ClassTrajectoriesReconstructionTest.analyseResults()**, and **testers.ClassWithoutGraphicsTest.↩ analyseResults()**.

7.68.3.4 setOutputName()

```
testers.ClassTest.setOutputName (
              self,
              name )

        This method sets the path to the obtained output that is going to be checked.
```

Referenced by **testers.ClassSciPyTest.analyseResults()**.

7.68.3.5 setResultName()

```
testers.ClassTest.setResultName (
              self,
              name )

        This method sets the output directory of the results of the tests.
```

Referenced by **testers.ClassSciPyTest.analyseResults()**.

7.68.3.6 setRunnerName()

```
testers.ClassTest.setRunnerName (
              self,
              name )

        This method sets the option file for the test.
```

7.68.4 Member Data Documentation

7.68.4.1 expectedResult

`testers.ClassTest.expectedResult`

Referenced by **testers.ClassIdentifiabilityTest.analyseResults()**, **testers.ClassOEDTest.analyseResults()**, **testers.ClassSciPyTest.analyseResults()**, **testers.ClassSensitivityTest.analyseResults()**, **testers.ClassTrajectoriesReconstructionTest.analyseResults()**, and **testers.ClassTest.dataFilesChecker()**.

7.68.4.2 id

`testers.ClassTest.id`

Referenced by **testers.ClassTest.__execute()**.

7.68.4.3 obtainedResult

`testers.ClassTest.obtainedResult`

Referenced by **testers.ClassCluster2DTest.analyseResults()**, **testers.ClassCluster3DTest.analyseResults()**, **testers.ClassFigureFormatTest.analyseResults()**, **testers.ClassFitnessFunctionCalculationTest.analyseResults()**, **testers.ClassFitnessFunctionSurfaceTest.analyseResults()**, **testers.ClassGeneticAlgorithmTest.analyseResults()**, **testers.ClassHybridOnePhaseTest.analyseResults()**, **testers.ClassHybridTwoPhasesTest.analyseResults()**, **testers.ClassIdentifiabilityTest.analyseResults()**, **testers.ClassLocalSearchOptimisationTest.analyseResults()**, **testers.ClassNumberOfSimulationsStopperTest.analyseResults()**, **testers.ClassOEDTest.analyseResults()**, **testers.ClassOptionalOutputFormatTest.analyseResults()**, **testers.ClassPlotKeysTest.analyseResults()**, **testers.ClassRandomSearchOptimisationTest.analyseResults()**, **testers.ClassSciPyTest.analyseResults()**, **testers.ClassScoreStopperTest.analyseResults()**, **testers.ClassSensitivityTest.analyseResults()**, **testers.ClassStochasticGeneralTest.analyseResults()**, **testers.ClassStochasticLastStateTest.analyseResults()**, **testers.ClassStochasticSeparatedGraphsTest.analyseResults()**, **testers.ClassStochasticSingleFigureTest.analyseResults()**, **testers.ClassTagFormatTest.analyseResults()**, **testers.ClassTrajectoriesReconstructionTest.analyseResults()**, and **testers.ClassTest.dataFilesChecker()**.

7.68.4.4 runner

`testers.ClassTest.runner`

Referenced by **testers.ClassTest.__execute()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.69 testers.ClassTrajectoriesReconstructionTest Class Reference

Inheritance diagram for testers.ClassTrajectoriesReconstructionTest:

Collaboration diagram for testers.ClassTrajectoriesReconstructionTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Public Attributes

- **expectedResult**
- **obtainedResult**

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.69.1 Detailed Description

```
Class for testing the trajectories reconstruction function.
```

7.69.2 Member Function Documentation

7.69.2.1 analyseResults()

```
testers.ClassTrajectoriesReconstructionTest.analyseResults (
              self )
```

```
        This method analyses the results of the trajectories reconstruction.
```

References **testers.ClassTest.expectedResult**, **testers.ClassOEDTest.expectedResult**, **testers.Class↩ SciPyTest.expectedResult**, **testers.ClassTrajectoriesReconstructionTest.expectedResult**, **testers.Class↩ Test.obtainedResult**, **testers.ClassFitnessFunctionSurfaceTest.obtainedResult**, **testers.ClassOEDTest.↩ obtainedResult**, **testers.ClassSciPyTest.obtainedResult**, **testers.ClassSensitivityTest.obtainedResult**, **testers.ClassTagFormatTest.obtainedResult**, **testers.ClassTrajectoriesReconstructionTest.obtainedResult**, and **testers.ClassWithoutGraphicsTest.obtainedResult**.

Referenced by **testers.ClassTest.run()**.

7.69.3 Member Data Documentation

7.69.3.1 expectedResult

`testers.ClassTrajectoriesReconstructionTest.expectedResult`

Referenced by **testers.ClassIdentifiabilityTest.analyseResults()**, **testers.ClassOEDTest.analyseResults()**, **testers.ClassSciPyTest.analyseResults()**, **testers.ClassSensitivityTest.analyseResults()**, **testers.ClassTrajectoriesReconstructionTest.analyseResults()**, and **testers.ClassTest.dataFilesChecker()**.

7.69.3.2 obtainedResult

`testers.ClassTrajectoriesReconstructionTest.obtainedResult`

Referenced by **testers.ClassCluster2DTest.analyseResults()**, **testers.ClassCluster3DTest.analyseResults()**, **testers.ClassFigureFormatTest.analyseResults()**, **testers.ClassFitnessFunctionCalculationTest.analyseResults()**, **testers.ClassFitnessFunctionSurfaceTest.analyseResults()**, **testers.ClassGeneticAlgorithmTest.analyseResults()**, **testers.ClassHybridOnePhaseTest.analyseResults()**, **testers.ClassHybridTwoPhasesTest.analyseResults()**, **testers.ClassIdentifiabilityTest.analyseResults()**, **testers.ClassLocalSearchOptimisationTest.analyseResults()**, **testers.ClassNumberOfSimulationsStopperTest.analyseResults()**, **testers.ClassOEDTest.analyseResults()**, **testers.ClassOptionalOutputFormatTest.analyseResults()**, **testers.ClassPlotKeysTest.analyseResults()**, **testers.ClassRandomSearchOptimisationTest.analyseResults()**, **testers.ClassSciPyTest.analyseResults()**, **testers.ClassScoreStopperTest.analyseResults()**, **testers.ClassSensitivityTest.analyseResults()**, **testers.ClassStochasticGeneralTest.analyseResults()**, **testers.ClassStochasticLastStateTest.analyseResults()**, **testers.ClassStochasticSeparatedGraphsTest.analyseResults()**, **testers.ClassStochasticSingleFigureTest.analyseResults()**, **testers.ClassTagFormatTest.analyseResults()**, **testers.ClassTrajectoriesReconstructionTest.analyseResults()**, and **testers.ClassTest.dataFilesChecker()**.

The documentation for this class was generated from the following file:

- **testers.py**

7.70 testers.ClassWithoutGraphicsTest Class Reference

Inheritance diagram for testers.ClassWithoutGraphicsTest:

Collaboration diagram for testers.ClassWithoutGraphicsTest:

Public Member Functions

- **analyseResults** (self)

Public Member Functions inherited from testers.ClassTest

- **__init__** (self)
- **dataFilesChecker** (self)
- **run** (self)
- **setOutputName** (self, name)
- **setResultName** (self, name)
- **setRunnerName** (self, name)

Public Attributes

- **obtainedResult**

Public Attributes inherited from testers.ClassTest

- **expectedResult**
- **id**
- **obtainedResult**
- **runner**

7.70.1 Detailed Description

```
Class for testing the results of the withoutGraphics variable.
```

7.70.2 Member Function Documentation

7.70.2.1 analyseResults()

```
testers.ClassWithoutGraphicsTest.analyseResults (
              self )

        This method analyses the results of a simulation without graphics.
```

Referenced by **testers.ClassTest.run()**.

7.70.3 Member Data Documentation

7.70.3.1 obtainedResult

```
testers.ClassWithoutGraphicsTest.obtainedResult
```

Referenced by **testers.ClassCluster2DTest.analyseResults()**, **testers.ClassCluster3DTest.analyseResults()**, **testers.ClassFigureFormatTest.analyseResults()**, **testers.ClassFitnessFunctionCalculationTest.analyse↩ Results()**, **testers.ClassFitnessFunctionSurfaceTest.analyseResults()**, **testers.ClassGeneticAlgorithm↩ Test.analyseResults()**, **testers.ClassHybridOnePhaseTest.analyseResults()**, **testers.ClassHybridTwo↩ PhasesTest.analyseResults()**, **testers.ClassIdentifiabilityTest.analyseResults()**, **testers.ClassLocal↩ SearchOptimisationTest.analyseResults()**, **testers.ClassNumberOfSimulationsStopperTest.analyse↩ Results()**, **testers.ClassOEDTest.analyseResults()**, **testers.ClassOptionalOutputFormatTest.analyse↩ Results()**, **testers.ClassPlotKeysTest.analyseResults()**, **testers.ClassRandomSearchOptimisationTest.↩ analyseResults()**, **testers.ClassSciPyTest.analyseResults()**, **testers.ClassScoreStopperTest.analyse↩ Results()**, **testers.ClassSensitivityTest.analyseResults()**, **testers.ClassStochasticGeneralTest.analyse↩ Results()**, **testers.ClassStochasticLastStateTest.analyseResults()**, **testers.ClassStochasticSeparated↩ GraphsTest.analyseResults()**, **testers.ClassStochasticSingleFigureTest.analyseResults()**, **testers.↩ ClassTagFormatTest.analyseResults()**, **testers.ClassTrajectoriesReconstructionTest.analyseResults()**, and **testers.ClassTest.dataFilesChecker()**.

The documentation for this class was generated from the following file:

- **testers.py**

Chapter 8

File Documentation

8.1 checker.py File Reference

This module is responsible of running the tests of the program.

Classes

- class **checker.ClassChecker**

Namespaces

- namespace **checker**

Functions

- **checker.main** ()

8.1.1 Detailed Description

This module is responsible of running the tests of the program.

8.2 testers.py File Reference

This module contais the different tests available for ByoDyn.

Classes

- class **testers.ClassCluster2DTest**
- class **testers.ClassCluster3DTest**
- class **testers.ClassExportingTest**
- class **testers.ClassFigureFormatTest**
- class **testers.ClassFitnessFunctionCalculationTest**
- class **testers.ClassFitnessFunctionSurfaceTest**
- class **testers.ClassGeneticAlgorithmTest**
- class **testers.ClassHybridOnePhaseTest**
- class **testers.ClassHybridTwoPhasesTest**
- class **testers.ClassIdentifiabilityTest**
- class **testers.ClassLocalSearchOptimisationTest**
- class **testers.ClassNumberOfSimulationsStopperTest**
- class **testers.ClassOEDTest**
- class **testers.ClassOptionalOutputFormatTest**
- class **testers.ClassPlotKeysTest**
- class **testers.ClassRandomSearchOptimisationTest**
- class **testers.ClassSciPyTest**
- class **testers.ClassScoreStopperTest**
- class **testers.ClassSensitivityTest**
- class **testers.ClassSeparatedGraphsTest**
- class **testers.ClassSimulationMethodsTest**
- class **testers.ClassStochasticGeneralTest**
- class **testers.ClassStochasticLastStateTest**
- class **testers.ClassStochasticSeparatedGraphsTest**
- class **testers.ClassStochasticSingleFigureTest**
- class **testers.ClassTagFormatTest**
- class **testers.ClassTest**
- class **testers.ClassTrajectoriesReconstructionTest**
- class **testers.ClassWithoutGraphicsTest**

Namespaces

- namespace **testers**

8.2.1 Detailed Description

This module contais the different tests available for ByoDyn.

8.3 initiator.py File Reference

ByoDyn is an open source computational package aimed at studying the dynamical behaviour of small to massive biochemical networks.

Namespaces

- namespace **initiator**

Functions

- **initiator.createExamples** ()
- **initiator.initial** (runnerFile=None)
- **initiator.printingHelp** ()
- **initiator.printingVersion** (version)
- **initiator.versionDefinitor** ()

Variables

- str **initiator.benchmarkdir** = os.environ.get('BYODYN_PATH') + '/benchmark'
- str **initiator.libdir** = os.environ.get('BYODYN_PATH') + '/lib'
- str **initiator.srcdir** = os.environ.get('BYODYN_PATH') + '/src'

8.3.1 Detailed Description

ByoDyn is an open source computational package aimed at studying the dynamical behaviour of small to massive biochemical networks.

Models are input in the standard format of systems biology markup language (SBML). The model can be simulated, the sensitivity and the identifiability of the system with respect to the parameters can be analysed and finally kinetic parameters can be estimated using experimental time course data. Several state of the art optimisation algorithms have been implemented for this purpose. ByoDyn can run in parallel for the some of them using MPI.

8.4 config.dox File Reference

8.5 affectors.py File Reference

This is the affectors module.

Namespaces

- namespace **affectors**

Functions

- **affectors.complexExtraBack** (model, file, option, cellIndex, definition, fieldsDefinition)
- **affectors.complexExtraFwd** (model, file, option, cellIndex, definition, fieldsDefinition)
- **affectors.constant** (model, file, option, cellIndex, definition, fieldsDefinition)
- **affectors.constitutive** (model, file, option, cellIndex, definition, fieldsDefinition)
- **affectors.degradation** (model, file, option, cellIndex, definition, fieldsDefinition)
- **affectors.dissociationExtraBack** (model, file, option, cellIndex, definition, fieldsDefinition)
- **affectors.dissociationExtraFwd** (model, file, option, cellIndex, definition, fieldsDefinition)
- **affectors.inhibition** (model, file, option, cellIndex, definition, fieldsDefinition)
- **affectors.NonDimBindingDegradation** (model, file, option, cellIndex, definition, fieldsDefinition)
- **affectors.NonDimConstitutiveDegradation** (model, file, option, cellIndex, definition, fieldsDefinition)

  From here below, non dimension affectors.
- **affectors.NonDimInhibitionDegradation** (model, file, option, cellIndex, definition, fieldsDefinition)
- **affectors.NonDimTranscriptionDegradation** (model, file, option, cellIndex, definition, fieldsDefinition)
- **affectors.NonDimTranslationDegradation** (model, file, option, cellIndex, definition, fieldsDefinition)
- **affectors.NonDimTranslationDegradationBinding** (model, file, option, cellIndex, definition, fields↩ Definition)
- **affectors.SBML** (model, file, option, cellIndex, definition, fieldsDefinition)
- **affectors.transcription** (model, file, option, cellIndex, definition, fieldsDefinition)
- **affectors.translation** (model, file, option, cellIndex, definition, fieldsDefinition)

8.5.1 Detailed Description

This is the affectors module.

Here you will find all the possible biochemical reactions and what they do. This library is prepared to handle python, octave and latex formats.

8.6 MeanShiftCluster.m File Reference

Typedefs

- using **plotFlag** = false

Functions

- add any point within bandWidth to the cluster **beenVisitedFlag** (**myMembers**)
- increment clusters **clustCent** (:, numClust)
- record which points inside **clusterVotes** (mergeWith,:)
- store my members **clusterVotes** (numClust,:)
- mark that these points have been visited ∗∗∗plot stuff ∗∗∗∗if **plotFlag figure** (12345)
- distance from posible new clust max to old clust max if distToOther< bandWidth/2 %if its within bandwidth/2 merge new and old mergeWith= **cN**; **break**;end end if mergeWith > something to merge clustCent(:, merge↩ With)=0.5 ∗(**myMean**+clustCent(record the max as the mean of the two **merged** (I know biased twoards new ones) % **clustMembsCell**
- points within bandWidth **thisClusterVotes** (**inInds**)

Variables

- **bandSq** = bandWidth^2
- center of clust **beenVisitedFlag** = zeros(1,numPts,'uint8')
- smallest size in each dimension **boundBox** = **maxPos**- **minPos**
- end **break**
- mark that these points have been visited ∗∗∗plot stuff ∗∗∗∗if **plotFlag clf**
- when mean has converged **clustCent** = [ ]
- a point belongs to the cluster with the most votes ∗∗∗If they want the cluster2data cell find it for them if nargout **cluster2dataCell** = cell(numClust,1)
- number of points to posibaly use as initilization points **clusterVotes** = zeros(1,numPts,'uint16')
- record the mean **clustMembsCell** {numClust} = **myMembers**
- for **cN**
- number of active points in set **end** [val, data2cluster] = max(**clusterVotes**,[ ],1)
- **function** [clustCent, data2cluster, cluster2dataCell] = MeanShiftCluster(dataPts,bandWidth, **plotFlag**)
- dist squared from mean to all points still active **inInds** = find(**sqDistToAll** < **bandSq**)
- **initPtInds** = 1:numPts
- **maxPos** = max(dataPts,[ ],2)
- biggest size in each dimension **minPos** = min(dataPts,[ ],2)
- use this point as start of mean **myMean** = dataPts(:, **stInd**)
- intilize mean to this points location **myMembers** = [ ]
- add a vote for all the in points belonging to this cluster **myOldMean** = **myMean**
- end ∗∗∗∗Initialize stuff ∗∗∗ **numClust** = 0
- mark that these points have been visited ∗∗∗plot stuff ∗∗∗∗if **plotFlag** hold on if **numDim**
- track if a points been seen already **numInitPts** = numPts
- bounding box size **sizeSpace** = norm(**boundBox**)
- used to resolve conflicts on cluster membership while loop untill convergence **sqDistToAll** = sum((repmat(**myMean**,1,numPts) - dataPts).^2)
- pick a random seed point **stInd** = **initPtInds**(**tempInd**)
- indicator of size of data space **stopThresh** = 1e-3∗bandWidth
- used to resolve conflicts on cluster membership while **numInitPts tempInd** = ceil((**numInitPts**-1e-6)∗rand)
- points that will get added to this cluster **thisClusterVotes** = zeros(1,numPts,'uint16')

8.6.1 Typedef Documentation

8.6.1.1 plotFlag

```
plotFlag = false
```

8.6.2 Function Documentation

8.6.2.1 beenVisitedFlag()

```
add any point within bandWidth to the cluster beenVisitedFlag (
              myMembers  )
```

8.6.2.2 clustCent()

```
increment clusters clustCent (
             :  ,
             numClust  )
```

8.6.2.3 clusterVotes() [1/2]

```
record which points inside clusterVotes (
             mergeWith ,
             :    )
```

8.6.2.4 clusterVotes() [2/2]

```
store my members clusterVotes (
             numClust ,
             :    )
```

8.6.2.5 figure()

```
mark that these points have been visited ***plot stuff ****if  plotFlag figure (
             12345  )
```

8.6.2.6 merged()

```
distance from posible new clust max to old clust max if distToOther< bandWidth/2 %if its within
bandwidth/2 merge new and old mergeWith= cN;  break; end end if mergeWith > something to merge
clustCent(:, mergeWith)=0.5 *( myMean+clustCent(record the max as the mean of the two merged (
             I know biased twoards new ones )  [pure virtual]
```

8.6.2.7 thisClusterVotes()

```
points within bandWidth thisClusterVotes (
              inInds  )
```

8.6.3 Variable Documentation

8.6.3.1 bandSq

```
bandSq = bandWidth^2
```

8.6.3.2 beenVisitedFlag

```
center of clust beenVisitedFlag = zeros(1,numPts,'uint8')
```

8.6.3.3 boundBox

```
smallest size in each dimension boundBox =  maxPos- minPos
```

8.6.3.4 break

```
end break
```

8.6.3.5 clf

```
mark that these points have been visited ***plot stuff ****if  plotFlag clf
```

8.6.3.6 clustCent

```
when mean has converged clustCent = []
```

8.6.3.7 cluster2dataCell

```
cluster2dataCell = cell(numClust,1)
```

8.6.3.8 clusterVotes

```
number of points to posibaly use as initilization points clusterVotes = zeros(1,numPts,'uint16')
```

8.6.3.9 clustMembsCell

```
record the mean clustMembsCell {numClust} =  myMembers
```

8.6.3.10 cN

for cN

Initial value:

```
= 1:numClust
                distToOther = norm(myMean-clustCent(:,cN))
```

8.6.3.11 end

number of active points in set end[val, data2cluster] = max(**clusterVotes**,[],1)

8.6.3.12 function

function[clustCent, data2cluster, cluster2dataCell] = MeanShiftCluster(dataPts,bandWidth, **plotFlag**)

8.6.3.13 inInds

dist squared from mean to all points still active inInds = find(**sqDistToAll** < **bandSq**)

8.6.3.14 initPtInds

end end initPtInds = 1:numPts

8.6.3.15 maxPos

maxPos = max(dataPts,[],2)

8.6.3.16 minPos

biggest size in each dimension minPos = min(dataPts,[],2)

8.6.3.17 myMean

save the old mean myMean = dataPts(:, **stInd**)

8.6.3.18 myMembers

compute the new mean myMembers = []

8.6.3.19 myOldMean

add a vote for all the in points belonging to this cluster myOldMean = **myMean**

8.6.3.20 numClust

add these votes to the **merged** cluster else its a new cluster numClust = 0

8.6.3.21 numDim

mark that these points have been visited ***plot stuff ****if **plotFlag** hold on if numDim

Initial value:

```
== 2
                plot(dataPts(1,:),dataPts(2,:),'.')
                plot(dataPts(1,myMembers),dataPts(2,myMembers),'ys')
                plot(myMean(1),myMean(2),'go')
                plot(myOldMean(1),myOldMean(2),'rd')
                pause
            end
        end

        %**** if mean doesn't move much stop this cluster ***
        if norm(myMean-myOldMean) < stopThresh

            %check for merge posibilities
            mergeWith = 0
```

8.6.3.22 numInitPts

we can initialize with any of the points not yet visited numInitPts = numPts

8.6.3.23 sizeSpace

bounding box size sizeSpace = norm(**boundBox**)

8.6.3.24 sqDistToAll

used to resolve conflicts on cluster membership while loop untill convergence sqDistToAll = sum((repmat(**myMean**,1,numPts) - dataPts).^2)

8.6.3.25 stInd

pick a random seed point stInd = **initPtInds**(**tempInd**)

8.6.3.26 stopThresh

indicator of size of data space stopThresh = 1e-3*bandWidth

8.6.3.27 tempInd

```
used to resolve conflicts on cluster membership while  numInitPts tempInd = ceil( ( numInit↩
Pts-1e-6)*rand)
```

8.6.3.28 thisClusterVotes

```
points that will get added to this cluster thisClusterVotes = zeros(1,numPts,'uint16')
```

8.7 formulas.py File Reference

This module contains different functions necessary to interconvert the formula string formats.

Namespaces

- namespace **formulas**

Functions

- **formulas.checkBrackets** (formula)
- **formulas.formatPowers** (formula, option)
- **formulas.formulaLatex** (xmlFormula, file, model)
- **formulas.getMathExpression** (ASTNode)
- **formulas.includeFunctions** (model, formula)
- **formulas.piecewise** (a, b, c)
- **formulas.readWriteFormula** (model, file, cellIndex, option, formula)
- **formulas.replaceConstants** (model, constant)
- **formulas.solveFormula** (model, formula)
- **formulas.translateMathFactor** (mathFactor)
- **formulas.writeOpenModelicaFormula** (formula, file)
- **formulas.writeXPPFormula** (formula, file)

8.7.1 Detailed Description

This module contains different functions necessary to interconvert the formula string formats.

8.8 localSearch.f File Reference

Functions/Subroutines

- subroutine **calca** (n, p, x, nf, r)
- subroutine **principal** (x, b, s)

8.8.1 Function/Subroutine Documentation

8.8.1.1 calca()

```
subroutine calca (
             integer n,
             integer p,
             double precision, dimension(p) x,
             integer nf,
             double precision, dimension(n) r )
```

Referenced by **principal()**.

8.8.1.2 principal()

```
subroutine principal (
             double precision, dimension(xxdim) x,
             double precision, dimension(2, xxdim) b,
             double precision s )
```

References **calca()**.

8.9 simulatorStochasticTables.py File Reference

This module contains the required Butcher table for the stochastic Runge-Kutta methods.

Namespaces

- namespace **simulatorStochasticTables**

Variables

- dict **simulatorStochasticTables.ButcherTableau**

8.9.1 Detailed Description

This module contains the required Butcher table for the stochastic Runge-Kutta methods.

8.10 gssa.py File Reference

This module is contains the the stochastic simulation algorithms for the Guillespie Stochastic Simulation Algorithm (SSA).

Namespaces

- namespace **gssa**

Functions

- **gssa.simulate** (evalPropensities, stoichiometry, x_0, time, hurdle, seed=None)

8.10.1 Detailed Description

This module is contains the the stochastic simulation algorithms for the Guillespie Stochastic Simulation Algorithm (SSA).

8.11 tauleap.py File Reference

This module is contains the the stochastic simulation algorithms for the tau-leap method.

Namespaces

- namespace **tauleap**

Functions

- **tauleap.simulate** (evalPropensities, stoichiometry, x_0, time, tau, seed=None)

8.11.1 Detailed Description

This module is contains the the stochastic simulation algorithms for the tau-leap method.

8.12 README.md File Reference

8.13 central.py File Reference

This module is the first module called from the initiator.

Classes

- class **central.ClassFile**

Namespaces

- namespace **central**

Functions

- **central.main** (runnerFile)

  Main Program #.
- **central.modelReader** (metamodel)

  Main Functions #

- **central.optionReader** (runnerFile)
- **central.profile** ()
- **central.runner** (metamodel, model)
- **central.sbmlReader** (metamodel)
- **central.tagsReader** (metamodel)

8.13.1 Detailed Description

This module is the first module called from the initiator.

This module contains the code to direct all the jobs of the execution.

8.14 centralFunctions.py File Reference

This module holds the central functions of the program.

Namespaces

- namespace **centralFunctions**

Functions

- **centralFunctions.callingOctave** (outputfiles)
- **centralFunctions.callingPython** (outputfiles)
- **centralFunctions.createInputOctave** (metamodel, model, outputfiles)
- **centralFunctions.createInputPython** (metamodel, model, outputfiles)
- **centralFunctions.createOctaveOutputs** (model, metamodel, outputfiles)
- **centralFunctions.matlabIntegrator** (metamodel, model, outputfiles)
- **centralFunctions.neighboursFinder** (model, cellIndex)
- **centralFunctions.octaveIntegration** (metamodel, model, outputfiles)
- **centralFunctions.pythonIntegration** (metamodel, model, outputfiles)
- **centralFunctions.writeFormulaOctave** (model, node, octave, option, cellIndex)
- **centralFunctions.writeInitialConditionsOctave** (model, metamodel, octave, **function**)
- **centralFunctions.writeParametersOctave** (model, octave)

8.14.1 Detailed Description

This module holds the central functions of the program.

By central functions we mean functions that may be required by either during the simulation, the sensitivity analysis or the optimisation.

8.15 cluster.py File Reference

This module is responsible for clustering.

Namespaces

- namespace **cluster**

Functions

- **cluster.dataTransformer** (metamodel, outputfiles)
- **cluster.defaultRunner** (clusteringRange, metamodel, outputfiles)
- **cluster.main** (metamodel, outputfiles)
- **cluster.octaveCodeWriter** (metamodel, outputfiles)
- **cluster.octaveExecuter** (outputfiles)
- **cluster.plotter** (outputfiles, dimension, labels)
- **cluster.resolutionChecker** (metamodel, outputfiles)
- **cluster.surfacePlotter** (outputfiles, labels)
- **cluster.volumePlotter** (outputfiles, labels)

8.15.1 Detailed Description

This module is responsible for clustering.

The method used is mean shift. We used a matlab implementation from Bryan Feldman.

8.16 dynamicsReconstructer.py File Reference

This module reconstructs the dynamics of a model given the parameter values.

Namespaces

- namespace **dynamicsReconstructer**

Functions

- **dynamicsReconstructer.central** (metamodel, model, outputfiles)
- **dynamicsReconstructer.initialConditionsDetector** (metamodel, model)
- **dynamicsReconstructer.modelDetermination** (model, simulation, solutions, initialConditions)
- **dynamicsReconstructer.parametersDetector** (metamodel, model)
- **dynamicsReconstructer.plotter** (solutions, model, outputfiles, metamodel)
- **dynamicsReconstructer.run** (model, metamodel, outputfiles)
- **dynamicsReconstructer.runner** (metamodel, model, outputfiles, solutions, initialConditions)
- **dynamicsReconstructer.storeInfo** (outputfiles, simulation, model)

8.16.1 Detailed Description

This module reconstructs the dynamics of a model given the parameter values.

8.17 errorMessages.py File Reference

This module deals with the error handling of the program.

Classes

- class **errorMessages.ClassByoDynException**
- class **errorMessages.ClassCentralException**
- class **errorMessages.ClassCheckerException**
- class **errorMessages.ClassClusterException**
- class **errorMessages.ClassDynamicsReconstructerException**
- class **errorMessages.ClassFormulasException**
- class **errorMessages.ClassIdentifiabilityAnalyzerException**
- class **errorMessages.ClassInitiatorException**
- class **errorMessages.ClassMatrixWorkerException**
- class **errorMessages.ClassOptimalExperimentalDesignException**
- class **errorMessages.ClassOptimiserException**
- class **errorMessages.ClassSamplerException**
- class **errorMessages.ClassSBMLWorkerException**
- class **errorMessages.ClassSensitivityAnalyzerException**
- class **errorMessages.ClassSimulatorEulerException**
- class **errorMessages.ClassSimulatorException**
- class **errorMessages.ClassSimulatorStochasticException**
- class **errorMessages.ClassSimulatorXPPException**
- class **errorMessages.ClassStarterException**
- class **errorMessages.ClassSurfaceException**

Namespaces

- namespace **errorMessages**

Functions

- **errorMessages.ErrorHandler** (type, value, traceback)

8.17.1 Detailed Description

This module deals with the error handling of the program.

8.18 exporter.py File Reference

This module contains the code for exporting SBML files.

Namespaces

- namespace **exporter**

Functions

- **exporter.central** (metamodel, model, outputfiles)

8.18.1 Detailed Description

This module contains the code for exporting SBML files.

8.19 fitnessFunctionEvaluator.py File Reference

This module calculates the fitness function value given an experimental data set for a specific model.

Namespaces

- namespace **fitnessFunctionEvaluator**

Functions

- **fitnessFunctionEvaluator.central** (model, metamodel, outputfiles)
- **fitnessFunctionEvaluator.scoreWriter** (score, outputfiles)

8.19.1 Detailed Description

This module calculates the fitness function value given an experimental data set for a specific model.

8.20 identifiabilityAnalyzer.py File Reference

This module contains the algorithms necessary for the analysis of identifiability of a given model.

Classes

- class **identifiabilityAnalyzer.ClassIdentifiabilityAnalyzer**

Namespaces

- namespace **identifiabilityAnalyzer**

Functions

- **identifiabilityAnalyzer.central** (model, metamodel, outputfiles)

8.20.1 Detailed Description

This module contains the algorithms necessary for the analysis of identifiability of a given model.

8.21 localOptimiser.py File Reference

This module minimises the fitness function based on the Fortran program dn2fb from the PORT Mathematical Subroutine Library.

Namespaces

- namespace **localOptimiser**

Functions

- **localOptimiser.central** (model, metamodel, outputfiles)
- **localOptimiser.globalDefiner** (model, metamodel, outputfiles)
- **localOptimiser.locatingModuleDir** (metamodel)
- **localOptimiser.optimisation** (threshold)
- **localOptimiser.pcalcr** (x, p)
- **localOptimiser.savingResults** (value, model, metamodel, outputfiles)

8.21.1 Detailed Description

This module minimises the fitness function based on the Fortran program dn2fb from the PORT Mathematical Subroutine Library.

8.22 matrixWorker.py File Reference

Classes

- class **matrixWorker.ClassMatrix**

Namespaces

- namespace **matrixWorker**

8.23 optimalExperimentalDesign.py File Reference

This module contains the algorithms necessary for the optimal experimental design protocols of a given model.

Namespaces

- namespace **optimalExperimentalDesign**

Functions

- **optimalExperimentalDesign.addNewPoint** (identifiability, sensitivity, metamodel, model, outputfiles, parametersToStudy)
- **optimalExperimentalDesign.central** (model, metamodel, outputfiles)
- **optimalExperimentalDesign.chooseValue** (criteria, values)
- **optimalExperimentalDesign.getCriteria** (criteria, identifiability)
- **optimalExperimentalDesign.rankTargets** (identifiability, sensitivity, metamodel, model, outputfiles, parametersToStudy)
- **optimalExperimentalDesign.timePointsDeterminer** (metamodel)

8.23.1 Detailed Description

This module contains the algorithms necessary for the optimal experimental design protocols of a given model.

8.24 optimiser.py File Reference

This module is responsible of the optimisation of fitness function.

Namespaces

- namespace **optimiser**

Functions

- **optimiser.calculateConfidenceIntervals** (model, metamodel, outputfiles)
- **optimiser.central** (metamodel, model, outputfiles, solutions)
- **optimiser.checkDataPoints** (model, metamodel)
- **optimiser.checkInitialConditionsToVary** (model, metamodel)
- **optimiser.checkParametersToVary** (model, metamodel)
- **optimiser.checkTargetNodes** (model, metamodel)
- **optimiser.evaluateIteration** (karyotype, model, metamodel, outputfiles, scores, iteration)
- **optimiser.evaluateSolution** (iteration, score, metamodel, model, outputfiles)
- **optimiser.evaluateStopping** (metamodel, iteration, scores)
- **optimiser.gaInitialisePopulation** (model, metamodel)
- **optimiser.gaNaturalSelection** (karyotype, iteration, scores, metamodel, outputfiles)
- **optimiser.gaPopulationScoresObtainer** (iteration, karyotype, metamodel, model, outputfiles)
- **optimiser.gaPopulationScoresObtainerLocal** (iteration, karyotype, metamodel, model, outputfiles)
- **optimiser.gaSex** (karyotype, bestChromosomes, sexChromosomes, model, metamodel)
- **optimiser.getSimulateValue** (originalSimulationValues, node, target, model, metamodel)
- **optimiser.linearVariation** (minimalValue, maximalValue)
- **optimiser.logarithmicVariation** (minimalValue, maximalValue)
- **optimiser.normalVariationForInitialCondition** (metamodel, model)
- **optimiser.randomSearch** (metamodel, model, outputfiles)
- **optimiser.scoreObtainer** (metamodel, model, outputfiles)
- **optimiser.storeSolution** (score, metamodel, model, workingDirectory)

8.24.1 Detailed Description

This module is responsible of the optimisation of fitness function.

It directs the flow of the program to the genetic algorithm or the local search. It contains most of the parallel code.

8.25 parallel.py File Reference

Namespaces

- namespace **parallel**

Functions

- **parallel.currentProcessor** ()
- **parallel.mainProcessor** ()
- **parallel.receive** (Source, Tag)
- **parallel.receiveAny** (Tag)
- **parallel.runsOnParallelMachine** ()
- **parallel.send** (Object, Destination, Tag)
- **parallel.sendAll** (Object, Tag)
- **parallel.totalProcessors** ()
- **parallel.waitProcessors** ()

Variables

- **parallel.communicator** = None
- dict **parallel.internalBuffer** = {}

8.26 pca.py File Reference

Classes

- class **pca.ClassPCA**

Namespaces

- namespace **pca**

Functions

- **pca.central** ()

8.27 profiler.py File Reference

This module profiles the program.

Namespaces

- namespace **profiler**

Functions

- **profiler.main** ()

Variables

- str **profiler.ERROR_FILE** = '%s/profilingErrorFile' % str(**scratchDir**)
- str **profiler.PROFILE_FILE** = '%s/profilingOutputFile' % str(**scratchDir**)
- **profiler.scratchDir** = os.environ.get('BYODYN_SCRATCH_DIR')
- **profiler.SCRIPTNAME** = central
- str **profiler.STAT_FILE** = '%s/profilingStatFile' % str(**scratchDir**)

8.27.1 Detailed Description

This module profiles the program.

In order to run use Profile.py [sciptname] [scriptargs]

8.28 sampler.py File Reference

This module is responsible of sample methods of fitness function surface.

Classes

- class **sampler.ClassSample**
- class **sampler.ClassSamplerMonteCarlo**

Namespaces

- namespace **sampler**

Functions

- **sampler.central** (model, metamodel, outputfiles)
- **sampler.plot1D** (metamodel, outputfiles, sample)
- **sampler.plot2D** (metamodel, outputfiles, sample)
- **sampler.plot3D** (metamodel, outputfiles, sample)
- **sampler.plotResults** (metamodel, outputfiles, sample)
- **sampler.rainbowColourCalculator** (value)
- **sampler.storeResults** (listOfSample, file, metamodel)

8.28.1 Detailed Description

This module is responsible of sample methods of fitness function surface.

8.29 sbmlWorker.py File Reference

This module is dedicated to the parsing of the models in SBML.

Classes

- class **sbmlWorker.ClassEvent**
- class **sbmlWorker.ClassFunction**
- class **sbmlWorker.ClassModelSBML**
- class **sbmlWorker.ClassReaction**
- class **sbmlWorker.ClassRules**

Namespaces

- namespace **sbmlWorker**

Functions

- **sbmlWorker.divCompartment** (sbmlModel, formula, node)
- **sbmlWorker.getBooleanExpression** (ASTNode)
- **sbmlWorker.sbmlWriter** (w, model, metamodel, file)
- **sbmlWorker.translateLocalParametersNamesInFormula** (formula, reaction)
- **sbmlWorker.translateMathFactor** (mathFactor)

8.29.1 Detailed Description

This module is dedicated to the parsing of the models in SBML.

All the information that ByoDyn requires is the object called "model".

8.30 sensitivityAnalyzer.py File Reference

This module contains the algorithms necessary for the analysis of sensitivity of a given model.

Classes

- class **sensitivityAnalyzer.ClassSensitivityAnalyzer**

Namespaces

- namespace **sensitivityAnalyzer**

Functions

- **sensitivityAnalyzer.central** (model, metamodel, outputfiles)
- **sensitivityAnalyzer.componentsChecker** (model, metamodel)
- **sensitivityAnalyzer.getNodeName** (value, model)
- **sensitivityAnalyzer.getPositionNode** (field, model)

8.30.1 Detailed Description

This module contains the algorithms necessary for the analysis of sensitivity of a given model.

8.31 simulator.py File Reference

This module is responsible of the simulation of the model.

Classes

- class **simulator.ClassEpithelium**

Namespaces

- namespace **simulator**

Functions

- **simulator.central** (metamodel, model, outputfiles)
- **simulator.checkResults** (metamodel, trajectoriesFile)
- **simulator.compatibilityChecker** (metamodel, model)
- **simulator.createEpiplot** (model, metamodel, outputfiles)
- **simulator.createGnuplot** (model, metamodel, outputfiles, type)
- **simulator.createPDFFormulae** (model, metamodel, outputfiles)
- **simulator.eventsDealer** (metamodel, model, outputfiles)
- **simulator.flatFileWriter** (outputfiles, grid)
- **simulator.isAGene** (node)
- **simulator.lastStateRetriever** (metamodel, model, outputfiles)
- **simulator.obtainSimulationValues** (metamodel, model, outputfiles, type)
- **simulator.plotMaker** (outputfiles, grid, model)
- **simulator.setIntegrationOption** (model, tester)

8.31.1 Detailed Description

This module is responsible of the simulation of the model.

8.32 simulatorEuler.py File Reference

This module is responsible of the numerical integrations functions and the Euler method.

Classes

- class **simulatorEuler.ClassSimulatorEuler**

Namespaces

- namespace **simulatorEuler**

8.32.1 Detailed Description

This module is responsible of the numerical integrations functions and the Euler method.

8.33 simulatorOpenModelica.py File Reference

This module simulates the model in the case the integration option OpenModelica has been selected.

Classes

- class **simulatorOpenModelica.ClassSimulatorOpenModelica**

Namespaces

- namespace **simulatorOpenModelica**

8.33.1 Detailed Description

This module simulates the model in the case the integration option OpenModelica has been selected.

8.34 simulatorRungeKutta.py File Reference

This module is responsible of the Runge-Kutta method for the numerical integrations.

Classes

- class **simulatorRungeKutta.ClassSimulatorRungeKutta**

Namespaces

- namespace **simulatorRungeKutta**

8.34.1 Detailed Description

This module is responsible of the Runge-Kutta method for the numerical integrations.

8.35 simulatorStochastic.py File Reference

This module is contains the code for the stochastic simulation algorithms.

Classes

- class **simulatorStochastic.ClassSimulatorStochastic**

Namespaces

- namespace **simulatorStochastic**

Variables

- **simulatorStochastic.stochasticdir** = os.path.join(os.environ.get('BYODYN_PATH'), 'lib', 'stochastic')

8.35.1 Detailed Description

This module is contains the code for the stochastic simulation algorithms.

8.36 simulatorXPP.py File Reference

This module simulates the model in the case the integration option xpp has been selected.

Classes

- class **simulatorXPP.ClassSimulatorXPP**

Namespaces

- namespace **simulatorXPP**

Functions

- **simulatorXPP.convertName** (name)
- **simulatorXPP.replaceNames** (expr, names, newNames, conflictives)

8.36.1 Detailed Description

This module simulates the model in the case the integration option xpp has been selected.

8.37 starter.py File Reference

This module is the parser for the running options of ByoDyn.

Classes

- class **starter.ClassMetaModel**

Namespaces

- namespace **starter**

Functions

- **starter.central** (runnerFile)
- **starter.fortranModulesCreator** (metamodel)
- **starter.incompatibilityChecker** (metamodel)

8.37.1 Detailed Description

This module is the parser for the running options of ByoDyn.

8.38 surface.py File Reference

This module builds the fitness function surface for combination of 2 parameters.

Namespaces

- namespace **surface**

Functions

- **surface.central** (metamodel, model, outputfiles)
- **surface.checkSurfaceParameters** (metamodel, model)
- **surface.rainbowColorCalculation** (value, minValue, maxValue)
- **surface.surfacePlotter** (metamodel, outputfiles, plot, surface, xGrid, yGrid)
- **surface.surfaceTextSaver** (outputfiles, metamodel, plot, surface)

8.38.1 Detailed Description

This module builds the fitness function surface for combination of 2 parameters.

8.39 tagParser.py File Reference

This module is the paser for the tag format files.

Classes

- class **tagParser.ClassModelTags**

Namespaces

- namespace **tagParser**

8.39.1 Detailed Description

This module is the paser for the tag format files.

A system model is obtained.