
X-stream API Documentation

Release 1.0.0

Luuk Thielen, Miriam Del Hoyo, Ralph Altenfeld

January 19, 2017

xstream package

Submodules

xstream.XStFieldID module

Additional test case and X-Stream specific field IDs

xstream.XStPropertyID module

Additional test case and X-Stream specific property IDs

xstream.xstream module

class `xstream.xstream.xstream` (*file*='', *workdir*='')

Bases: `mupif.Application.Application`

X-Stream application interface class This class implements the methods of the Mupif Application class for X-Stream.

Note This example API uses pre-calculated results. X-Stream itself will not be started.

__init__ (*file*='', *workdir*='')

Interface constructor: Prepares the environment for the steering the external X-Stream binary, i.e. setting of variables for license server and global memory size for the MPI execution. A copy of the binary to the working directory is done (might be unnecessary, if X-Stream would be called in another way). Interface attributes are initialized.

Parameters

- **file** (*string*) – test case configuration file name
- **workdir** (*string*) – name of work directory

fieldEvaluate (*pos*, *fieldID*, *time*)

This method evaluates the Mupif field specified by its field ID at a given position. It combines the method calls of `getField()` and `evaluate()` to stay locally to the interface and avoids a data transfer of the whole field.

Note This method extends the virtual Mupif application interface. It returns only the value of the first position. See also fieldID specific limitation of getField().

Parameters

- **fieldID** (*integer*) – ID of the field to get values from
- **position** (*tuple, a list of tuples*) – 1D/2D/3D position vectors
- **time** (*float*) – the field as it was at time will be evaluated. Ignored: see getField()

Returns field value at first position

Return type field value type

getApplicationSignature()

Get the interface and external application signature.

Returns returns application signature

Return type string

getCriticalTimeStep()

Get the critical time step. This step is given in the X-Stream case file. It is the normal application time step.

Returns float

getField(fieldID)

This method reads native application results at the current simulation time (see solveStep) and converts them to a Mupif field which will be returned.

Note In the current implementation the ID will be ignored. The field file specified in the configuration will be read, i.e. for temperature.

Parameters **fieldID** (*FieldID*) – ID of the field to get (see XStFieldID.py or Mupif FieldID.py)

Returns returns Field instance

Return type Field

setField(field, objectID=0)

Sets a given field as part of the input for the next time step of X-Stream. The field is written to file which is addressed in the X-Stream case file. This is the usual method to set 2D boundary conditions.

Parameters **field** (*Mupif.Field*) – the Mupif field to be set

setProperty(property, objectID=0)

Sets the given property in the X-Stream input file for the next time step. According to the property ID, the specific parts of the input file are replaced directly (no markers).

Parameters **property** (*Mupif.Property*) – the Mupif property to be set

solveStep(tstep, stageID=0, runInBackground=False)

This method steers X-Stream in order to restart the current simulation to the simulation time given in 'tstep'. As an additional feature, the results can be backuped according to the setting in 'xstreamConfig'.

Note In the current implementation the behaviour of this method is not parameter dependent. The solution will proceed a critical time step as defined in the X-Stream case file (a constant step, see getCriticalTimeStep()). The external X-Stream application will be run in the foreground.

Note Instead of running X-Stream in this example, the results will be restored from an backup.

Parameters **tstep** (*Mupif.TimeStep*) – target time; ignored

terminate()

Close the interface and clean up.

xstream.xstreamConfig module

X-Stream interface configuration file

This file provides necessary information about the X-Stream installation which should be steered by the interface.

note In this example, X-Stream will not be run directly. All result data is pre-calculated

Module contents

X-Stream application interface to be used by MuPIF (Multi-Physics Integration Framework)