AMIE configuration documentation

Alain B. Giorla

December 12, 2014

This document shows how to write initiation files for AMIE. These files can be used to set up variables in simulations without re-compiling the main executable. An example of initiation file usage in AMIE can be found in the `main_2d_composite` example.

Contents

1 First-level objects **4**

2 Behaviour **4**
2.1 Common parameters 4
2.2 Elastic behaviour 5
2.3 Elastic and damage behaviour 5
2.4 Elastic and imposed deformation behaviour 5
2.5 Viscous behaviour 5
2.6 Kelvin-Voigt viscoelastic behaviour 5
2.7 Maxwell viscoelastic behaviour 6
2.8 Burger viscoelastic behaviour 6
2.9 Generalized Kelvin-Voigt viscoelastic behaviour 6
2.10 Generalized Maxwell viscoelastic behaviour 6
2.11 Predefined cement paste behaviour 6
2.12 Predefined aggregate behaviour 7
2.13 Predefined alkali-silica reaction gel behaviour 7
2.14 Predefined concrete behaviour 8
2.15 Predefined rebar behaviour 8
2.16 Predefined steel behaviour 8
2.17 Generalized logarithmic creep behaviour 9

3 Boundary Conditions **9**
3.1 Common parameters 9
3.2 Constant boundary conditions 10
3.3 Time-dependent boundary conditions 10

4 Boundary Condition Time Evolution **10**

5 Bounding Box Restriction **11**

6 Damage Model **11**
6.1 Common parameters 11
6.2 Isotropic Linear Damage 11
6.3 Isotropic Incremental Linear Damage 11
6.4 Plastic Strain 12

7 Discretization 12

8 Export 12

9 Fracture Criterion 12

9.1 Common parameters 12
9.2 Maximum tensile strain criterion 13
9.3 Maximum tensile stress criterion 13
9.4 Maximum tensile strain criterion with linear softening 13
9.5 Maximum tensile stress criterion with ellipsoidal softening 13
9.6 Mohr-Coulomb criterion 14
9.7 Linear softening Mohr-Coulomb criterion 14
9.8 Exponential softening Mohr-Coulomb criterion 14
9.9 Von Mises stress criterion 14
9.10 Modified Compressive Field Theory criterion 15
9.11 Space-time asymmetric multi-linear softening fracture criterion 15

10 Geometry 15

11 Inclusions 16

12 Inclusion Output 17

13 Material Parameters 17

13.1 Elastic behaviour 17
13.2 Creep behaviour 18
13.3 Imposed deformation 18
13.4 Damage 18
13.5 Additional parameters 19

14 Material Law 20

14.1 Common parameters 20
14.2 Thermal expansion 20
14.3 Drying shrinkage 20
14.4 Arrhenius law 20
14.5 Arrhenius law for the creep parameters 21
14.6 Relative humidity effect for the creep parameters 21
14.7 Material law function of the space and time coordinates 21
14.8 Material law function of a single parameter 21
14.9 Material law function of a set of parameters 22
14.10 Linearly interpolated material law 22
14.11 Assignment material law 23
14.12 Minimum material law 23
14.13 Maximum material law 23
14.14 Field extractor material law 23
14.15 Time derivative material law 24
14.16 Time integral material law 24

15 Output 24

16 Parallelism 25

17 Particle Size Distribution 25
17.1 Common parameters . . . 25
17.2 Analytic particle size distribution . . . 25
17.3 Particle size distribution from file . . . 26
17.4 Pregenerated inclusions . . . 26
17.5 Manually placed inclusion . . . 26
17.6 Concentric inclusions . . . 26

18 Particle Size Distribution Cutoff 27

19 Placement 27

20 Point 27

21 Rebar 28

22 Sample 28

23 Stepping 28
23.1 Common parameters . . . 28
23.2 Constant time step . . . 28
23.3 Logarithmic time step . . . 29
23.4 Function-defined time step . . . 29
23.5 File-defined time steps . . . 29

24 Time step output 29

25 Viscoelastic Unit 29

26 Enumerations 30
26.1 Behaviour Type . . . 30
26.2 Boolean . . . 30
26.3 Boundary Condition Type . . . 30
26.4 Bounding Box Position . . . 31
26.5 Column Identifier . . . 33
26.6 Damage Model Type . . . 33
26.7 Element Order . . . 33
26.8 Extended Field Type . . . 34
26.9 Field Type . . . 34
26.10 Fracture Criterion Type . . . 35
26.11 Geometry Type . . . 35
26.12 Material Law Type . . . 35
26.13 Particle Size Distribution Type . . . 36
26.14 Particle Size Distribution Specification Type . . . 36
26.15 Sampling Restriction . . . 36
26.16 Time Step Selection . . . 36

27 History 37

1 First-level objects

Required parameters

Object	Type	Description
`.sample`	Sample	Box in which the simulation is performed.
`.discretization`	Discretization	Parameters for the mesh generation.
`.stepping`	Stepping	Parameters for the time-stepping and damage iterations.
`.output`	Output	Exports specified field values in a table format.
`.export`	Export	Exports the mesh in SVG files.

Optional parameters

Object	Type	Description
`.inclusions`	Inclusions	Defines the geometry and mechanical behaviour of the inclusions.
`.boundary_condition`	BoundaryCondition	Mechanical boundary conditions applied to the sample.
`.parallel`	Parallelism	Manages parallel computation.

2 Behaviour

There are many types of mechanical behaviour available in AMIE. The main flag which differentiates them is:
`.behaviour`
`..type`

2.1 Common parameters

The following parameters can be used for all mechanical behaviours.

Required parameters

Object	Type	Description
`.type`	BehaviourType	Defines the general type of mechanical behaviour.

Optional parameters

Object	Type	Description
`.additional_viscoelastic_variables`	Numeral	In the case of a space-time finite element analysis, this indicates the number of unused visco-elastic displacement fields must be added to the model. The total number of used and unused visco-elastic displacement fields must be equal between all the Behaviour defined in the simulation.

Note: in the following, each time the pair `.young_modulus/.poisson_ratio` is found, it can be replaced with the pair `.bulk_modulus/.shear_modulus`, unless specified otherwise.

2.2 Elastic behaviour

Required parameters

Object	Type	Description
`.type`	BehaviourType	`ELASTICITY`
`.young_modulus`	Numeral	Young's modulus of the material (in pascal).
`.poisson_ratio`	Numeral	Poisson ratio of the material (arbitrary unit).

2.3 Elastic and damage behaviour

Required parameters

Object	Type	Description
`.type`	BehaviourType	`ELASTICITY_AND_FRACTURE`
`.fracture_criterion`	FractureCriterion	Defines the failure surface and post-peak behaviour of the material.
`.damage_model`	DamageModel	Defines the damage algorithm used.
`.young_modulus`	Numeral	Young's modulus of the material (in pascal).
`.poisson_ratio`	Numeral	Poisson ratio of the material (arbitrary unit).

2.4 Elastic and imposed deformation behaviour

Required parameters

Object	Type	Description
`.type`	BehaviourType	`ELASTICITY_AND_FRACTURE`
`.young_modulus`	Numeral	Young's modulus of the material (in pascal).
`.poisson_ratio`	Numeral	Poisson ratio of the material (arbitrary unit).
`.imposed_deformation`	Numeral	Imposed deformation of the material (in meter/meter).

2.5 Viscous behaviour

Required parameters

Object	Type	Description
`.type`	BehaviourType	`VISCOSITY`
`.young_modulus`	Numeral	Young's modulus of the material (in pascal).
`.poisson_ratio`	Numeral	Poisson ratio of the material (arbitrary unit).
`.characteristic_time`	Numeral	Characteristic time of the dashpot unit (in days).

2.6 Kelvin-Voigt viscoelastic behaviour

Required parameters

Object	Type	Description
`.type`	BehaviourType	`KELVIN_VOIGT`
`.young_modulus`	Numeral	Young's modulus of the material (in pascal).
`.poisson_ratio`	Numeral	Poisson ratio of the material (arbitrary unit).
`.characteristic_time`	Numeral	Characteristic time of the dashpot unit (in days).

2.7 Maxwell viscoelastic behaviour

Required parameters

Object	Type	Description
`.type`	BehaviourType	`MAXWELL`
`.young_modulus`	Numeral	Young's modulus of the material (in pascal).
`.poisson_ratio`	Numeral	Poisson ratio of the material (arbitrary unit).
`.characteristic_time`	Numeral	Characteristic time of the dashpot unit (in days).

2.8 Burger viscoelastic behaviour

Required parameters

Object	Type	Description
`.type`	BehaviourType	`BURGER`
`.kelvin_voigt`	ViscoelasticUnit	Deredefined steelscribes the Kelvin-Voigt section of the behaviour.
`.maxwell`	ViscoelasticUnit	Describes the Maxwell section of the behaviour.

2.9 Generalized Kelvin-Voigt viscoelastic behaviour

Required parameters

Object	Type	Description
`.type`	BehaviourType	`GENERALIZED_KELVIN_VOIGT`
`.first_branch`	ViscoelasticUnit	Describes the initial elastic stiffness of the visco-elastic model.

Optional parameters

Object	Type	Description
`.branch`	ViscoelasticUnit	Describes each Kelvin-Voigt unit in the model. The model can have as many branches as required.

2.10 Generalized Maxwell viscoelastic behaviour

Required parameters

Object	Type	Description
`.type`	BehaviourType	`GENERALIZED_MAXWELL`
`.first_branch`	ViscoelasticUnit	Describes the final relaxed stiffness of the visco-elastic model.

Optional parameters

Object	Type	Description
`.branch`	ViscoelasticUnit	Describes each Maxwell unit in the model. The model can have as many branches as required.

2.11 Predefined cement paste behaviour

For this behaviour, the pair `.young_modulus`/`.poisson_ratio` CANNOT be replaced by `.bulk_modulus`/`.shear_modulus`.

Required parameters

Object	Type	Description
`.type`	BehaviourType	`PASTE_BEHAVIOUR`

Optional parameters

Object	Type	Description
`.young_modulus`	Numeral	Young's modulus of the material (in pascal).
`.poisson_ratio`	Numeral	Poisson ratio of the material (arbitrary unit).
`.tensile_strain_limit`	Numeral	Maximum tensile strain the material can be subjected to before failure (in meter/meter). This will not be used if the `.damage` variable is set to `FALSE`
`.short_term_creep_modulus`	Numeral	Elastic modulus of short-term Kelvin-Voigt unit. This will only be used in space-time finite element analysis.
`.long_term_creep_modulus`	Numeral	Elastic modulus of long-term Kelvin-Voigt unit. This will only be used in space-time finite element analysis.
`.damage`	Boolean	Indicates whether to activate the damage model for this behaviour.

If the user does not provide the values for these parameters, the program will use the default values.

2.12 Predefined aggregate behaviour

For this behaviour, the pair `.young_modulus`/`.poisson_ratio` CANNOT be replaced by `.bulk_modulus`/`.shear_modulus`.

Required parameters

Object	Type	Description
`.type`	BehaviourType	`AGGREGATE_BEHAVIOUR`

Optional parameters

Object	Type	Description
`.young_modulus`	Numeral	Young's modulus of the material (in pascal).
`.poisson_ratio`	Numeral	Poisson ratio of the material (arbitrary unit).
`.tensile_strain_limit`	Numeral	Maximum tensile strain the material can be subjected to before failure (in meter/meter). This will not be used if the `.damage` variable is set to `FALSE`
`.damage`	Boolean	Indicates whether to activate the damage model for this behaviour.

If the user does not provide the values for these parameters, the program will use the default values.

2.13 Predefined alkali-silica reaction gel behaviour

For this behaviour, the pair `.young_modulus`/`.poisson_ratio` CANNOT be replaced by `.bulk_modulus`/`.shear_modulus`.

Required parameters

Object	Type	Description
`.type`	BehaviourType	`ASR_GEL_BEHAVIOUR`

Optional parameters

Object	Type	Description
`.young_modulus`	Numeral	Young's modulus of the material (in pascal).
`.poisson_ratio`	Numeral	Poisson ratio of the material (arbitrary unit).
`.imposed_deformation`	Numeral	Imposed deformation of the material (in meter/meter).

If the user does not provide the values for these parameters, the program will use the default values.

2.14 Predefined concrete behaviour

For this behaviour, the pair `.young_modulus`/`.poisson_ratio` CANNOT be replaced by `.bulk_modulus`/`.shear_modulus`. This behaviour CANNOT be used in space-time finite element analysis.

Required parameters

Object	Type	Description
`.type`	BehaviourType	`CONCRETE_BEHAVIOUR`

Optional parameters

Object	Type	Description
`.young_modulus`	Numeral	Young's modulus of the material (in pascal).
`.poisson_ratio`	Numeral	Poisson ratio of the material (arbitrary unit).
`.compressive_strength`	Numeral	Compressive strength of the material (in pascal).

If the user does not provide the values for these parameters, the program will use the default values.

2.15 Predefined rebar behaviour

For this behaviour, the pair `.young_modulus`/`.poisson_ratio` CANNOT be replaced by `.bulk_modulus`/`.shear_modulus`. This behaviour CANNOT be used in space-time finite element analysis.

Required parameters

Object	Type	Description
`.type`	BehaviourType	`REBAR_BEHAVIOUR`

Optional parameters

Object	Type	Description
`.young_modulus`	Numeral	Young's modulus of the material (in pascal).
`.poisson_ratio`	Numeral	Poisson ratio of the material (arbitrary unit).
`.tensile_strength`	Numeral	Tensile strength of the material (in pascal).

If the user does not provide the values for these parameters, the program will use the default values.

2.16 Predefined steel behaviour

For this behaviour, the pair `.young_modulus`/`.poisson_ratio` CANNOT be replaced by `.bulk_modulus`/`.shear_modulus`. This behaviour CANNOT be used in space-time finite element analysis.

Required parameters

Object	Type	Description
`.type`	BehaviourType	`STEEL_BEHAVIOUR`

Optional parameters

Object	Type	Description
`.young_modulus`	Numeral	Young's modulus of the material (in pascal).
`.poisson_ratio`	Numeral	Poisson ratio of the material (arbitrary unit).
`.tensile_strength`	Numeral	Tensile strength of the material (in pascal).

If the user does not provide the values for these parameters, the program will use the default values.

2.17 Generalized logarithmic creep behaviour

Required parameters

Object	Type	Description
`.type`	BehaviourType	`LOGARITHMIC_CREEP`
`.parameters`	MaterialParameters	Lists of all material parameters used in the model.

Optional parameters

Object	Type	Description
`.fracture_criterion`	FractureCriterion	Defines the failure surface and post-peak behaviour of the material.
`.damage_model`	DamageModel	Defines the damage algorithm used.
`.material_law`	MaterialLaw	Defines the evolution of the material parameters with time, or other material parameters. Any arbitrary number of material laws may be defined.

3 Boundary Conditions

Note: multiple `.boundary_conditions` can be defined as first-level objects. If several boundary conditions are defined on the same edge, their effects stack.

The boundary conditions can either be defined with constant values, or with values varying with time.

3.1 Common parameters

The following parameters can be used for both of the cases described below.

Required parameters

Object	Type	Description
`.condition`	BoundaryConditionType	Defines which type of boundary condition is to be applied.
`.position`	BoundingBoxPosition	Defines on which position of the sample the boundary condition must be applied.

Optional parameters

Object	Type	Description
`.axis`	Numeral	Indicates the index of the unknown on which the boundary condition acts. This is only used when making a space-time finite element analysis. See the BoundaryConditionType `SET_ALONG_INDEXED_AXIS` for more details.
`.restriction`	BoundingBoxRestriction	Limits the area on which the boundary condition is applied.
`.point`	Point	Limits the boundary condition to the node closest to the specified point.

The `.restriction` and `.point` option are incompatible one with another. The first limits the section on which the condition is applied (for example, only half of one edge is under load), while the second applies the boundary condition to a single node.

3.2 Constant boundary conditions

The following parameters can be used for both of the cases described below.

Optional parameters

Object	Type	Description
`.value`	Numeral	Defines the value of the imposed displacement (in meters) or stress (in pascals). Positive values correspond to tension, negative values to compression.

If `.value` is not defined, it takes the value 0.

3.3 Time-dependent boundary conditions

The following parameters can be used for both of the cases described below.

Required parameters

Object	Type	Description
`.time_evolution`	BCTimeEvolution	Indicates how the value of the boundary condition evolves in time.

4 Boundary Condition Time Evolution

One (and only one) of the three parameters below must be set:

Optional parameters

Object	Type	Description
`.file_name`	String	Path to the file in the values of the boundary condition are stored. They must be set as a two-column file, the first column being the instants, the second the value at those instants. The value will be linearly interpolated between these points.
`.function`	Function	The value will be set following a specified function of time.
`.rate`	Numeral	The value will be set using a constant rate.

5 Bounding Box Restriction

This object is used with the `boundary_condition`+ object to restrict the surface where the condition is applied. The effective surface will be defined as the intersection between `.boundary_condition.position` and the box defined by the `restriction`.

Required parameters

Object	Type	Description
`.top_right`	Point	Top-right corner of the restriction.
`.bottom_left`	Point	Bottom-left corner of the restriction.

6 Damage Model

6.1 Common parameters

The following parameters can be defined for all types of damage models.

Required parameters

Object	Type	Description
`.type`	DamageModelType	Describes which damage model to use.

Optional parameters

Object	Type	Description
`.maximum_damage`	Numeral	Sets a threshold above which the material is considered to be totally damaged.

6.2 Isotropic Linear Damage

This damage model cannot be used for space-time finite element analysis.

Required parameters

Object	Type	Description
`.type`	DamageModelType	`ISOTROPIC_LINEAR_DAMAGE`.

6.3 Isotropic Incremental Linear Damage

Required parameters

Object	Type	Description
`.type`	DamageModelType	`ISOTROPIC_INCREMENTAL_LINEAR_DAMAGE`.
`.damage_increment`	Numeral	The damage increment to apply at each step of the algorithm.

Optional parameters

Object	Type	Description
`.time_tolerance`	Numeral	Tolerance in the detection of the instant at which damage occurs. This is not used in purely spatial finite element applications.

6.4 Plastic Strain

This damage model cannot be used for space-time finite element analysis.

Required parameters

Object	Type	Description
.type	DamageModelType	PLASTIC_STRAIN.

7 Discretization

Required parameters

Object	Type	Description
.sampling_number	Numeral	Number of mesh points on a edge of the sample.
.order	ElementOrder	Order of the finite element discretization.

Optional parameters

Object	Type	Description
.sampling_restriction	SamplingRestriction	Determines if the smallest inclusions are meshed or not.

8 Export

Note: the .export object can be left empty, in which case no deformed mesh files will be extracted from the simulation.

Required parameters

Object	Type	Description
.at_time_step	TimeStepOuput	Defines when at which time steps the deformed mesh files must be extracted.
.file_name	String	Path template for the files in which the deformed mesh will be written. If these files already exist, their content will be overwritten.
.field	ExtendedFieldType	Defines which fields will be exported.

The export will create a set of deformed mesh files which show the fields defined by .field. All files will share the same base name, followed by a number which corresponds to the index of the file in the set. Two sets of files will be generated: one set of text files formatted for AMIE internal viewer, and one set of corresponding SVG files. Furthermore, the export will write one header file containing the list of all files generated.

9 Fracture Criterion

9.1 Common parameters

The following parameters can be defined for all types of fracture criteria.

Required parameters

Object	Type	Description
`.type`	FractureCriterionType	Describes which fracture criterion to use.

Optional parameters

Object	Type	Description
`.material_characteristic_radius`	Numeral	Sets the characteristic radius of the non-local damage band.

9.2 Maximum tensile strain criterion

Required parameters

Object	Type	Description
`.type`	FractureCriterionType	`MAXIMUM_TENSILE_STRAIN`.
`.limit_tensile_strain`	Numeral	The strain at which failure occurs (in meter/meter).

9.3 Maximum tensile stress criterion

Required parameters

Object	Type	Description
`.type`	FractureCriterionType	`MAXIMUM_TENSILE_STRESS`.
`.limit_tensile_stress`	Numeral	The stress at which failure occurs (in pascal).

9.4 Maximum tensile strain criterion with linear softening

Required parameters

Object	Type	Description
`.type`	FractureCriterionType	`LINEAR_SOFTENING_MAXIMUM_TENSILE_STRAIN`.
`.limit_tensile_strain`	Numeral	The strain at which failure starts (in meter/meter).
`.limit_tensile_stress`	Numeral	The corresponding stress (in pascal).
`.maximum_tensile_strain`	Numeral	The strain at which failure ends (in meter/meter).

9.5 Maximum tensile stress criterion with ellipsoidal softening

Required parameters

Object	Type	Description
`.type`	FractureCriterionType	`ELLIPSOIDAL_SOFTENING_MAXIMUM_TENSILE_STRESS`.
`.limit_tensile_strain`	Numeral	The strain at which failure starts in instantaneous loading conditions (in meter/meter).
`.limit_tensile_stress`	Numeral	The stress at which failure starts in infinitely slow loading conditions (in pascal).
`.instantaneous_modulus`	Numeral	The value of apparent elastic modulus of the material in instantaneous loading conditions (in pascal).
`.relaxed_modulus`	Numeral	The value of apparent elastic modulus of the material in infinitely slow loading conditions (in pascal).

9.6 Mohr-Coulomb criterion

Required parameters

Object	Type	Description
`.type`	FractureCriterionType	`MOHR_COULOMB`.
`.limit_tensile_strain`	Numeral	The strain at which failure occurs in tension (in meter/meter).
`.limit_compressive_strain`	Numeral	The strain at which failure occurs in compression (in meter/meter).

9.7 Linear softening Mohr-Coulomb criterion

Required parameters

Object	Type	Description
`.type`	FractureCriterionType	`LINEAR_SOFTENING_MOHR_COULOMB`.
`.limit_tensile_strain`	Numeral	The strain at which failure occurs in tension (in meter/meter).
`.limit_compressive_strain`	Numeral	The strain at which failure occurs in compression (in meter/meter).
`.maximum_tensile_strain`	Numeral	The strain at which failure ends in tension (in meter/meter).
`.maximum_compressive_strain`	Numeral	The strain at which failure ends in compression (in meter/meter).

9.8 Exponential softening Mohr-Coulomb criterion

Required parameters

Object	Type	Description
`.type`	FractureCriterionType	`EXPONENTIAL_SOFTENING_MOHR_COULOMB`.
`.limit_tensile_strain`	Numeral	The strain at which failure occurs in tension (in meter/meter).
`.limit_compressive_strain`	Numeral	The strain at which failure occurs in compression (in meter/meter).
`.maximum_tensile_strain`	Numeral	The strain at which failure ends in tension (in meter/meter).
`.maximum_compressive_strain`	Numeral	The strain at which failure ends in compression (in meter/meter).

9.9 Von Mises stress criterion

Required parameters

Object	Type	Description
`.type`	FractureCriterionType	`VON_MISES`.
`.limit_tensile_stress`	Numeral	The stress at which failure occurs (in pascal).
`.material_characteristic_radius`	Numeral	Sets the characteristic radius of the non-local damage band.

9.10 Modified Compressive Field Theory criterion

Required parameters

Object	Type	Description
`.type`	FractureCriterionType	`MCFT`.
`.limit_compressive_strain`	Numeral	The strain at which failure occurs in compression (in meter/meter).
`.material_characteristic_radius`	Numeral	Sets the characteristic radius of the non-local damage band.
`.rebar`	Rebar	Describes the location and diameter of the rebars. Several `.rebar` objects may be defined in the same criterion.

9.11 Space-time asymmetric multi-linear softening fracture criterion

For this criterion, the parent behaviour must have the properties `parameters.young_modulus` explicitly defined.

Required parameters

Object	Type	Description
`.type`	FractureCriterionType	`MULTI_LINEAR_SOFTENING_TENSILE_COMPRESSIVE_STRESS`.

Optional parameters

Object	Type	Description
`.strain_renormalization_factor`	Numeral	Arbitrary scaling coefficient to avoid geometrical singularities (default value 10^4).
`.stress_renormalization_factor`	Numeral	Arbitrary scaling coefficient to avoid geometrical singularities (default value 10^{-6}).
`.tension_file_name`	String	Path to the file containing the strain-stress values for the tensile part of the behaviour. If no file is defined, then the material does not fail in tension.
`.compression_file_name`	String	Path to the file containing the strain-stress values for the compressive part of the behaviour (all values should be negative). If no file is defined, then the material does not fail in compression.

10 Geometry

Required parameters

Object	Type	Description
`.type`	GeometryType	Defines which type of inclusion to generate.

Optional parameters for standard particle size distributions

Object	Type	Description
`.aspect_ratio`	Numeral	Defines the elongation of the inclusion. This parameter will not be used for circles or spheres.
`.orientation`	Numeral	Defines the random spread of the orientation of the inclusions. This parameter will not be used for circles or spheres.

Parameters for manually-placed circles

Object	Type	Description
`.type`	GeometryType	`CIRCLE`
`.radius`	Numeral	The radius of the inclusion.
`.center`	Point	The center of the inclusion.

Parameters for manually-placed ellipses

Object	Type	Description
`.type`	GeometryType	`ELLIPSE`
`.major_radius`	Numeral	The major radius of the inclusion.
`.minor_radius`	Numeral	The minor radius of the inclusion (must be smaller than the major radius).
`.center`	Point	The center of the inclusion.
`.major_axis`	Point	The direction of the major axis of the inclusion (must be a vector of norm 1).

Parameters for manually-placed rectangle

Object	Type	Description
`.type`	GeometryType	`RECTANGLE`
`.height`	Numeral	The height of the rectangle.
`.width`	Numeral	The width of the rectangle.
`.center`	Point	The center of the inclusion.

11 Inclusions

Note: multiple `.inclusions` may be defined in the same parent object, in which case each family of inclusion will be generated sequentially one after the other. `.inclusions` defined at the same level may not intersect nor overlap.

Required parameters

Object	Type	Description
`.behaviour`	Behaviour	Defines the mechanical behaviour of the inclusions in the family.
`.particle_size_distribution`	ParticleSizeDistribution	Defines the particle size distribution of the inclusions in the family.

Optional parameters

Object	Type	Description
`.geometry`	Geometry	Defines the mechanical behaviour of the inclusions in the family.
`.placement`	Placement	Defines how the particles are placed in the sample.
`.inclusions`	Inclusions	Generates families of inclusions within the current family of inclusion.
`.sampling_factor`	Numeral	Multiplies the number of mesh points in all inclusions in the family.
`.intersection_sampling_factor`	Numeral	Multiplies the number of mesh points in all inclusions in the family which intersects with the edges of the sample.

There are five different ways the inclusions might be generated. The choice of generation is controlled by the following flag:

```
.inclusions
..particle_size_distribution
...type
```

The five different methods are:

- using an analytic particle size distribution curve,
- using a particle size distribution defined in a text file,
- using a pre-generated inclusion distribution defined in a text file (including a pre-defined placement of said inclusions)
- manually placing of a single inclusion,
- creating the inclusions at the center of a parent list of inclusions.

These are explained in more details in the ParticleSizeDistribution object.

12 Inclusion Output

These objects are used in the `output` to extract the average fields over certain inclusions.

Required parameters

Object	Type	Description
`.index`	Numeral	Indicates the family of inclusions on which the fields will be computed. 0 denotes the material matrix. Other number indicates the `inclusions` families define above, in the order in which they are generated.
`.field`	FieldType	Indicates the field to export in the output. Several of these can be defined.

13 Material Parameters

This object is related to the generalized logarithmic creep behaviour. In addition to the common material properties, it allows the user to define any additional properties or local variables like temperature or relative humidity, which may then be used with the material laws.

13.1 Elastic behaviour

The following properties MUST be defined for any generalized logarithmic creep behaviour. The `.young_modulus/.poisson_r` pair can be replaced with the `.bulk_modulus/shear_modulus` pair (both values in pascal).

Object	Type	Description
`.young_modulus`	Numeral	The elastic stiffness of the material (in pascal).
`.poisson_ratio`	Numeral	The poisson ratio of the material (no unit).

13.2 Creep behaviour

The following properties describe the viscoelastic creep of the behaviour. If they are not defined, then the material behaves like an elastic material. The `.creep_modulus`/`.creep_poisson` pair can be replaced with the `.creep_bulk`/`creep_shear` pair (both values in pascal); same with `.recoverable_modulus`/`recoverable_poisson`.

Object	Type	Description
`.creep_modulus`	Numeral	The initial creep viscosity of the material (in pascal).
`.creep_poisson`	Numeral	The initial creep poisson ratio of the material (no unit).
`.recoverable_modulus`	Numeral	The recoverable creep modulus of the material (in pascal).
`.recoverable_poisson`	Numeral	The recoverable creep poisson ratio of the material (no unit).
`.creep_characteristic_time`	Numeral	The characteristic time of the logarithmic creep law (in days).

If `recoverable_poisson` or `recoverable_modulus` are not defined, then it will be set equal to `.creep_poisson` or `creep_modulus` respectively.

13.3 Imposed deformation

The following properties describe the imposed deformation of the behaviour. If they are not defined, then the material has no imposed deformation and behaves as a elastic (or visco-elastic) material.

Object	Type	Description
`.imposed_deformation`	Numeral	The value of the imposed deformation.

13.4 Damage

The following properties describe the material failure behaviour. If they are defined, then there is no need to specify the `.fracture_criterion` or `.damage_model`. There are two options possible: importing a file with the stress-strain curves, or defining the start and end point of the curve manually. The two methods are mutually exclusive (priority is given to the files if they are defined).

Common parameters

Object	Type	Description
`.material_characteristic_radius`	Numeral	Radius of the area for the non-local averaging (in meters).
`.strain_normalization_factor`	Numeral	Normalization of the strain to improve the accuracy of the computation of intersections in the stress-strain space (optional, default value is 10^4).
`.stress_normalization_factor`	Numeral	Normalization of the stress to improve the accuracy of the computation of intersections in the stress-strain space (optional, default value is 10^{-6}).
`.damage_increment`	Numeral	Amount of damage to apply at each step (no unit).
`.maximum_damage`	Numeral	Amount of damage above which the element is considered entirely broken (no unit).
`.time_tolerance`	Numeral	Minimum time between two damage events (no unit).

Stress-strain curve imported from a file

Object	Type	Description
`.tension_file_name`	String	Path to the stress-strain curve in tension (optional).
`.compression_file_name`	String	Path to the stress-strain curve in compression (optional).

The files must contain a table with the strain (first column) and the stress (second column) of the stress-strain curve. Stresses and strains must be in standard units. Values must be positive for tension and negative for compression. If one of the files is not defined, then the material does not fail in this direction.

Simplified model

Object	Type	Description
`.tensile_strength`	Numeral	Value of the stress at the peak (in Pascal).
`.tensile_ultimate_strength`	Numeral	Value of the stress at the end of the damage process (in Pascal).
`.tensile_ultimate_strain`	Numeral	Value of the strain at the end of the damage process (in meter/meter).
`.compressive_strength`	Numeral	Value of the stress at the peak (in Pascal).
`.compressive_ultimate_strength`	Numeral	Value of the stress at the end of the damage process (in Pascal).
`.compressive_ultimate_strain`	Numeral	Value of the strain at the end of the damage process (in meter/meter).

The values must be positive in tension, negative in compression.

`tensile_strength` can be replaced with `tensile_strain` (the value of the strain at the peak). `tensile_ultimate_strength` can be replaced with `tensile_strength_decrease_factor` (value between 0 and 1) which is the ratio between `tensile_strength` and `tensile_ultimate_strength`. `tensile_ultimate_strain` can be replaced with `tensile_strain_increase_factor` (value greater than 1) which is the ratio between `tensile_ultimate_strain` and `tensile_strain`.

Similar replacements can be made for the compressive strains or stresses.

If `tensile_strength` and `tensile_strain` are both not defined, then the material does not fail in tension (same with compression).

13.5 Additional parameters

Any additional parameter can be defined. These parameters will not affect the material behaviour unless they are used in a subsequent material law.

Object	Type	Description
`.$$$`	Numeral	The value of the parameter called `$$$`.

For example, the following object defines a temperature and relative humidity field with an initial value of 293 K and 95%:

```
.parameters
..temperature = 293
..relative_humidity = 0.95
```

If these values are not constant through the simulation (for example, if there is a gradient in temperature or if the temperature changes in time), then an appropriate Material Law must be defined to describe these evolutions.

Note that these additional parameters do NOT affect the material behaviour (modulus, creep or imposed deformation) unless an according Material Law has been defined. For example, only defining the temperature field is not enough to simulate thermal expansion; a Thermal Expansion Material Law must be added to the model.

14 Material Law

These objects relate the material properties of a generalized logarithmic creep behaviour one to another, and allow changes to the elastic, viscoelastic or imposed deformation properties dynamically in the simulation according to various effects.

14.1 Common parameters

The following parameter must be defined for all types of material laws.

Required parameters

Object	Type	Description
`.type`	MaterialLawType	Describes the type of material laws.

14.2 Thermal expansion

For this material law to be valid, the material parameters must include `temperature` (in Kelvin) and `thermal_expansion_coefficient`.

Required parameters

Object	Type	Description
`.type`	MaterialLawType	`THERMAL_EXPANSION`.
`.reference_temperature`	Numeral	Temperature at which the mechanical properties of the material were measured (in Kelvin).

14.3 Drying shrinkage

For this material law to be valid, the material parameters must include `relative_humidity` (in Kelvin) and `drying_shrinkage_coefficient`.

Required parameters

Object	Type	Description
`.type`	MaterialLawType	`DRYING_SHRINKAGE`.
`.reference_relative_humidity`	Numeral	Relative humidity above which there is no drying shrinkage.

14.4 Arrhenius law

For this material law to be valid, the material parameters must include `temperature` (in Kelvin), the selected parameter `$$$`, and the activation energy `$$$_activation_energy` (in 1/Kelvin).

Required parameters

Object	Type	Description
`.type`	MaterialLawType	`ARRHENIUS`.
`.parameter_affected`	String	Name of the parameters affected by the Arrhenius law.
`.reference_temperature`	Numeral	Temperature at which the nominal properties of the material were measured (in Kelvin).

14.5 Arrhenius law for the creep parameters

For this material law to be valid, the material parameters must include `temperature` (in Kelvin), the three creep parameters `creep_modulus`, `creep_poisson` and `creep_characteristic_time`, and the creep activation energy `creep_activation_energy` (in 1/Kelvin).

Required parameters

Object	Type	Description
`.type`	MaterialLawType	`CREEP_ARRHENIUS`.
`.reference_temperature`	Numeral	Temperature at which the creep properties of the material were measured (in Kelvin).

14.6 Relative humidity effect for the creep parameters

For this material law to be valid, the material parameters must include `relative_humidity`, the three creep parameters `creep_modulus`, `creep_poisson` and `creep_characteristic_time`, and the creep relative humidity coefficient `creep_humidity_coefficient`).

Required parameters

Object	Type	Description
`.type`	MaterialLawType	`CREEP_HUMIDITY`.

14.7 Material law function of the space and time coordinates

This material laws sets one of the predefined internal variable as the result of a function of the space and time coordinates.

Required parameters

Object	Type	Description
`.type`	MaterialLawType	`SPACE_TIME_DEPENDENT`.
`.output_parameter`	String	The name of the parameter in which the results will be stored.
`.function`	Function	The function to apply.
`.additive`	Boolean	If `TRUE`, then the result of the function will be added to the pre-existing value of the output parameter.

14.8 Material law function of a single parameter

This material laws sets one of the predefined internal variable as the result of a function of another existing internal variable.

Required parameters

Object	Type	Description
`.type`	MaterialLawType	`SIMPLE_DEPENDENT`.
`.input_parameter`	String	The name of the parameter used as the `x` argument of the function.
`.output_parameter`	String	The name of the parameter in which the results will be stored.
`.function`	Function	The function to apply. This function must only be defined as a function of the `x` coordinate. Any other variable will be ignored.
`.additive`	Boolean	If `TRUE`, then the result of the function will be added to the pre-existing value of the output parameter.

14.9 Material law function of a set of parameters

This material laws sets one of the predefined internal variable as the result of a function of several existing internal variable or any of the space-time coordinate.

Required parameters

Object	Type	Description
`.type`	MaterialLawType	`VARIABLE_DEPENDENT`.
`.output_parameter`	String	The name of the parameter in which the results will be stored.
`.function`	Function	The function to apply.
`.additive`	Boolean	If `TRUE`, then the result of the function will be added to the pre-existing value of the output parameter.

Optional parameters

Object	Type	Description
`.x`	String	Replaces the `x` argument in the function with the variable defined here.
`.y`	String	Replaces the `y` argument in the function with the variable defined here.
`.z`	String	Replaces the `z` argument in the function with the variable defined here.
`.t`	String	Replaces the `t` argument in the function with the variable defined here.
`.u`	String	Replaces the `u` argument in the function with the variable defined here.
`.v`	String	Replaces the `v` argument in the function with the variable defined here.
`.w`	String	Replaces the `w` argument in the function with the variable defined here.

14.10 Linearly interpolated material law

This material laws sets one of the predefined internal variable as the result of the linear interpolation of another variable.

Required parameters

Object	Type	Description
`.type`	MaterialLawType	`LINEAR_INTERPOLATED`.
`.input_parameter`	String	The name of the parameter used as the argument of the function.
`.output_parameter`	String	The name of the parameter in which the results will be stored.
`.file_name`	String	Path to the file in which the points for the linear interpolation are stored. The file must contain two columns, the first one being the values of the input parameter, the second the corresponding values of the output parameter.

14.11 Assignment material law

This material laws assigns the value of the input variable to the output variable.

Required parameters

Object	Type	Description
`.type`	MaterialLawType	`ASSIGN`.
`.input_parameter`	String	The name of the input parameter.
`.output_parameter`	String	The name of the output parameter.

14.12 Minimum material law

This material laws extracts the minimum value of a set of input parameters.

Required parameters

Object	Type	Description
`.type`	MaterialLawType	`MINIMUM`.
`.input_parameter`	String	The name of the parameter used as the argument of the minimum function. Several of these parameters may be defined.
`.output_parameter`	String	The name of the parameter in which the results will be stored.

14.13 Maximum material law

This material laws extracts the maximum value of a set of input parameters.

Required parameters

Object	Type	Description
`.type`	MaterialLawType	`MAXIMUM`.
`.input_parameter`	String	The name of the parameter used as the argument of the maximum function. Several of these parameters may be defined.
`.output_parameter`	String	The name of the parameter in which the results will be stored.

14.14 Field extractor material law

This material laws extracts the value of one standard field (strain, stress, etc) and assigns it to a specific internal variable.

Required parameters

Object	Type	Description
`.type`	MaterialLawType	`GET_FIELD`.
`.field`	FieldType	Defines which field to extract.

14.15 Time derivative material law

This material laws sets one of the predefined internal variable as the time derivation of another variable.

Required parameters

Object	Type	Description
`.type`	MaterialLawType	`TIME_DERIVATIVE`.
`.input_parameter`	String	The name of the parameter to derive.

Optional parameters

Object	Type	Description
`.output_parameter`	String	The name of the parameter in which the results of the derivation will be stored. If this is not defined, then the results will be stored in `$$$_rate`, where `$$$` is the name of the input parameter.

14.16 Time integral material law

This material laws sets one of the predefined internal variable as the integration over time of another variable.

Required parameters

Object	Type	Description
`.type`	MaterialLawType	`TIME_INTEGRAL`.
`.input_parameter`	String	The name of the parameter to derive.

Optional parameters

Object	Type	Description
`.output_parameter`	String	The name of the parameter in which the results of the derivation will be stored. If this is not defined, then the results will be stored in `$$$_integral`, where `$$$` is the name of the input parameter.

15 Output

Note: the `.output` object can be left empty, in which case no results will be extracted from the simulation.

Required parameters

Object	Type	Description
`.at_time_step`	TimeStepOuput	Defines at which time steps the results must be extracted.
`.file_name`	String	Path to the file in which the results will be written. If the file already exists, its content will be overwritten.
`.field`	FieldType	Defines which fields will be exported.

Optional parameters

Object	Type	Description
`.inclusions`	InclusionOuput	Defines families of inclusion from which fields can be extracted.

The output file is a simple text file containing a table describing the results. Each line corresponds to a time step of the simulation. Each column corresponds to the average value in the sample of the field specified in the `.field` objects (except for the first column, which corresponds to the current time of the simulation). The columns are ordered from the left to the right in the same order the `.fields` are declared (most fields span several columns).

16 Parallelism

This object manages the number of threads used for the computations. It is not required. If not defined, then AMIE will use the current settings with which AMIE was compiled.

Optional parameters

Object	Type	Description
`.number_of_threads`	Numeral	Number of threads used in the simulation.

17 Particle Size Distribution

There are four different methods to generate particles:

- using an analytic particle size distribution curve,
- using a particle size distribution defined in a text file,
- using a pre-generated inclusion distribution defined in a text file (including a pre-defined placement of said inclusions)
- creating the inclusions at the center of a parent list of inclusions.

17.1 Common parameters

The following parameters can be used for all type of particle size distributions.

Required parameters

Object	Type	Description
`.type`	PSDType	Indicates which generation method to use.

17.2 Analytic particle size distribution

Required parameters

Object	Type	Description
`.type`	PSDType	`CONSTANT`, `BOLOME_A`, `BOLOME_B`, `BOLOME_C`, or `BOLOME_D`.
`.rmax`	Numeral	Radius of the largest inclusion.
`.number`	Numeral	Number of inclusions to generate.
`.fraction`	Numeral	Surface or volume fraction of the placement box to cover (no unit).

17.3 Particle size distribution from file

Required parameters

Object	Type	Description
`.type`	PSDType	`FROM_CUMULATIVE_FILE`.
`.rmax`	Numeral	Radius of the largest inclusion.
`.number`	Numeral	Number of inclusions to generate.
`.fraction`	Numeral	Surface or volume fraction of the placement box to cover (no unit).
`.file_name`	String	Path to the file in which the particle size distribution is located.
`.psd_specification_type`	PSDSpecificationType	Defines the format in which the file is written.

Optional parameters

Object	Type	Description
`.factor`	Numeral	Multiplies all radii in the distribution by the specified number.
`.cutoff`	PSDCutoff	Removes the largest or smallest inclusions in the distribution.

17.4 Pregenerated inclusions

Required parameters

Object	Type	Description
`.type`	PSDType	`FROM_INCLUSION_FILE`.
`.number`	Numeral	Number of inclusions to import.
`.file_name`	String	Path to the file in which the inclusions geometric information is located.
`.column`	ColumnIdentifier	Indicates the data contained in each column of the file to import. In general, several `.column` objects must be defined.

17.5 Manually placed inclusion

Required parameters

Object	Type	Description
`.type`	PSDType	`UNIQUE`.

The placement of the inclusion in then covered in the `.inclusions.geometry` object

17.6 Concentric inclusions

This will generate the inclusions at the center the p

Required parameters

Object	Type	Description
`.type`	PSDType	`FROM_PARENT_DISTRIBUTION`.

Optional parameters

Object	Type	Description
`.layer_thickness`	Numeral	Indicates the radius difference between the parent set of inclusions and the current.
`.layer_thickness_function`	Function	Indicates the radius difference between the parent set of inclusions and the current as a function of the radius of the parent distribution.

Note: at least one parameter between `.layer_thickness` and `.layer_thickness_function` must be declared.

18 Particle Size Distribution Cutoff

Describes whether to include or remove the largest and/or smallest aggregates in a particle size distribution.

Optional parameters

Object	Type	Description
`.up`	Numeral	Describes the largest aggregate radius authorized in the distribution. If it is not defined, then there is no upper limit.
`.down`	Numeral	Describes the smallest aggregate radius authorized in the distribution. If it is not defined, then there is no lower limit.

19 Placement

Required parameters

Object	Type	Description
`.tries`	Numeral	Number of tries to make during the random placement of particles.
`.spacing`	Numeral	Defines the minimum distance between two particles or one particle and the edges of the placing box.

Optional parameters

Object	Type	Description
`.box`	Sample	Defines a rectangle in which the particles will be generated. If this variable is not set, the first-level `.sampl` will be used instead.
`.orientation`	Numeral	Defines the random spread of the orientation of the inclusions. This parameter will not be used for circles or spheres.

20 Point

Optional parameters

Object	Type	Description
`.x`	Numeral	Value of the `x` coordinate.
`.y`	Numeral	Value of the `y` coordinate.
`.z`	Numeral	Value of the `z` coordinate.
`.t`	Numeral	Value of the `t` coordinate.

The default values for any unspecified coordinate is equal to 0.

21 Rebar

Required parameters

Object	Type	Description
`.location`	Numeral	Position of the rebar (in meter).
`.diameter`	Numeral	Diameter of the rebar (in meter).

22 Sample

Required parameters

Object	Type	Description
`.width`	Numeral	Width of the box (in meters).
`.height`	Numeral	Height of the box (in meters).
`.behaviour`	Behaviour	Mechanical behaviour of the matrix phase.

Optional parameters

Object	Type	Description
`.center`	Point	Coordinates of the center.

23 Stepping

The stepping procedure can be initialized either with a constant predefined time steps, or from a file which lists the instants at which the simulation is carried out.

23.1 Common parameters

The following parameters can be used for both of the cases described below.

Optional parameters

Object	Type	Description
`.minimum_time_step`	Numeral	Minimum duration between two damage events.
`.maximum_iterations_per_step`	Numeral	Maximum number of iterations of the damage algorithm between two time steps.

23.2 Constant time step

Required parameters

Object	Type	Description
`.time_step`	Numeral	Duration of a time step (in days).
`.number_of_time_steps`	Numeral	Number of time steps to perform.

23.3 Logarithmic time step

Required parameters

Object	Type	Description
.logarithmic	Boolean	Must be TRUE to use a time step constant in the logarithmic space.
.first_time_step	Numeral	Duration of the first time step (in the normal time space).
.time_step	Numeral	Duration of a time step in the logarithmic space.
.number_of_time_steps	Numeral	Number of time steps to perform.

23.4 Function-defined time step

Required parameters

Object	Type	Description
.time_step	Numeral	Duration of the first time step (in days).
.next_time_step	Function	Function used to compute the next time step from the previous, with "x" being the previous time step and "t" the actual time at the end of the previous time step.

23.5 File-defined time steps

Required parameters

Object	Type	Description
.list_of_time_steps	String	Path to the file which lists the instants at which calculations are performed. The file must contain only one column of increasing numbers starting with 0.

24 Time step output

Required parameters

Object	Type	Description
.at	TimeStepSelection	Defines at which time steps output or export must be done.

Optional parameters

Object	Type	Description
.every	Numeral	In case of a REGULAR type of output, this indicates at which time steps the output must be done.

25 Viscoelastic Unit

This object stores the mechanical properties of a spring-dashpot pair. However, their assembly (in parallel or in series) is managed by the parent viscoelastic behaviour object.

Note: the .young_modulus/.poisson_ratio pair can be replaced with the .bulk_modulus/shear_modulus pair (both values in pascal).

Required parameters

Object	Type	Description
`.young_modulus`	Numeral	The elastic stiffness of the spring (in pascal).
`.poisson_ratio`	Numeral	The poisson ratio of the spring (no unit).
`.characteristic_time`	Numeral	The characteristic time of the dashpot (in days).

26 Enumerations

This section details the list of accepted values for the different enumerations found in the configuration.

26.1 Behaviour Type

`VOID_BEHAVIOUR` (default value),
`ELASTICITY`,
`ELASTICITY_AND_FRACTURE`,
`ELASTICITY_AND_IMPOSED_DEFORMATION`,
`LOGARITHMIC_CREEP`,
`PASTE_BEHAVIOUR`,
`AGGREGATE_BEHAVIOUR`,
`ASR_GEL_BEHAVIOUR`,
`CONCRETE_BEHAVIOUR`,
`REBAR_BEHAVIOUR`,
`STEEL_BEHAVIOUR`,
`VISCOSITY`,
`KELVIN_VOIGT`,
`MAXWELL`,
`BURGER`,
`GENERALIZED_KELVIN_VOIGT`,
`GENERALIZED_MAXWELL`,

26.2 Boolean

`TRUE`,
`FALSE`

26.3 Boundary Condition Type

`GENERAL`,
`FIX_ALONG_ALL`,
`FIX_ALONG_XI`,
`SET_ALONG_XI`,
`FIX_ALONG_ETA`,
`SET_ALONG_ETA`,
`FIX_ALONG_ZETA`,
`SET_ALONG_ZETA`,
`FIX_ALONG_XI_ETA`,
`SET_ALONG_XI_ETA`,
`FIX_ALONG_XI_ZETA`,
`SET_ALONG_XI_ZETA`,
`FIX_ALONG_ETA_ZETA`,

```
SET_ALONG_ETA_ZETA,
FIX_ALONG_INDEXED_AXIS,
SET_ALONG_INDEXED_AXIS,
SET_FORCE_XI,
SET_FORCE_ETA,
SET_FORCE_ZETA,
SET_FORCE_INDEXED_AXIS,
SET_FLUX_XI,
SET_FLUX_ETA,
SET_FLUX_ZETA,
SET_VOLUMIC_STRESS_XI,
SET_VOLUMIC_STRESS_ETA,
SET_VOLUMIC_STRESS_ZETA,
SET_STRESS_XI,
SET_STRESS_ETA,
SET_STRESS_ZETA,
SET_NORMAL_STRESS,
SET_TANGENT_STRESS,
VERTICAL_PLANE_SECTIONS,
HORIZONTAL_PLANE_SECTIONS,
nullptr_CONDITION,
SET_GLOBAL_FORCE_VECTOR
```

26.4 Bounding Box Position

```
TOP,
LEFT,
BOTTOM,
RIGHT,
FRONT,
BACK,
BEFORE,
NOW,
AFTER,
TOP_LEFT,
TOP_RIGHT,
BOTTOM_LEFT,
BOTTOM_RIGHT,
FRONT_LEFT,
FRONT_RIGHT,
BACK_LEFT,
BACK_RIGHT,
FRONT_TOP,
FRONT_BOTTOM,
BOTTOM_BACK,
TOP_BACK,
TOP_LEFT_FRONT,
TOP_LEFT_BACK,
BOTTOM_LEFT_FRONT,
BOTTOM_LEFT_BACK,
TOP_RIGHT_FRONT,
TOP_RIGHT_BACK,
```

```
BOTTOM_RIGHT_FRONT,
BOTTOM_RIGHT_BACK,
TOP_BEFORE,
LEFT_BEFORE,
BOTTOM_BEFORE,
RIGHT_BEFORE,
FRONT_BEFORE,
BACK_BEFORE,
TOP_LEFT_BEFORE,
TOP_RIGHT_BEFORE,
BOTTOM_LEFT_BEFORE,
BOTTOM_RIGHT_BEFORE,
FRONT_LEFT_BEFORE,
FRONT_RIGHT_BEFORE,
BACK_LEFT_BEFORE,
BACK_RIGHT_BEFORE,
FRONT_TOP_BEFORE,
FRONT_BOTTOM_BEFORE,
TOP_LEFT_FRONT_BEFORE,
TOP_LEFT_BACK_BEFORE,
BOTTOM_LEFT_FRONT_BEFORE,
BOTTOM_LEFT_BACK_BEFORE,
TOP_RIGHT_FRONT_BEFORE,
TOP_RIGHT_BACK_BEFORE,
BOTTOM_RIGHT_FRONT_BEFORE,
BOTTOM_RIGHT_BACK_BEFORE,
BOTTOM_BACK_BEFORE,
TOP_BACK_BEFORE,
TOP_NOW,
LEFT_NOW,
BOTTOM_NOW,
RIGHT_NOW,
FRONT_NOW,
BACK_NOW,
TOP_LEFT_NOW,
TOP_RIGHT_NOW,
BOTTOM_LEFT_NOW,
BOTTOM_RIGHT_NOW,
FRONT_LEFT_NOW,
FRONT_RIGHT_NOW,
BACK_LEFT_NOW,
BACK_RIGHT_NOW,
FRONT_TOP_NOW,
FRONT_BOTTOM_NOW,
TOP_LEFT_FRONT_NOW,
TOP_LEFT_BACK_NOW,
BOTTOM_LEFT_FRONT_NOW,
BOTTOM_LEFT_BACK_NOW,
TOP_RIGHT_FRONT_NOW,
TOP_RIGHT_BACK_NOW,
BOTTOM_RIGHT_FRONT_NOW,
```

BOTTOM_RIGHT_BACK_NOW,
TOP_AFTER,
LEFT_AFTER,
BOTTOM_AFTER,
RIGHT_AFTER,
FRONT_AFTER,
BACK_AFTER,
TOP_LEFT_AFTER,
TOP_RIGHT_AFTER,
BOTTOM_LEFT_AFTER,
BOTTOM_RIGHT_AFTER,
FRONT_LEFT_AFTER,
FRONT_RIGHT_AFTER,
BACK_LEFT_AFTER,
BACK_RIGHT_AFTER,
FRONT_TOP_AFTER,
FRONT_BOTTOM_AFTER,
TOP_LEFT_FRONT_AFTER,
TOP_LEFT_BACK_AFTER,
BOTTOM_LEFT_FRONT_AFTER,
BOTTOM_LEFT_BACK_AFTER,
TOP_RIGHT_FRONT_AFTER,
TOP_RIGHT_BACK_AFTER,
BOTTOM_RIGHT_FRONT_AFTER,
BOTTOM_RIGHT_BACK_AFTER,
BOTTOM_BACK_AFTER,
TOP_BACK_AFTER.

The bounding box positions involving `TOP` and `BOTTOM` are only used in three dimensions. The bounding box positions involving `BEFORE`, `NOW` and `AFTER` are only used in space-time finite element analysis.

26.5 Column Identifier

RADIUS,
RADIUS_A,
RADIUS_B,
CENTER_X,
CENTER_Y,
CENTER_Z,

26.6 Damage Model Type

ISOTROPIC_LINEAR_DAMAGE,
ISOTROPIC_INCREMENTAL_LINEAR_DAMAGE,
PLASTIC_STRAIN,

26.7 Element Order

CONSTANT,
LINEAR (default value),
QUADRATIC,
CUBIC,

QUADRIC,
QUINTIC,
CONSTANT_TIME_LINEAR,
CONSTANT_TIME_QUADRATIC,
LINEAR_TIME_LINEAR,
LINEAR_TIME_QUADRATIC,
QUADRATIC_TIME_LINEAR,
QUADRATIC_TIME_QUADRATIC,
CUBIC_TIME_LINEAR,
CUBIC_TIME_QUADRATIC,
QUADRIC_TIME_LINEAR,
QUADRIC_TIME_QUADRATIC,
QUINTIC_TIME_LINEAR,
QUINTIC_TIME_QUADRATIC,
QUADTREE_REFINED,
REGULAR_GRID.

26.8 Extended Field Type

CRITERION,
STIFFNESS,
VISCOSITY.

This object can also take the value of any Field Type defined above. They can also accept any String, as long as they correspond to material parameters defined in the case of generalized logarithmic creep behaviour.

26.9 Field Type

DISPLACEMENT_FIELD,
ENRICHED_DISPLACEMENT_FIELD,
SPEED_FIELD,
FLUX_FIELD,
GRADIENT_FIELD,
STRAIN_FIELD,
STRAIN_RATE_FIELD,
EFFECTIVE_STRESS_FIELD,
REAL_STRESS_FIELD,
PRINCIPAL_STRAIN_FIELD,
PRINCIPAL_EFFECTIVE_STRESS_FIELD,
PRINCIPAL_REAL_STRESS_FIELD,
NON_ENRICHED_STRAIN_FIELD,
NON_ENRICHED_STRAIN_RATE_FIELD,
NON_ENRICHED_EFFECTIVE_STRESS_FIELD,
NON_ENRICHED_REAL_STRESS_FIELD,
VON_MISES_STRAIN_FIELD,
VON_MISES_REAL_STRESS_FIELD,
VON_MISES_EFFECTIVE_STRESS_FIELD,
PRINCIPAL_ANGLE_FIELD,
INTERNAL_VARIABLE_FIELD,
GENERALIZED_VISCOELASTIC_DISPLACEMENT_FIELD,
GENERALIZED_VISCOELASTIC_ENRICHED_DISPLACEMENT_FIELD,
GENERALIZED_VISCOELASTIC_SPEED_FIELD,

GENERALIZED_VISCOELASTIC_STRAIN_FIELD,
GENERALIZED_VISCOELASTIC_STRAIN_RATE_FIELD,
GENERALIZED_VISCOELASTIC_EFFECTIVE_STRESS_FIELD,
GENERALIZED_VISCOELASTIC_REAL_STRESS_FIELD,
GENERALIZED_VISCOELASTIC_PRINCIPAL_STRAIN_FIELD,
GENERALIZED_VISCOELASTIC_PRINCIPAL_EFFECTIVE_STRESS_FIELD,
GENERALIZED_VISCOELASTIC_PRINCIPAL_REAL_STRESS_FIELD,
GENERALIZED_VISCOELASTIC_NON_ENRICHED_STRAIN_FIELD,
GENERALIZED_VISCOELASTIC_NON_ENRICHED_STRAIN_RATE_FIELD,
GENERALIZED_VISCOELASTIC_NON_ENRICHED_EFFECTIVE_STRESS_FIELD,
GENERALIZED_VISCOELASTIC_NON_ENRICHED_REAL_STRESS_FIELD,
SCALAR_DAMAGE_FIELD

26.10 Fracture Criterion Type

MAXIMUM_TENSILE_STRAIN,
MAXIMUM_TENSILE_STRESS,
LINEAR_SOFTENING_MAXIMUM_TENSILE_STRAIN,
ELLIPSOIDAL_SOFTENING_MAXIMUM_TENSILE_STRESS,
MOHR_COULOMB,
LINEAR_SOFTENING_MOHR_COULOMB,
EXPONENTIAL_SOFTENING_MOHR_COULOMB,
MCFT,
VON_MISES,
MULTI_LINEAR_SOFTENING_TENSILE_COMPRESSIVE_STRESS,

26.11 Geometry Type

CIRCLE (default value),
LAYERED_CIRCLE,
TRIANGLE,
RECTANGLE,
PARALLELOGRAMME,
CONVEX_POLYGON,
SEGMENTED_LINE,
ORIENTABLE_CIRCLE,
CLOSED_NURB,
TETRAHEDRON,
HEXAHEDRON,
SPHERE,
LAYERED_SPHERE,
REGULAR_OCTAHEDRON,
ELLIPSE,
LEVEL_SET,
TIME_DEPENDENT_CIRCLE,

26.12 Material Law Type

THERMAL_EXPANSION,
DRYING_SHRINKAGE,
ARRHENIUS,

CREEP_ARRHENIUS,
CREEP_HUMIDITY,
SPACE_TIME_DEPENDENT,
SIMPLE_DEPENDENT,
VARIABLE_DEPENDENT,
LINEAR_INTERPOLATED,
ASSIGN,
MINIMUM,
MAXIMUM,
GET_FIELD,
TIME_DERIVATIVE,
TIME_INTEGRAL,

26.13 Particle Size Distribution Type

CONSTANT (default value),
BOLOME_A,
BOLOME_B,
BOLOME_C,
BOLOME_D,
FROM_CUMULATIVE_FILE,
FROM_INCLUSION_FILE,
FROM_PARENT_DISTRIBUTION,

26.14 Particle Size Distribution Specification Type

CUMULATIVE_PERCENT,
CUMULATIVE_FRACTION,
CUMULATIVE_ABSOLUTE,
CUMULATIVE_PERCENT_REVERSE,
CUMULATIVE_FRACTION_REVERSE,
CUMULATIVE_ABSOLUTE_REVERSE

26.15 Sampling Restriction

SAMPLE_RESTRICT_4,
SAMPLE_RESTRICT_8,
SAMPLE_RESTRICT_16,
SAMPLE_NO_RESTRICTION (default value).

26.16 Time Step Selection

NONE
ALL,
FIRST
LAST
REGULAR

27 History

Version 1.2

- Added embedded fracture properties for the logarithmic creep material.
- Added restriction- and point-defined boundary conditions.
- Added manually pre-placed inclusions.
- Added parallelism object.

Version 1.1

- Added Space-time asymmetric multi-linear softening fracture criterion.
- Added Inclusion Output.
- Added Assignment, Minimum, Maximum and Field extractor material laws.
- Completed Output.
- Added logarithmic and incremental time stepping.

Version 1.0

- Initial version.