Global Groundwater Model

Generated by Doxygen 1.8.8

Thu Nov 16 2017 09:25:09

Contents

Chapter 1

Deprecated List

Namespace GlobalFlow

use methods from DataProcessing

Member GlobalFlow::GW_Interface::writeData ()=0

Class GlobalFlow::Logging::InfoSinkInterface

Use methods from DataProcessing

Chapter 2

Bug List

Namespace GlobalFlow

Some conversions make no sense

Chapter 3

Namespace Index

3.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 4

Hierarchical Index

4.1 Class Hierarchy

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

Chapter 5

Class Index

5.1 Class List

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

Chapter 6

Namespace Documentation

6.1 GlobalFlow Namespace Reference

Classes

- class Callback
- class Container
- class DataReader
- class GW_Interface

Typedefs

- using **NodeVector** = std::shared_ptr< std::vector< std::unique_ptr< Model::NodeInterface >>>

Enumerations

- enum Field { **RIVER**, **LAKE**, **WETLANDS** }
- enum **severity_level** {
  debug, **userinfo**, **stateinfo**, **numerics**,
  error, **critical** }

6.1.1 Detailed Description

Converter Functions intended to be used internaly only Currently only needed to allow compilation of different internal types There should be a more beautiful solution to this

Bug Some conversions make no sense

Deprecated use methods from DataProcessing

6.1.2 Enumeration Type Documentation

6.1.2.1 enum GlobalFlow::Field [strong]

Container for transferring data from GW-modell to WaterGAP Where as get(Field) always returns a map from type <ArcID, value>

Rivers:

Lakes:

Wetlands:

Definition at line 31 of file GW_Interface.hpp.

Chapter 7

Class Documentation

7.1 A Class Reference

7.1.1 Detailed Description

to write logs to files

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

- /home/robert/development/G3M-framework/src/Logging/Logging.hpp

7.2 Abstract Class Reference

7.2.1 Detailed Description

a data output

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputManager.hpp

7.3 GlobalFlow::Simulation::AbstractStepper Class Reference

Inheritance diagram for GlobalFlow::Simulation::AbstractStepper:

Public Member Functions

- virtual Solver::Equation $*$ **get** (int col) const =0

7.3.1 Detailed Description

in order to iterate simply over simulation steps Holds a pointer to the equation and the choosen stepsize

Definition at line 34 of file Stepper.hpp.

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

- /home/robert/development/G3M-framework/src/Simulation/Stepper.hpp

7.4 GlobalFlow::Solver::AdaptiveDamping Class Reference

```
#include <Numerics.hpp>
```

Public Member Functions

- **AdaptiveDamping** (double Sigma_Min, double Sigma_Max, double Change_Max, vector x_t0)
- vector **getDamping** (vector &residuals, vector &x, bool apply)
- double **getNorm** ()

7.4.1 Detailed Description

Can be used to dampen the head change per iteration

Definition at line 26 of file Numerics.hpp.

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

- /home/robert/development/G3M-framework/src/Solver/Numerics.hpp

7.5 GlobalFlow::Callback Class Reference

Public Member Functions

- void **call** (Container container)
- void **operator()** (Container container)

7.5.1 Detailed Description

Definition at line 57 of file GW_Interface.hpp.

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

- /home/robert/development/G3M-framework/src/GW_Interface.hpp

7.6 GlobalFlow::Container Class Reference

Public Member Functions

- **Container** (unordered_map< int, double > rivers)
- const unordered_map< int, double > **get** (Field field)

7.6.1 Detailed Description

Definition at line 39 of file GW_Interface.hpp.

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

- /home/robert/development/G3M-framework/src/GW_Interface.hpp

7.7 GlobalFlow::DataProcessing::DataOutput::CSVOutput< T > Class Template Reference

Inheritance diagram for GlobalFlow::DataProcessing::DataOutput::CSVOutput< T >:

Collaboration diagram for GlobalFlow::DataProcessing::DataOutput::CSVOutput< T >:

Public Member Functions

- void **write** (path filePath, bool printID, bool printXY, std::vector< std::pair< double, double >> data, pos_v p)
- void **write** (path filePath, bool printID, bool printXY, std::vector< bool > data, pos_v p)
- void **write** (path filePath, bool printID, bool printXY, std::vector< double > data, pos_v p)
- void **write** (path filePath, bool printID, bool printXY, std::vector< std::string > data, pos_v p)

7.7.1 Detailed Description

template<typename T>class GlobalFlow::DataProcessing::DataOutput::CSVOutput< T >

Definition at line 71 of file OutputFactory.hpp.

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputFactory.hpp

7.8 GlobalFlow::DataReader Class Reference

Public Member Functions

- void **initNodes** (NodeVector nodes)
- virtual void readData (Simulation::Options op)=0
- template<class Fun >
  void loopFiles (std::string path, std::vector< std::string > files, Fun fun)
- int **check** (int globid)
- template<class ProcessDataFunction >
  void readTwoColumns (std::string path, ProcessDataFunction processData)
- void readZeroPointFiveToFiveMin (std::string path)
- const std::unordered_map< int, std::vector< int > > & **getArcIDMapping** ()
- const std::unordered_map< int, int > & **getGlobIDMapping** ()
- std::string **buildDir** (std::string path)

Public Attributes

- fs::path **data_dir** {"data"}

Protected Attributes

- NodeVector **nodes**
- int **stepMod**
- std::unordered_map< int, int > **lookupglobIDtoID**
- std::unordered_map< int, std::vector< int > > **lookupZeroPointFivetoFiveMinute**

7.8.1 Detailed Description

Definition at line 43 of file DataReader.hpp.

7.8.2 Member Function Documentation

7.8.2.1 template<class Fun > void GlobalFlow::DataReader::loopFiles (std::string *path,* std::vector< std::string > *files,* Fun *fun*) [inline]

Gereric method for looping through files inside a directory and apply a gernic function

Parameters

path	
files	
fun	a function

Definition at line 70 of file DataReader.hpp.

7.8.2.2 virtual void GlobalFlow::DataReader::readData (Simulation::Options *op*) [pure virtual]

Attention

Needs to be implemented!

Note

Is called by simulation at startup

Parameters

op	Options object

7.8.2.3 template<class ProcessDataFunction > void GlobalFlow::DataReader::readTwoColumns (std::string *path,* ProcessDataFunction *processData*) [inline]

Read data from a two-column csv file and apply function to data

Parameters

path	
processData	A processing function e.g. upscaling of data

Definition at line 94 of file DataReader.hpp.

7.8.2.4 void GlobalFlow::DataReader::readZeroPointFiveToFiveMin (std::string *path*) [inline]

Creates a mapping of 0.5° ArcIDs to a list of contained 5' GlobIDs

Parameters

path	

Definition at line 116 of file DataReader.hpp.

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataReader.hpp

7.9 GlobalFlow::Solver::Equation Class Reference

Public Types

- typedef Eigen::MatrixXd::Scalar **Scalar**
- typedef Matrix< Scalar, Dynamic, 1 > **VectorType**

Public Member Functions

- **Equation** (unsigned long numberOfNodes, NodeVector nodes, Simulation::Options options)
- void solve ()
- int getItter ()
- double getError ()
- **Equation** (const Equation &)=delete
- Equation & **operator=** (const Equation &)=delete
- VectorXd **getResults** ()
- bool toogleSteadyState ()
- void updateStepSize (size_t mod)
- VectorType **getResiduals** ()
- void **updateClosingCrit** (double crit)

Friends

- std::ostream & operator<< (std::ostream &os, Equation &eq)

7.9.1 Detailed Description

finite difference equation Should only be accessed through the stepper

Definition at line 45 of file Equation.hpp.

7.9.2 Member Function Documentation

7.9.2.1 double GlobalFlow::Solver::Equation::getError ( )

Returns

The current residual error

Definition at line 328 of file Equation.cpp.

7.9.2.2 int GlobalFlow::Solver::Equation::getItter ( )

Returns

The number of iterations

Definition at line 323 of file Equation.cpp.

7.9.2.3 void GlobalFlow::Solver::Equation::solve ( )

Solve the current iteration step

Solve Equation head change convergance

residual norm convergance

Definition at line 188 of file Equation.cpp.

7.9.2.4 bool GlobalFlow::Solver::Equation::toogleSteadyState ( ) [inline]

Toogle the steady-state in all nodes

Returns

Definition at line 93 of file Equation.hpp.

7.9.2.5 void GlobalFlow::Solver::Equation::updateStepSize (size_t *mod*) [inline]

Set the correct stepsize (default is DAY)

Parameters

mod	

Definition at line 108 of file Equation.hpp.

7.9.3 Friends And Related Function Documentation

7.9.3.1 std::ostream& operator<< (std::ostream & *os*, Equation & *eq*) [friend]

Helper to write out current residuals

Parameters

os	
eq	

Returns

Definition at line 79 of file Equation.hpp.

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

- /home/robert/development/G3M-framework/src/Solver/Equation.hpp
- /home/robert/development/G3M-framework/src/Solver/Equation.cpp

7.10 GlobalFlow::Model::ExternalFlow Class Reference

```
#include <ExternalFlows.hpp>
```

Public Member Functions

- **ExternalFlow** (int id, FlowType type, t_meter flowHead, t_s_meter_t cond, t_meter bottom)
- ExternalFlow (int id, t_vol_t recharge, FlowType type)
- ExternalFlow (int id, t_meter flowHead, t_meter bottom, t_vol_t evapotrans)

  Constructor for Evapotranspiration.
- bool flowIsHeadDependant (t_meter head) const noexcept
- t_s_meter_t getP (t_meter head, t_meter eq_head, t_vol_t recharge, t_dim slope, t_vol_t eqFlow) const noexcept
- t_vol_t getQ (t_meter head, t_meter eq_head, t_vol_t recharge, t_dim slope, t_vol_t eqFlow) const noexcept
- FlowType **getType** () const noexcept
- t_meter **getBottom** () const noexcept
- t_vol_t **getRecharge** () const noexcept
- t_meter **getFlowHead** () const noexcept
- t_s_meter_t **getDyn** (t_vol_t current_recharge, t_meter eq_head, t_meter head, t_vol_t eq_flow) const noexcept
- t_meter **getRiverDiff** (t_meter eqHead) const noexcept
- t_s_meter_t **getConductance** () const noexcept
- int **getID** () const noexcept
- void **setMult** (double mult)

7.10.1 Detailed Description

TODO add flow equation here

Definition at line 81 of file ExternalFlows.hpp.

7.10.2 Constructor & Destructor Documentation

7.10.2.1 GlobalFlow::Model::ExternalFlow::ExternalFlow (int *id,* t_vol_t *recharge,* FlowType *type*) [inline]

Only for RECHARGE FAST_SURFACE_RUNOFF

Definition at line 94 of file ExternalFlows.hpp.

7.10.2.2 GlobalFlow::Model::ExternalFlow::ExternalFlow (int *id,* t_meter *flowHead,* t_meter *bottom,* t_vol_t *evapotrans*) [inline]

Constructor for Evapotranspiration.

Parameters

id	
flowHead	
bottom	
evapotrans	

Returns

Definition at line 105 of file ExternalFlows.hpp.

7.10.3 Member Function Documentation

7.10.3.1 bool GlobalFlow::Model::ExternalFlow::flowIsHeadDependant (t_meter *head*) const [inline], [noexcept]

Check if flow can be calculated on the right hand side

Parameters

head	The current hydraulic head

Returns

Bool

Definition at line 113 of file ExternalFlows.hpp.

7.10.3.2 t_s_meter_t GlobalFlow::Model::ExternalFlow::getP (t_meter *head*, t_meter *eq_head*, t_vol_t *recharge*, t_dim *slope*, t_vol_t *eqFlow*) const [noexcept]

The head dependant part of the external flow equation

Parameters

head	The current hydraulic head
eq_head	The equilibrium head
recharge	The current recharge
slope	
eqFlow	

Returns

Definition at line 6 of file ExternalFlows.cpp.

7.10.3.3 t_vol_t GlobalFlow::Model::ExternalFlow::getQ (t_meter *head*, t_meter *eq_head*, t_vol_t *recharge*, t_dim *slope*, t_vol_t *eqFlow*) const [noexcept]

The head independant part of the external flow equation

Parameters

head	
eq_head	
recharge	
slope	
eqFlow	

Returns

Definition at line 52 of file ExternalFlows.cpp.

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

- /home/robert/development/G3M-framework/src/Model/ExternalFlows.hpp
- /home/robert/development/G3M-framework/src/Model/ExternalFlows.cpp

7.11 Factory Class Reference

7.11.1 Detailed Description

correct data output file

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputFactory.hpp

7.12 GlobalFlow::DataProcessing::DataOutput::FieldCollector Class Reference

```
#include <FieldCollector.hpp>
```

Public Types

- using **pos_v** = std::vector< std::pair< double, double >>

Public Member Functions

- **FieldCollector** (FieldType enumField)
- pos_v **getPositions** (Simulation::Simulation const &simulation)
- template<typename T >
  data_vector< T > get (Simulation::Simulation const &simulation)

7.12.1 Detailed Description

Iterates over internal fields and searches for data to be written out This is currently relatively inefficient

Definition at line 94 of file FieldCollector.hpp.

7.12.2 Member Function Documentation

7.12.2.1 template<typename T > data_vector<T> GlobalFlow::DataProcessing::DataOutput::FieldCollector::get (Simulation::Simulation const & *simulation*) [inline]

Collects the data from simulation nodes

Note

Relatively inefficient currently

Parameters

simulation	The simulation

Returns

The collected data

Definition at line 130 of file FieldCollector.hpp.

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/FieldCollector.hpp

7.13 GlobalFlow::DataProcessing::DataOutput::FieldFactory Class Reference

Static Public Member Functions

- static field_vector build (boost::property_tree::ptree tree, Simulation::Simulation const &sim)

7.13.1 Detailed Description

Definition at line 120 of file OutputManager.hpp.

7.13.2 Member Function Documentation

7.13.2.1 static field_vector GlobalFlow::DataProcessing::DataOutput::FieldFactory::build (boost::property_tree::ptree *tree*, Simulation::Simulation const & *sim*) `[inline]`, `[static]`

Parameters

tree	

Definition at line 131 of file OutputManager.hpp.

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputManager.hpp

7.14 GlobalFlow::Model::FlowTypeHash Struct Reference

Public Member Functions

- template<typename T >
  std::size_t **operator()** (T t) const

7.14.1 Detailed Description

Definition at line 69 of file ExternalFlows.hpp.

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

- /home/robert/development/G3M-framework/src/Model/ExternalFlows.hpp

7.15 GlobalFlow::Model::FluidMechanics Class Reference

```
#include <FluidMechanics.hpp>
```

Public Member Functions

- t_meter calcDeltaV (t_meter head, t_meter elevation, t_meter depth) noexcept
- t_s_meter_t calculateHarmonicMeanConductance (t_vel k_neig, t_vel k_self, t_meter edgeLength_neig, t_meter edgeLength_self, t_meter head_neig, t_meter head_self, t_meter ele_neig, t_meter ele_self, t_meter deltaV_neig, t_meter deltaV_self, bool confined) noexcept

  Calculates the horizontal flow between two nodes.
- double smoothFunction__NWT (t_meter elevation, t_meter verticalSize, t_meter head)
- t_s_meter_t getHCOF (bool steadyState, quantity< Dimensionless > stepModifier, t_s_meter storageCapacity, t_s_meter_t P) noexcept
- t_s_meter_t calculateVerticalConductance (t_vel k_vert_neig, t_vel k_vert_self, t_meter verticalSize_self, t_meter head_self, t_meter elevation_self, t_s_meter area_self, t_meter elevation_neig, t_meter depth_neig, t_meter head_neig, bool confined) noexcept
- double getDerivate__NWT (t_meter elevation, t_meter verticalSize, t_meter head)

7.15.1 Detailed Description

Provides helper functions for conductance calulcations

Definition at line 36 of file FluidMechanics.hpp.

7.15.2 Member Function Documentation

7.15.2.1 t_meter GlobalFlow::Model::FluidMechanics::calcDeltaV (t_meter *head*, t_meter *elevation*, t_meter *depth*) [noexcept]

Used to calculate if a cell is dry

Definition at line 7 of file FluidMechanics.cpp.

7.15.2.2 quantity< MeterSquaredPerTime > GlobalFlow::Model::FluidMechanics::calculateHarmonicMeanConductance (t_vel *k_neig*, t_vel *k_self*, t_meter *edgeLength_neig*, t_meter *edgeLength_self*, t_meter *head_neig*, t_meter *head_self*, t_meter *ele_neig*, t_meter *ele_self*, t_meter *deltaV_neig*, t_meter *deltaV_self*, bool *confined*) [noexcept]

Calculates the horizontal flow between two nodes.

Parameters

k_neig	K-value of neighbouring node
edgeLength_↩ neig	The edge size of the neighbouring node (assumes rectengular cells)

Returns

A weighted conductance value for the flow between two nodes

Calculates the harmonic mean conductance between two nodes. C = 2∗EdgeLenght_1 ∗ (TR_1 ∗ TR_2) / (TR_1 ∗ EdgeLenght_1 + TR_2 ∗ EdgeLenght_2)

Definition at line 18 of file FluidMechanics.cpp.

7.15.2.3 quantity< MeterSquaredPerTime > GlobalFlow::Model::FluidMechanics::calculateVerticalConductance (t_vel *k_vert_neig*, t_vel *k_vert_self*, t_meter *verticalSize_self*, t_meter *head_self*, t_meter *elevation_self*, t_s_meter *area_self*, t_meter *elevation_neig*, t_meter *depth_neig*, t_meter *head_neig*, bool *confined*) [noexcept]

Calculates the vertical flow between two nodes

Parameters

k_vert_neig	
k_vert_self	
verticalSize_self	
head_self	
elevation_self	
area_self	
elevation_neig	
depth_neig	
head_neig	

confined	

Returns

Definition at line 76 of file FluidMechanics.cpp.

7.15.2.4 double GlobalFlow::Model::FluidMechanics::getDerivate__NWT (t_meter *elevation,* t_meter *verticalSize,* t_meter *head*)

Calculate derivates for NWT approach

Parameters

elevation	
verticalSize	
head	

Returns

Definition at line 115 of file FluidMechanics.cpp.

7.15.2.5 quantity< MeterSquaredPerTime > GlobalFlow::Model::FluidMechanics::getHCOF (bool *steadyState,* quantity< Dimensionless > *stepModifier,* t_s_meter *storageCapacity,* t_s_meter_t *P*) [noexcept]

Get the coeffiecients for storage and P components

Parameters

steadyState	
stepModifier	
storageCapacity	
P	

Returns

Definition at line 103 of file FluidMechanics.cpp.

7.15.2.6 double GlobalFlow::Model::FluidMechanics::smoothFunction__NWT (t_meter *elevation,* t_meter *verticalSize,* t_meter *head*)

Simple smoother function to buffer iteration steps in NWT approach

Parameters

elevation	
verticalSize	
head	

Returns

Definition at line 55 of file FluidMechanics.cpp.

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

- /home/robert/development/G3M-framework/src/Model/FluidMechanics.hpp
- /home/robert/development/G3M-framework/src/Model/FluidMechanics.cpp

7.16 GlobalFlow::DataProcessing::DataOutput::GFS_JSONOutput< T > Class Template Reference

Inheritance diagram for GlobalFlow::DataProcessing::DataOutput::GFS_JSONOutput< T >:

Collaboration diagram for GlobalFlow::DataProcessing::DataOutput::GFS_JSONOutput< T >:

Public Member Functions

- void write (path filePath, bool printID, bool printXY, std::vector< T > data, pos_v p)

7.16.1 Detailed Description

template<typename T>class GlobalFlow::DataProcessing::DataOutput::GFS_JSONOutput< T >

Definition at line 155 of file OutputFactory.hpp.

7.16.2 Member Function Documentation

7.16.2.1 template<typename T > void **GlobalFlow::DataProcessing::DataOutput::GFS_JSONOutput**< T >::write (path *filePath,* bool *printID,* bool *printXY,* std::vector< T > *data,* pos_v *p*) [inline], [virtual]

Needs to be implemented

Parameters

filePath	
printID	Bool
printXY	Bool
data	Data vector
p	Position vector

Implements GlobalFlow::DataProcessing::DataOutput::OutputInterface< T >.

Definition at line 299 of file OutputFactory.hpp.

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputFactory.hpp

7.17 GlobalFlow::GW_Interface Class Reference

Public Member Functions

- virtual void loadSettings ()=0

- virtual void setupSimulation ()=0
- virtual void writeData ()=0
- void setupCallBack (Callback callback)
- virtual void simulate ()=0

Protected Attributes

- std::function< void(Container container)> **callback**

7.17.1 Detailed Description

Definition at line 72 of file GW_Interface.hpp.

7.17.2 Member Function Documentation

7.17.2.1 virtual void GlobalFlow::GW_Interface::loadSettings ( ) [pure virtual]

Read general simulation settings e.g. Options

7.17.2.2 void GlobalFlow::GW_Interface::setupCallBack (Callback *callback*) [inline]

Set up the calback for model coupling Could be inefficient

Parameters

callback	

Definition at line 98 of file GW_Interface.hpp.

7.17.2.3 virtual void GlobalFlow::GW_Interface::setupSimulation ( ) [pure virtual]

Do additional work required for a running simulation

7.17.2.4 virtual void GlobalFlow::GW_Interface::simulate ( ) [pure virtual]

Simulate/Run the model

7.17.2.5 virtual void GlobalFlow::GW_Interface::writeData ( ) [pure virtual]

How should the data be written out

Deprecated

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

- /home/robert/development/G3M-framework/src/GW_Interface.hpp

7.18 std::hash< GlobalFlow::Model::NeighbourPosition > Struct Template Reference

Public Types

- typedef GlobalFlow::Model::NeighbourPosition **argument_type**

- typedef std::underlying_type< argument_type >::type **underlying_type**
- typedef std::hash< underlying_type >::result_type **result_type**

Public Member Functions

- result_type **operator()** (const argument_type &arg) const

7.18.1 Detailed Description

template<>struct std::hash< GlobalFlow::Model::NeighbourPosition >

Definition at line 63 of file Node.hpp.

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

- /home/robert/development/G3M-framework/src/Model/Node.hpp

7.19 GlobalFlow::Logging::InfoSinkInterface Class Reference

```
#include <Sinks.hpp>
```

Inheritance diagram for GlobalFlow::Logging::InfoSinkInterface:

Collaboration diagram for GlobalFlow::Logging::InfoSinkInterface:

Public Member Functions

- **InfoSinkInterface** (std::string file_name)
- virtual void **consume** (logboost::record_view const &rec)=0
- void **flush** ()
- void **write_data** ()

7.19.1 Detailed Description

Deprecated Use methods from DataProcessing

Definition at line 39 of file Sinks.hpp.

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

- /home/robert/development/G3M-framework/src/Logging/Sinks.hpp

7.20 Interface Interface Reference

7.20.1 Detailed Description

to be implemented for reading in required data for the model

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataReader.hpp

7.21 GlobalFlow::DataProcessing::DataOutput::InternalTypeHash Struct Reference

Public Member Functions

- template< typename T >
  std::size_t **operator()** (T t) const

7.21.1 Detailed Description

Definition at line 87 of file OutputManager.hpp.

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputManager.hpp

7.22 Is< T > Struct Template Reference

```
#include <Helpers.hpp>
```

Public Member Functions

- bool **in** (T a)
- template<class Arg , class... Args>
  bool **in** (Arg a, Args...args)

Public Attributes

- T **d_**

7.22.1 Detailed Description

template<class T>struct Is< T >

http://stackoverflow.com/questions/15181579

Definition at line 60 of file Helpers.hpp.

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

- /home/robert/development/G3M-framework/src/Misc/Helpers.hpp

7.23 GlobalFlow::Simulation::Iterator Class Reference

Public Member Functions

- **Iterator** (const AbstractStepper ∗stepper, TimeFrame time, int pos)
- bool **operator!=** (const Iterator &other) const
- step **operator∗** () const
- const Iterator & **operator++** ()
- int **getPos** ()

7.23.1 Detailed Description

iterator holding the current simulation step

Definition at line 43 of file Stepper.hpp.

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

- /home/robert/development/G3M-framework/src/Simulation/Stepper.hpp

7.24 GlobalFlow::Logging::Logger Class Reference

Inheritance diagram for GlobalFlow::Logging::Logger:

Collaboration diagram for GlobalFlow::Logging::Logger:

Additional Inherited Members

7.24.1 Detailed Description

Definition at line 80 of file Logging.hpp.

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

- /home/robert/development/G3M-framework/src/Logging/Logging.hpp

7.25 GlobalFlow::Logging::LoggerInterface Class Reference

Inheritance diagram for GlobalFlow::Logging::LoggerInterface:

Collaboration diagram for GlobalFlow::Logging::LoggerInterface:

Friends

- class **Singleton**< **LoggerInterface** >

Additional Inherited Members

7.25.1 Detailed Description

Definition at line 68 of file Logging.hpp.

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

- /home/robert/development/G3M-framework/src/Logging/Logging.hpp

7.26 GlobalFlow::Simulation::MassError Class Reference

```
#include <Simulation.hpp>
```

Public Member Functions

- **MassError** (mpf_float_1000 OUT, mpf_float_1000 IN, mpf_float ERR)

Public Attributes

- mpf_float_1000 **OUT**
- mpf_float_1000 **IN**
- mpf_float **ERR** = 0

7.26.1 Detailed Description

Simple container for the mass error calulations

Definition at line 55 of file Simulation.hpp.

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

- /home/robert/development/G3M-framework/src/Simulation/Simulation.hpp

7.27 GlobalFlow::DataProcessing::DataOutput::NETCDFOutput< T > Class Template Reference

Inheritance diagram for GlobalFlow::DataProcessing::DataOutput::NETCDFOutput< T >:

Collaboration diagram for GlobalFlow::DataProcessing::DataOutput::NETCDFOutput< T >:

Public Member Functions

- void write (path filePath, bool printID, bool printXY, std::vector< T > data, pos_v p)

7.27.1 Detailed Description

template<typename T>class GlobalFlow::DataProcessing::DataOutput::NETCDFOutput< T >

Definition at line 112 of file OutputFactory.hpp.

7.27.2 Member Function Documentation

7.27.2.1 template< typename T > void GlobalFlow::DataProcessing::DataOutput::NETCDFOutput< T >::write (path *filePath*, bool *printID*, bool *printXY*, std::vector< T > *data*, pos_v *p*) [inline], [virtual]

Needs to be implemented

Parameters

filePath	
printID	Bool
printXY	Bool
data	Data vector
p	Position vector

Implements GlobalFlow::DataProcessing::DataOutput::OutputInterface< T >.

Definition at line 114 of file OutputFactory.hpp.

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputFactory.hpp

7.28 GlobalFlow::Model::NodeInterface Class Reference

```
#include <Node.hpp>
```

Inheritance diagram for GlobalFlow::Model::NodeInterface:

Collaboration diagram for GlobalFlow::Model::NodeInterface:

Classes

- class NodeNotFoundException

Public Member Functions

- NodeInterface (NodeVector nodes, double lat, double lon, t_s_meter area, unsigned long ArcID, unsigned long ID, t_vel K, int stepModifier, double aquiferDepth, double anisotropy, double specificYield, double specificStorage, bool confined)

  Constructor of abstract class NodeInterface.
- unsigned long **getID** ()
- void setElevation (t_meter elevation)

  Set elevation on top layer and propagate to lower layers.
- void setSlope (double slope_percent)

  Set slope from data on all layers Slope input is in % but is required as absolut thus: slope = sloper_percent / 100.
- void setEfold (double efold)

  Set e-folding factor from data on all layers.
- void setEqHead (t_meter wtd)

  Calculated equilibrium groundwater-head from eq_wtd Assumes that if initialhead = false that the eq_head is also used as initial head.
- template<class HeadType >
  t_vol_t calcLateralFlows ()
- t_vol_t getEqFlow () noexcept
- t_vol_t getLateralFlows ()
- bool resetFloodingHead () noexcept

  Cuts off all heads above surface elevation.
- void scaleRiverConduct ()

  Scales river conduct by 50%.
- void updateHeadChange () noexcept

  Update the current head change (in comparison to last time step)
- void **initHead_t0** () noexcept
- void **setHead_direct** (double head) noexcept
- t_vel **getK__pure** () noexcept
- t_vel getK () noexcept

  Get hydraulic conductivity.
- t_vel getK_vertical () noexcept

  Get hydraulic vertical conductivity.
- void setK (t_vel conduct)

  Modify hydraulic conductivity (applied to all layers below)
- void setK_direct (t_vel conduct)

  Modify hydraulic conductivity (no e-folding, no layers)
- t_c_meter getOUT () noexcept

  Get all outflow since simulation start.
- t_c_meter getIN () noexcept

  Get all inflow since simulation start.

- void toogleStadyState (bool onOFF)

  Toogle steady state simulation.
- void **updateStepSize** (double mod)
- t_s_meter getStorageCapacity () noexcept

  Storage capacity based on yield or specific storage.
- ExternalFlow & getExternalFlowByName (FlowType type) throw (out_of_range)

  Get and external flow by its FlowType.
- t_vol_t getExternalFlowVolumeByName (FlowType type)

  Get and external flow volume by its FlowType.
- t_vol_t getTotalStorageFlow () noexcept

  Get flow budget based on head change.
- t_vol_t calculateExternalFlowVolume (const ExternalFlow &flow)

  Get flow budget of a specific external flows.
- t_vol_t calculateDewateredFlow () noexcept

  Caluclate dewatered flow.
- t_vol_t getCurrentIN () noexcept

  Get all current IN flow.
- t_vol_t getCurrentOUT () noexcept

  Get all current OUT flow.
- void saveMassBalance () noexcept

  Tell cell to save its flow budget.
- void setNeighbour (unsigned long ID, NeighbourPosition neighbour)

  Add a neighbour.
- int **getNumofNeighbours** ()
- NodeInterface ∗ getNeighbour (NeighbourPosition neighbour) noexcept(false)

  Get a neighbour by position.
- int addExternalFlow (FlowType type, t_meter flowHead, double cond, t_meter bottom)

  At an external flow to the cell.
- void removeExternalFlow (FlowType type)

  Remove an external flow to the cell by id.
- bool hasTypeOfExternalFlow (FlowType type)

  Check for an external flow by type.
- void updateUniqueFlow (double amount, FlowType flow=RECHARGE)

  Updates GW recharge Curently assumes only one recharge as external flow!
- void scaleDynamicRivers (double mult)
- void updateExternalFlowConduct (double amount, FlowType type)

  Update wetlands, lakes.
- void updateLakeBottoms (double amount)

  Update lake bottoms Used for sensitivity.
- bool hasRiver ()

  Check for type river.
- bool hasOcean ()

  Check for type ocean.
- t_vol_t getQ () noexcept

  Get Q part of flow equations.
- t_s_meter_t getP () noexcept

  Get P part of flow equations.
- t_vol_t calculateNotHeadDependandFlows () noexcept

  Get flow which is not groundwater head dependent.
- std::unordered_map< int, t_s_meter_t > getJacobian () noexcept

  The jacobian entry for the cell (NWT approach)

- std::unordered_map< int, t_s_meter_t > getConductance ()

  The matrix entry for the cell.
- t_vol_t getRHS () noexcept

  The right hand side of the equation.
- double getRHS__NWT () noexcept

  The right hand side of the equation (NWT)
- void **setHead** (t_meter head) noexcept
- t_meter **calcInitialHead** (t_meter initialParam) noexcept
- bool **isStaticNode** () noexcept
- PhysicalProperties & **getProperties** ()
- void **enableNWT** ()
- template<typename CompareFunction >
  t_vol_t getNonStorageFlow (CompareFunction compare) noexcept

  Caluclate non storage related in and out flow.
- quantity< Velocity > getVelocity (map_itter pos)
- std::pair< double, double > getVelocityVector ()

  Calculate flow velocity for flow tracking Vx and Vy represent the flow velocity in x and y direction. A negative value represents a flow in the oposite direction.

Public Attributes

- bool cached {false}
- t_vol_t **eq_flow** {0 ∗ si::cubic_meter / day}

Protected Types

- using **map_itter** = std::unordered_map< NeighbourPosition, unsigned long >::const_iterator

Protected Member Functions

- template<typename T , typename F >
  T **get** ()
- template<typename T , typename F >
  void **set** (const T &value)
- template<typename T , typename F >
  T **getFrom** (NodeInterface ∗nodeInterface, NeighbourPosition pos)
- p_node & **at** (map_itter pos)
- template<typename T , typename F >
  T **getAt** (map_itter pos)
- t_s_meter getStorageCapacity__Primary () noexcept

  Uses specific storage to caluclate storativity.
- t_s_meter getStorageCapacity__Secondary () noexcept

  Uses specific yield to caluclate storativity.
- t_s_meter getStorageCapacity__SecondaryNWT () noexcept

  Uses specific yield with NWT smoother.
- template<typename CompareFunction >
  t_vol_t getFlow (CompareFunction compare) noexcept

  Flow volume of cell.

Protected Attributes

- const std::shared_ptr< std::vector< std::unique_ptr< NodeInterface > > > **nodes**
- unordered_map< NeighbourPosition, unsigned long > **neighbours**
- unordered_map< FlowType, ExternalFlow, FlowTypeHash > **externalFlows**
- int **numOfExternalFlows** {0}
- bool **nwt** {false}
- bool **initial_head** {true}
- bool **simpleDistance** {false}
- bool **simpleK** {false}
- bool **steadyState** {false}
- FluidMechanics **mechanics**
- PhysicalProperties **fields**

Friends

- class **FluidMechanics**

7.28.1 Detailed Description

Interface defining required fields for a node. A node is the central comutational and spatial unit. A simulated area is seperated into a discrete raster of cells or nodes (seperate computational units which stay in contact to ech other). Is equal to 'cell'.

Nodes can be of different physical property e.g. different size.

Definition at line 96 of file Node.hpp.

7.28.2 Constructor & Destructor Documentation

7.28.2.1 GlobalFlow::Model::NodeInterface::NodeInterface (NodeVector *nodes,* double *lat,* double *lon,* t_s_meter *area,* unsigned long *ArcID,* unsigned long *ID,* t_vel *K,* int *stepModifier,* double *aquiferDepth,* double *anisotropy,* double *specificYield,* double *specificStorage,* bool *confined*)

Constructor of abstract class NodeInterface.

Parameters

nodes	Vector of all other existing nodes
lat	The latitude
lon	The Longitude
area	Area in m²
ArcID	Unique ARC-ID specified by Kassel
ID	Internal ID = Position in vector
K	Hydraulic conductivity in meter/day (default)
stepModifier	Modfies default step size of day (default=1)
aquiferDepth	Vertical size of the cell
anisotropy	Modifier for vertical conductivity based on horizontal
specificYield	Yield of storage for dewatered conditions
specificStorage	Specific storage - currently for confined and unconfined

confined	Is node in a confined layer

Definition at line 42 of file Node.cpp.

7.28.3 Member Function Documentation

7.28.3.1 int GlobalFlow::Model::NodeInterface::addExternalFlow (FlowType *type,* t_meter *flowHead,* double *cond,* t_meter *bottom*) [inline]

At an external flow to the cell.

Parameters

type	The flow type
flowHead	The flow head
cond	The conductance
bottom	The bottom of the flow (e.g river bottom)

Returns

Number assigned by cell to flow

Definition at line 711 of file Node.hpp.

7.28.3.2 template<class HeadType > t_vol_t GlobalFlow::Model::NodeInterface::calcLateralFlows ( ) [inline]

Calculate the lateral groundwater flow to the neighbouring nodes Generic function used for calulating equlibrium and current step flow

Returns

Definition at line 357 of file Node.hpp.

Here is the call graph for this function:

7.28.3.3 t_vol_t GlobalFlow::Model::NodeInterface::calculateDewateredFlow ( ) [inline], [noexcept]

Caluclate dewatered flow.

Returns

Flow volume per time If a cell is dewatered but below a saturated or partly saturated cell: this calculates the needed additional exchange volume

Definition at line 605 of file Node.hpp.

7.28.3.4 t_vol_t GlobalFlow::Model::NodeInterface::calculateExternalFlowVolume (const ExternalFlow & *flow*) [inline]

Get flow budget of a specific external flows.

Parameters

&flow	A external flow

Returns

Flow volume Note: Water entering storage is treated as an outflow (-), that is a loss of water from the flow system while water released from storage is treated as inflow (+), that is a source of water to the flow system

Definition at line 568 of file Node.hpp.

Here is the call graph for this function:

7.28.3.5 t_vol_t GlobalFlow::Model::NodeInterface::calculateNotHeadDependandFlows ( ) [inline], [noexcept]

Get flow which is not groundwater head dependent.

Returns

volume over time Flow can be added to constant flows on right side of the equations If head is above river bottom for example

Definition at line 884 of file Node.hpp.

7.28.3.6 std::unordered_map<int, t_s_meter_t> GlobalFlow::Model::NodeInterface::getConductance ( ) [inline]

The matrix entry for the cell.

Returns

map <CellID,Conductance> The left hand side of the equation

Definition at line 958 of file Node.hpp.

7.28.3.7 t_vol_t GlobalFlow::Model::NodeInterface::getCurrentIN ( ) [inline], [noexcept]

Get all current IN flow.

Returns

Flow volume

Definition at line 659 of file Node.hpp.

7.28.3.8 t_vol_t GlobalFlow::Model::NodeInterface::getCurrentOUT ( ) [inline], [noexcept]

Get all current OUT flow.

Returns

Flow volume

Definition at line 665 of file Node.hpp.

7.28.3.9 t_vol_t GlobalFlow::Model::NodeInterface::getEqFlow ( ) [inline], [noexcept]

Calculate the equlibrium lateral flows

Returns

eq lateral flow

Definition at line 391 of file Node.hpp.

7.28.3.10 ExternalFlow& GlobalFlow::Model::NodeInterface::getExternalFlowByName (FlowType *type*) throw out_of_range) [inline]

Get and external flow by its FlowType.

Parameters

type	The flow type

Returns

Ref to external flow

Exceptions

OutOfRangeException	

Definition at line 530 of file Node.hpp.

7.28.3.11 t_vol_t GlobalFlow::Model::NodeInterface::getExternalFlowVolumeByName (FlowType *type*) [inline]

Get and external flow volume by its FlowType.

Parameters

type	The flow type

Returns

Flow volume

Definition at line 541 of file Node.hpp.

7.28.3.12 template<typename CompareFunction > t_vol_t GlobalFlow::Model::NodeInterface::getFlow (CompareFunction *compare*) [inline], [protected], [noexcept]

Flow volume of cell.

Returns

Flow volume

Definition at line 222 of file Node.hpp.

7.28.3.13 std::unordered_map<int, t_s_meter_t> GlobalFlow::Model::NodeInterface::getJacobian ( ) [inline], [noexcept]

The jacobian entry for the cell (NWT approach)

Returns

map <CellID,Conductance>

Definition at line 911 of file Node.hpp.

7.28.3.14 t_vel GlobalFlow::Model::NodeInterface::getK ( ) [inline], [noexcept]

Get hydraulic conductivity.

Returns

hydraulic conductivity (scaled by e-folding)

Definition at line 451 of file Node.hpp.

7.28.3.15 t_vel GlobalFlow::Model::NodeInterface::getK_vertical ( ) [inline], [noexcept]

Get hydraulic vertical conductivity.

Returns

hydraulic conductivity scaled by anisotropy (scaled by e-folding)

Definition at line 468 of file Node.hpp.

7.28.3.16 t_vol_t GlobalFlow::Model::NodeInterface::getLateralFlows ( ) [inline]

Get the current lateral flow

Returns

Definition at line 405 of file Node.hpp.

7.28.3.17 NodeInterface* GlobalFlow::Model::NodeInterface::getNeighbour (NeighbourPosition *neighbour*) [inline], [noexcept]

Get a neighbour by position.

Parameters

neighbour	The position relative to the cell

Returns

Pointer to cell object

Definition at line 693 of file Node.hpp.

7.28.3.18 template<typename CompareFunction > t_vol_t GlobalFlow::Model::NodeInterface::getNonStorageFlow (CompareFunction *compare*) [inline], [noexcept]

Caluclate non storage related in and out flow.

Returns

Flow volume

Definition at line 1101 of file Node.hpp.

7.28.3.19 t_s_meter_t GlobalFlow::Model::NodeInterface::getP ( ) [inline], [noexcept]

Get P part of flow equations.

Returns

volume over time

Definition at line 855 of file Node.hpp.

7.28.3.20 t_vol_t GlobalFlow::Model::NodeInterface::getQ ( ) [inline], [noexcept]

Get Q part of flow equations.

Returns

volume over time

Definition at line 834 of file Node.hpp.

7.28.3.21 t_vol_t GlobalFlow::Model::NodeInterface::getRHS ( ) [inline], [noexcept]

The right hand side of the equation.

Returns

volume per time

Definition at line 1045 of file Node.hpp.

7.28.3.22 double GlobalFlow::Model::NodeInterface::getRHS__NWT ( ) [inline], [noexcept]

The right hand side of the equation (NWT)

Returns

volume per time

Definition at line 1064 of file Node.hpp.

7.28.3.23 t_s_meter GlobalFlow::Model::NodeInterface::getStorageCapacity ( ) [inline], [noexcept]

Storage capacity based on yield or specific storage.

Returns

Potential flow budget when multiplied by head change Uses an 0.001m epsilon to determine if a water-table condition is present. If the layer is confined or not in water-table condition returns primary capacity.

Definition at line 512 of file Node.hpp.

7.28.3.24 t_s_meter GlobalFlow::Model::NodeInterface::getStorageCapacity__Primary ( ) [inline], [protected], [noexcept]

Uses specific storage to caluclate storativity.

Returns

Flow budget for cell depending on head change

Definition at line 191 of file Node.hpp.

7.28.3.25 t_s_meter GlobalFlow::Model::NodeInterface::getStorageCapacity__Secondary ( ) [inline], [protected], [noexcept]

Uses specific yield to caluclate storativity.

Returns

Flow budget for cell depending on head change

Definition at line 201 of file Node.hpp.

7.28.3.26 t_s_meter GlobalFlow::Model::NodeInterface::getStorageCapacity__SecondaryNWT ( ) [inline], [protected], [noexcept]

Uses specific yield with NWT smoother.

Returns

Flow budget for cell depending on head change Assumes that smooth function is linear

Definition at line 210 of file Node.hpp.

Here is the call graph for this function:

7.28.3.27 t_vol_t GlobalFlow::Model::NodeInterface::getTotalStorageFlow ( ) [inline], [noexcept]

Get flow budget based on head change.

Returns

Flow volume Note: Water entering storage is treated as an outflow (-), that is a loss of water from the flow system while water released from storage is treated as inflow (+), that is a source of water to the flow system

Definition at line 557 of file Node.hpp.

7.28.3.28 quantity<Velocity> GlobalFlow::Model::NodeInterface::getVelocity (map_itter *pos*) [inline]

Calculate the lateral flow velocity

Parameters

pos	

Returns

Definition at line 1122 of file Node.hpp.

7.28.3.29 std::pair<double, double> GlobalFlow::Model::NodeInterface::getVelocityVector ( ) [inline]

Calculate flow velocity for flow tracking Vx and Vy represent the flow velocity in x and y direction. A negative value represents a flow in the oposite direction.

Returns

Velocity vector (x,y)

Definition at line 1147 of file Node.hpp.

7.28.3.30 bool GlobalFlow::Model::NodeInterface::hasOcean ( ) [inline]

Check for type ocean.

Returns

bool

Definition at line 827 of file Node.hpp.

7.28.3.31 bool GlobalFlow::Model::NodeInterface::hasRiver ( ) [inline]

Check for type river.

Returns

bool

Definition at line 821 of file Node.hpp.

7.28.3.32 bool GlobalFlow::Model::NodeInterface::hasTypeOfExternalFlow (FlowType *type*) [inline]

Check for an external flow by type.

Parameters

type	The flow type

Returns

bool

Definition at line 754 of file Node.hpp.

7.28.3.33 void GlobalFlow::Model::NodeInterface::removeExternalFlow (FlowType *type*) [inline]

Remove an external flow to the cell by id.

Parameters

ID	The flow id

Definition at line 743 of file Node.hpp.

7.28.3.34 bool GlobalFlow::Model::NodeInterface::resetFloodingHead ( ) [inline], [noexcept]

Cuts off all heads above surface elevation.

Warning

Should only be used in spinn up phase!

Returns

Bool if node was reset

Definition at line 414 of file Node.hpp.

7.28.3.35 void GlobalFlow::Model::NodeInterface::scaleDynamicRivers (double *mult*) [inline]

Scale dyn rivers for sensitivity

Parameters

mult	

Definition at line 780 of file Node.hpp.

7.28.3.36 void GlobalFlow::Model::NodeInterface::scaleRiverConduct ( ) [inline]

Scales river conduct by 50%.

Warning

Should only be used in spinn up phase

Definition at line 427 of file Node.hpp.

7.28.3.37 void GlobalFlow::Model::NodeInterface::setEfold (double *efold*) [inline]

Set e-folding factor from data on all layers.

Parameters

e-fold	

Definition at line 314 of file Node.hpp.

Here is the call graph for this function:

7.28.3.38 void GlobalFlow::Model::NodeInterface::setElevation (t_meter *elevation*) [inline]

Set elevation on top layer and propagate to lower layers.

Parameters

elevation	The top elevation (e.g. from DEM)

Definition at line 279 of file Node.hpp.

7.28.3.39 void GlobalFlow::Model::NodeInterface::setEqHead (t_meter *wtd*) [inline]

Calculated equilibrium groundwater-head from eq_wtd Assumes that if initialhead = false that the eq_head is also used as initial head.

Parameters

head	

Definition at line 328 of file Node.hpp.

7.28.3.40 void GlobalFlow::Model::NodeInterface::setK (t_vel *conduct*) [inline]

Modify hydraulic conductivity (applied to all layers below)

Parameters

New	conductivity (if e-folding enabled scaled on layers)

Definition at line 474 of file Node.hpp.

7.28.3.41 void GlobalFlow::Model::NodeInterface::setK_direct (t_vel *conduct*) [inline]

Modify hydraulic conductivity (no e-folding, no layers)

Parameters

New	conductivity

Definition at line 485 of file Node.hpp.

7.28.3.42 void GlobalFlow::Model::NodeInterface::setNeighbour (unsigned long *ID,* NeighbourPosition *neighbour*) [inline]

Add a neighbour.

Parameters

ID	The internal ID and position in vector
neighbour	The position relative to the cell

Definition at line 680 of file Node.hpp.

7.28.3.43 void GlobalFlow::Model::NodeInterface::setSlope (double *slope_percent*) [inline]

Set slope from data on all layers Slope input is in % but is required as absolut thus: slope = sloper_percent / 100.

Parameters

slope	

Definition at line 299 of file Node.hpp.

Here is the call graph for this function:

7.28.3.44 void GlobalFlow::Model::NodeInterface::toogleStadyState (bool *onOFF*) [inline]

Toogle steady state simulation.

Parameters

onOFF	true=on Turns all storage equations to zero with no timesteps

Definition at line 502 of file Node.hpp.

7.28.3.45 void GlobalFlow::Model::NodeInterface::updateExternalFlowConduct (double *amount,* FlowType *type*) [inline]

Update wetlands, lakes.

Parameters

amount	
type	

Definition at line 792 of file Node.hpp.

7.28.3.46 void GlobalFlow::Model::NodeInterface::updateHeadChange ( ) [inline], [noexcept]

Update the current head change (in comparison to last time step)

Note

Should only be caled at end of timestep

Definition at line 435 of file Node.hpp.

7.28.3.47 void GlobalFlow::Model::NodeInterface::updateLakeBottoms (double *amount*) [inline]

Update lake bottoms Used for sensitivity.

Parameters

amount	

Definition at line 807 of file Node.hpp.

7.28.3.48 void GlobalFlow::Model::NodeInterface::updateUniqueFlow (double *amount,* FlowType *flow* = RECHARGE) [inline]

Updates GW recharge Curently assumes only one recharge as external flow!

Parameters

amount	The new flow amount

Definition at line 763 of file Node.hpp.

7.28.4 Member Data Documentation

7.28.4.1 bool GlobalFlow::Model::NodeInterface::cached {false}

Calculated equilibrium flow to neighbouring cells Static thus calculated only once.

Depends on: K in cell and eq_head in all 6 neighbours

Definition at line 348 of file Node.hpp.

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

- /home/robert/development/G3M-framework/src/Model/Node.hpp
- /home/robert/development/G3M-framework/src/Model/Node.cpp

7.29 GlobalFlow::Model::NodeInterface::NodeNotFoundException Class Reference

Inheritance diagram for GlobalFlow::Model::NodeInterface::NodeNotFoundException:

Collaboration diagram for GlobalFlow::Model::NodeInterface::NodeNotFoundException:

7.29.1 Detailed Description

Definition at line 684 of file Node.hpp.

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

- /home/robert/development/G3M-framework/src/Model/Node.hpp

7.30 GlobalFlow::Simulation::Options Class Reference

```
#include <Options.hpp>
```

Public Types

- enum **BoundaryCondition** { **CONSTANT_HEAD_SEA_LEVEL**, **CONSTANT_HEAD_NEIGHBOUR**, **STATIC_HEAD_SEA_LEVEL** }

Public Member Functions

- void **setClosingCrit** (double crit)
- void **setDamping** (bool set)
- bool **isDampingEnabled** ()
- double **getMinDamp** ()
- double **getMaxDamp** ()
- double **getMaxHeadChange** ()
- bool **isConfined** (int layer)
- vector< bool > **getConfinements** ()
- BoundaryCondition **getBoundaryCondition** ()
- bool **isSensitivity** ()
- bool **isKFromLith** ()
- bool **isKOceanFile** ()
- bool **isSpecificStorageFile** ()
- bool **isSpecificYieldFile** ()
- bool **isKRiverFile** ()
- bool **isAquiferDepthDile** ()
- string **getKDir** ()
- string **getKRiverDir** ()
- string **getKOceanDir** ()
- string **getSSDir** ()
- string **getSYDir** ()
- string **getAQDepthDir** ()
- bool **isRowCol** ()
- int **getInnerItter** ()
- long **getNumberOfNodes** ()
- int **getNumberOfLayers** ()
- int **getMaxIterations** ()
- double **getConverganceCriteria** ()
- string **getSolverName** ()
- string **getNodesDir** ()
- string **getElevation** ()
- string **getEfolding** ()
- string **getEqWTD** ()
- string **getSlope** ()

- string **getBlue** ()
- vector< string > **getElevation_A** ()
- vector< string > **getEfolding_a** ()
- vector< string > **getEqWTD_a** ()
- vector< string > **getSlope_a** ()
- vector< string > **getBlue_a** ()
- string **getRecharge** ()
- string **getLithology** ()
- string **getRivers** ()
- string **getGlobalLakes** ()
- string **getGlobalWetlands** ()
- string **getLocalLakes** ()
- string **getLocalWetlands** ()
- string **getMapping** ()
- int **getThreads** ()
- const bool **adaptiveStepsizeEnabled** ()
- const int **getStepsizeModifier** () throw (out_of_range)
- bool **cacheEnabled** ()
- int **getInitialHead** ()
- double **getInitialK** ()
- double **getOceanConduct** ()
- vector< int > **getAquiferDepth** ()
- double **getAnisotropy** ()
- double **getSpecificYield** ()
- double **getSpecificStorage** ()
- void **load** (const std::string &filename)
- void **save** (const std::string &filename)

7.30.1 Detailed Description

Reads simulation options from a JSON file Defines getters and setters for options

Definition at line 37 of file Options.hpp.

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

- /home/robert/development/G3M-framework/src/Simulation/Options.hpp
- /home/robert/development/G3M-framework/src/Simulation/Options.cpp

7.31 GlobalFlow::DataProcessing::DataOutput::OutputFactory< T > Class Template Reference

Static Public Member Functions

- static OutputInterface< T > ∗ **getOutput** (OutputType type)

7.31.1 Detailed Description

template<typename T>class GlobalFlow::DataProcessing::DataOutput::OutputFactory< T >

Definition at line 312 of file OutputFactory.hpp.

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputFactory.hpp

7.32 GlobalFlow::DataProcessing::DataOutput::OutputField< T > Class Template Reference

Public Member Functions

- **OutputField** (path filePath, bool printID, bool printXY, Simulation::Simulation const &sim, OutputType outputType, FieldType fieldType)
- void **write** ()

7.32.1 Detailed Description

template<typename T>class GlobalFlow::DataProcessing::DataOutput::OutputField< T >

Definition at line 53 of file OutputManager.hpp.

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputManager.hpp

7.33 GlobalFlow::DataProcessing::DataOutput::OutputInterface< T > Class Template Reference

Inheritance diagram for GlobalFlow::DataProcessing::DataOutput::OutputInterface< T >:

Public Member Functions

- virtual void write (path filePath, bool printID, bool printXY, std::vector< T > data, pos_v p)=0

7.33.1 Detailed Description

template<typename T>class GlobalFlow::DataProcessing::DataOutput::OutputInterface< T >

Definition at line 52 of file OutputFactory.hpp.

7.33.2 Member Function Documentation

7.33.2.1 template<typename T> virtual void GlobalFlow::DataProcessing::DataOutput::OutputInterface< T >::write (path *filePath*, bool *printID*, bool *printXY*, std::vector< T > *data*, pos_v *p*) `[pure virtual]`

Needs to be implemented

Parameters

filePath	
printID	Bool
printXY	Bool
data	Data vector
p	Position vector

Implemented in GlobalFlow::DataProcessing::DataOutput::GFS_JSONOutput< T >, and GlobalFlow::DataProcessing::DataOutput::NETCDFOutput< T >.

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputFactory.hpp

7.34 GlobalFlow::DataProcessing::DataOutput::OutputManager Class Reference

Public Member Functions

- **OutputManager** (path output_spec_path, Simulation::Simulation const &sim)
- void write ()

  Visits all registered output options and triggers write.

7.34.1 Detailed Description

Definition at line 170 of file OutputManager.hpp.

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputManager.hpp

7.35 GlobalFlow::DataProcessing::DataOutput::Outputvisitor Class Reference

Inheritance diagram for GlobalFlow::DataProcessing::DataOutput::Outputvisitor:

Collaboration diagram for GlobalFlow::DataProcessing::DataOutput::Outputvisitor:

Public Member Functions

- template<typename T >
  void **operator()** (T &operand) const

7.35.1 Detailed Description

Definition at line 41 of file OutputManager.hpp.

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputManager.hpp

7.36 GlobalFlow::Model::PhysicalProperty< Type, Key > Class Template Reference

```
#include <PhysicalProperties.hpp>
```

Protected Member Functions

- Type & **__get** () const
- Type **__get** ()
- void **__set** (const Type &value)
- template<class... Args>
  void **__emplace** (Args &&...args)

7.36.1 Detailed Description

template< class Type, class Key = DefaultProperty >class GlobalFlow::Model::PhysicalProperty< Type, Key >

Actual container holding the physical propertie

Definition at line 47 of file PhysicalProperties.hpp.

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

- /home/robert/development/G3M-framework/src/Model/PhysicalProperties.hpp

7.37 Position Class Reference

Public Member Functions

- **Position** (double lat, double lon)

Public Attributes

- const double **lat** {0}
- const double **lon** {0}

7.37.1 Detailed Description

Definition at line 83 of file Helpers.hpp.

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

- /home/robert/development/G3M-framework/src/Misc/Helpers.hpp

7.38 PropertiyRepository Class Reference

```
#include <PhysicalProperties.hpp>
```

7.38.1 Detailed Description

Type repository holding the properties

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

- /home/robert/development/G3M-framework/src/Model/PhysicalProperties.hpp

7.39 GlobalFlow::Model::PropertyRepository< PhysicalProperties > Class Template Reference

Inheritance diagram for GlobalFlow::Model::PropertyRepository< PhysicalProperties >:

Collaboration diagram for GlobalFlow::Model::PropertyRepository< PhysicalProperties >:

Public Member Functions

- template<class Type , class Key = DefaultProperty>
  Type & **get** () const
- template<class Type , class Key = DefaultProperty>
  Type **get** ()
- template<class Type , class Key = DefaultProperty>
  void **set** (const Type &value)
- template<class Type , class Key = DefaultProperty, class... Args>
  void **emplace** (Args &&...args)
- template<class Type , class Key = DefaultProperty>
  void **addTo** (const Type &value)

7.39.1 Detailed Description

template<class... PhysicalProperties>class GlobalFlow::Model::PropertyRepository< PhysicalProperties >

Definition at line 88 of file PhysicalProperties.hpp.

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

- /home/robert/development/G3M-framework/src/Model/PhysicalProperties.hpp

7.40 SaticHeadNode Class Reference

```
#include <Node.hpp>
```

7.40.1 Detailed Description

A node without changing head Can be used as boundary condition

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

- /home/robert/development/G3M-framework/src/Model/Node.hpp

7.41 GlobalFlow::Simulation::Simulation Class Reference

```
#include <Simulation.hpp>
```

Public Types

- enum **Flows** {
  RIVERS = 1, **DRAINS**, **RIVER_MM**, **LAKES**,
  WETLANDS, **GLOBAL_WETLANDS**, **RECHARGE**, **FASTSURFACE**,
  NAG, **STORAGE**, **GENERAL_HEAD_BOUNDARY** }

Public Member Functions

- **Simulation** (Options op, DataReader ∗reader)
- Solver::Equation ∗ **getEquation** ()
- void **save** ()
- std::string NodeInfosByID (unsigned long nodeID)

- template<int FieldNum>
  std::string getFlowSumByIDs (std::array< int, FieldNum > ids)
- std::string NodeFlowsByID (unsigned long nodeID)
- template<class FunOut , class FunIn >
  MassError getError (FunOut fun1, FunIn fun2)
- MassError getMassError ()
- MassError getCurrentMassError ()
- double getLossToRivers ()
- template<class Fun >
  MassError getError (Fun fun)
- string getFlowByName (Flows flow)
- void printMassBalances ()
- const double **calcRecharge** (const double recharge, const double &area)
- DataReader ∗ **getDataReader** ()
- NodeVector & **getNodes** ()
- void **scaleByIds** (vector< int > ids, string field, double mult)
- template<class Fun , class ChangeFunction >
  void **scaleByFunction** (Fun fun, ChangeFunction apply)
- template<class Fun >
  void scaleByFunction (Fun fun, string field, double mult)
- void readMultipliers (string path)
- void writeResiduals (string path)
- template<typename DataArray >
  void updateGWRechargeFromWaterGAP (DataArray field, string fieldname, short month, int numberOfGrid↩ Cells)

Public Attributes

- NodeVector **nodes**

7.41.1 Detailed Description

The simulation class which holds the equation, options and data instance Further contains methods for calulating the mass balance and sensitivity methods TODO: Clean me up!

Definition at line 70 of file Simulation.hpp.

7.41.2 Member Function Documentation

7.41.2.1 MassError GlobalFlow::Simulation::Simulation::getCurrentMassError ( ) [inline]

Get the mass balance for the current step

Returns

Definition at line 222 of file Simulation.hpp.

7.41.2.2 template<class FunOut , class FunIn > MassError GlobalFlow::Simulation::Simulation::getError (FunOut *fun1,* FunIn *fun2*) [inline]

Calulate the mass error

Parameters

fun1	Function to get OutFlow
fun2	Function to get InFlow

Returns

Definition at line 190 of file Simulation.hpp.

7.41.2.3 template< class Fun > **MassError** GlobalFlow::Simulation::Simulation::getError (Fun *fun*) [inline]

Decide if its an In or Outflow

Parameters

fun	

Returns

Definition at line 265 of file Simulation.hpp.

7.41.2.4 string GlobalFlow::Simulation::Simulation::getFlowByName (Flows *flow*) [inline]

Helper function for printing the mass balance for each flow

Parameters

flow	

Returns

Definition at line 281 of file Simulation.hpp.

7.41.2.5 template<int FieldNum> std::string GlobalFlow::Simulation::Simulation::getFlowSumByIDs (std::array< int, FieldNum > *ids*) [inline]

Get budget per node

Parameters

ids	

Returns

Definition at line 131 of file Simulation.hpp.

7.41.2.6 double GlobalFlow::Simulation::Simulation::getLossToRivers ( ) [inline]

Get the flow lost to external flows

Returns

Definition at line 235 of file Simulation.hpp.

7.41.2.7 MassError GlobalFlow::Simulation::Simulation::getMassError ( ) [inline]

Get the total mass balance

Returns

Definition at line 209 of file Simulation.hpp.

7.41.2.8 std::string GlobalFlow::Simulation::Simulation::NodeFlowsByID (unsigned long *nodeID*) [inline]

Return all external flows seperatly

Definition at line 150 of file Simulation.hpp.

Here is the call graph for this function:

7.41.2.9 std::string GlobalFlow::Simulation::Simulation::NodeInfosByID (unsigned long *nodeID*) [inline]

Get basic node information by its id

Parameters

nodeID	

Returns

A string of information

Definition at line 88 of file Simulation.hpp.

Here is the call graph for this function:

7.41.2.10 void GlobalFlow::Simulation::Simulation::printMassBalances ( ) [inline]

Prints all mass balances

Definition at line 344 of file Simulation.hpp.

7.41.2.11 void GlobalFlow::Simulation::Simulation::readMultipliers (string *path*) [inline]

Scale values for sensitivity analysis

Parameters

path	

Definition at line 497 of file Simulation.hpp.

7.41.2.12 template<class Fun > void GlobalFlow::Simulation::Simulation::scaleByFunction (Fun *fun,* string *field,* double *mult*) [inline]

Helper function for sensitivity

Parameters

fun	
field	
mult	

Definition at line 415 of file Simulation.hpp.

7.41.2.13 template<typename DataArray > void GlobalFlow::Simulation::Simulation::updateGWRechargeFromWaterGAP (DataArray *field,* string *fieldname,* short *month,* int *numberOfGridCells*) [inline]

Coupling function for WaterGAP

Note

> Under development

Parameters

field	
fieldname	
month	
numberOfGrid←↩ Cells	

Definition at line 582 of file Simulation.hpp.

7.41.2.14 void GlobalFlow::Simulation::Simulation::writeResiduals (string *path*) [inline]

Get the residuals of the current iteration

Parameters

path	

Definition at line 557 of file Simulation.hpp.

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

- /home/robert/development/G3M-framework/src/Simulation/Simulation.hpp
- /home/robert/development/G3M-framework/src/Simulation/Simulation.cpp

7.42 GlobalFlow::Simulation::SimulationOutput Class Reference

Public Member Functions

- **SimulationOutput** (int field_count)
- template<typename Query , class... FieldNames>
void **query** (NodeVector nodes, Query query, FieldNames...names)

Friends

- ostream & **operator**<< (ostream &os, SimulationOutput const &simulation)

7.42.1 Detailed Description

Definition at line 35 of file SimulationOutput.hpp.

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

- /home/robert/development/G3M-framework/src/Misc/SimulationOutput.hpp

7.43 GlobalFlow::Logging::Singleton< C > Class Template Reference

Static Public Member Functions

- static C $*$ **instance** ()

7.43.1 Detailed Description

template< typename C >class GlobalFlow::Logging::Singleton< C >

Definition at line 46 of file Logging.hpp.

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

- /home/robert/development/G3M-framework/src/Logging/Logging.hpp

7.44 GlobalFlow::Model::StandardNode Class Reference

```
#include <Node.hpp>
```

Inheritance diagram for GlobalFlow::Model::StandardNode:

Collaboration diagram for GlobalFlow::Model::StandardNode:

Public Member Functions

- **StandardNode** (std::shared_ptr< std::vector< std::unique_ptr< NodeInterface >>> nodes, double lat, double lon, t_s_meter area, unsigned long ArcID, unsigned long ID, t_vel K, int stepmodifier, double aquiferDepth, double anisotropy, double specificYield, double specificStorage, bool confined)

Friends

- class **NodeInterface**

Additional Inherited Members

7.44.1 Detailed Description

A standard groundwater node

Definition at line 1183 of file Node.hpp.

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

- /home/robert/development/G3M-framework/src/Model/Node.hpp

7.45 GlobalFlow::Model::StaticHeadNode Class Reference

Inheritance diagram for GlobalFlow::Model::StaticHeadNode:

Collaboration diagram for GlobalFlow::Model::StaticHeadNode:

Public Member Functions

- **StaticHeadNode** (std::shared_ptr< std::vector< std::unique_ptr< NodeInterface >>> nodes, unsigned long ID, t_s_meter area)

Friends

- class **NodeInterface**

Additional Inherited Members

7.45.1 Detailed Description

Definition at line 1261 of file Node.hpp.

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

- /home/robert/development/G3M-framework/src/Model/Node.hpp

7.46 GlobalFlow::Simulation::Stepper Class Reference

Inheritance diagram for GlobalFlow::Simulation::Stepper:

Collaboration diagram for GlobalFlow::Simulation::Stepper:

Public Member Functions

- **Stepper** (Solver::Equation ∗eq, const TimeFrame time, const size_t steps)
- virtual Solver::Equation ∗ **get** (int col) const
- Iterator **begin** () const
- Iterator **end** () const
- const TimeFrame **getStepSize** ()

7.46.1 Detailed Description

holding the simulation iterator

Definition at line 76 of file Stepper.hpp.

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

- /home/robert/development/G3M-framework/src/Simulation/Stepper.hpp

7.47 Write Class Reference

7.47.1 Detailed Description

a pseudo GFS json-like file

Note

provides additonal methods for downscaling

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputFactory.hpp

7.48 Writes Class Reference

7.48.1 Detailed Description

a CSV file

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputFactory.hpp

7.49 Writes Class Reference

7.49.1 Detailed Description

a CSV file

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputFactory.hpp

7.50 Writes Interface Reference

7.50.1 Detailed Description

a file

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

- /home/robert/development/G3M-framework/src/DataProcessing/DataOutput/OutputFactory.hpp

Index