

MOHIDLagrangian

0.01

Generated by Doxygen 1.8.14

Contents

1	Modules Index	1
1.1	Modules List	1
2	Data Type Index	3
2.1	Class Hierarchy	3
3	Data Type Index	5
3.1	Data Types List	5
4	File Index	7
4.1	File List	7
5	Module Documentation	9
5.1	commom_modules Module Reference	9
5.1.1	Detailed Description	9
5.2	geometry Module Reference	9
5.2.1	Detailed Description	10
5.3	initialize Module Reference	10
5.3.1	Function/Subroutine Documentation	10
5.3.1.1	init_caseconstants()	10
5.3.1.2	init_parameters()	11
5.3.1.3	init_simdefs()	12
5.3.1.4	init_sources()	13
5.3.1.5	initmohidlagrangian()	14
5.3.1.6	read_xml_geometry()	15

5.3.1.7	readxmlatt()	16
5.3.1.8	readxmlvector()	16
5.4	simulation_globals Module Reference	17
5.4.1	Detailed Description	18
5.4.2	Function/Subroutine Documentation	18
5.4.2.1	setsimbounds()	18
5.4.2.2	setsimdp()	19
5.4.2.3	setsimgravity()	19
5.4.2.4	setsimparameter()	20
5.4.2.5	setsimrho()	20
5.4.3	Variable Documentation	21
5.4.3.1	cfl	21
5.4.3.2	dp	21
5.4.3.3	gravity	21
5.4.3.4	initfreeze	21
5.4.3.5	integrator	21
5.4.3.6	pointmax	22
5.4.3.7	pointmin	22
5.4.3.8	rho_ref	22
5.4.3.9	timemax	22
5.4.3.10	timeout	22
5.5	source Module Reference	22
5.5.1	Detailed Description	22
5.6	source_emitter Module Reference	23
5.6.1	Detailed Description	23
5.7	source_identity Module Reference	23
5.7.1	Detailed Description	23
5.7.2	Function/Subroutine Documentation	24
5.7.2.1	allocsources()	24
5.7.2.2	initsource()	24

5.7.2.3	printsource()	25
5.7.3	Variable Documentation	25
5.7.3.1	source	25
5.8	tracer Module Reference	26
5.8.1	Detailed Description	26
5.9	tracer2d Module Reference	26
5.9.1	Detailed Description	26
5.9.2	Function/Subroutine Documentation	26
5.9.2.1	tracer2d_init()	26
5.10	tracer3d Module Reference	27
5.10.1	Detailed Description	27
5.10.2	Function/Subroutine Documentation	27
5.10.2.1	tracer_init()	27
5.11	tracer_interp Module Reference	28
5.12	tracer_precision Module Reference	28
5.12.1	Detailed Description	28
5.12.2	Variable Documentation	28
5.12.2.1	dp	28
5.12.2.2	err_dist	29
5.12.2.3	err_ind	29
5.12.2.4	missing_value_default	29
5.12.2.5	mv	29
5.12.2.6	mv_int	29
5.12.2.7	prec	29
5.12.2.8	prec_time	29
5.12.2.9	prec_wrt	29
5.12.2.10	sp	29

6	Data Type Documentation	31
6.1	geometry::box Type Reference	31
6.1.1	Detailed Description	32
6.1.2	Member Data Documentation	32
6.1.2.1	size	32
6.2	geometry::line Type Reference	32
6.2.1	Detailed Description	33
6.2.2	Member Data Documentation	33
6.2.2.1	last	33
6.3	geometry::point Type Reference	34
6.3.1	Detailed Description	34
6.4	geometry::shape Type Reference	35
6.4.1	Detailed Description	35
6.4.2	Member Data Documentation	35
6.4.2.1	pt	35
6.5	source_identity::source_class Type Reference	36
6.5.1	Detailed Description	36
6.5.2	Member Data Documentation	36
6.5.2.1	now	36
6.5.2.2	par	37
6.5.2.3	stats	37
6.6	source_identity::source_par_class Type Reference	37
6.6.1	Member Data Documentation	37
6.6.1.1	emitting_rate	37
6.6.1.2	geometry	38
6.6.1.3	id	38
6.6.1.4	name	38
6.6.1.5	property_name	38
6.6.1.6	source_geometry	38
6.7	source_identity::source_state_class Type Reference	38

6.7.1	Detailed Description	39
6.7.2	Member Data Documentation	39
6.7.2.1	active	39
6.7.2.2	age	39
6.7.2.3	depth	39
6.7.2.4	pos	40
6.7.2.5	t	40
6.7.2.6	vel	40
6.8	source_identity::source_stats_class Type Reference	40
6.8.1	Detailed Description	40
6.8.2	Member Data Documentation	40
6.8.2.1	acc_t	41
6.8.2.2	ns	41
6.8.2.3	particles_emitted	41
6.9	geometry::sphere Type Reference	41
6.9.1	Detailed Description	42
6.9.2	Member Data Documentation	42
6.9.2.1	radius	42
6.10	tracer3d::tracer_class Type Reference	43
6.10.1	Detailed Description	43
6.10.2	Member Data Documentation	43
6.10.2.1	now	43
6.10.2.2	par	44
6.10.2.3	stats	44
6.11	tracer3d::tracer_par_class Type Reference	44
6.11.1	Detailed Description	45
6.11.2	Member Data Documentation	45
6.11.2.1	group	45
6.11.2.2	id	45
6.11.2.3	interp_method	45

6.11.2.4	noise	45
6.11.2.5	tpar	45
6.11.2.6	vel_max	46
6.12	tracer3d::tracer_par_trans_class Type Reference	46
6.12.1	Member Data Documentation	46
6.12.1.1	par_trans_file	46
6.12.1.2	use_par_trans	46
6.13	tracer3d::tracer_state_class Type Reference	46
6.13.1	Detailed Description	47
6.13.2	Member Data Documentation	47
6.13.2.1	acc	47
6.13.2.2	active	47
6.13.2.3	age	47
6.13.2.4	depth	48
6.13.2.5	pos	48
6.13.2.6	t	48
6.13.2.7	vel	48
6.14	tracer3d::tracer_stats_class Type Reference	48
6.14.1	Detailed Description	49
6.14.2	Member Data Documentation	49
6.14.2.1	acc_depth	49
6.14.2.2	acc_pos	49
6.14.2.3	acc_t	49
6.14.2.4	acc_vel	49
6.14.2.5	ns	49
7	File Documentation	51
7.1	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/app/main.f90 File Reference	51
7.2	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/common_modules.f90 File Reference	51
7.3	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/geometry.f90 File Reference	51
7.4	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/initialize.f90 File Reference	52
7.5	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/simulation_globals.f90 File Reference	52
7.6	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source.f90 File Reference	53
7.7	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source_emitter.f90 File Reference	53
7.8	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source_identity.f90 File Reference	53
7.9	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer.f90 File Reference	54
7.10	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer2D.f90 File Reference	54
7.11	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer3D.f90 File Reference	55
7.12	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_interp.f90 File Reference	55
7.13	C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_precision.f90 File Reference	55

Chapter 1

Modules Index

1.1 Modules List

Here is a list of all modules with brief descriptions:

common_modules	Module to hold all of the commonly used base modules	9
geometry	Module that defines geometry classes and related methods	9
initialize	10
simulation_globals	Module to finalize the simulation. This presents a public routine that is in charge of deallocating all global variables, closing all files and print some simulation-related statistics	17
source	Module to hold and wrap all the tracer sources respective modules. Defines a source class and related methods	22
source_emitter	Module that defines a source geometry class, emitter class and related methods	23
source_identity	Module that defines a source class and related methods	23
tracer	Module to hold and wrap all the tracer respective modules. Defines a pure Lagrangian tracer class. This is intended to serve as the base class for every type of tracer class needed, that should be built as derived of this class, with the necessary modifiers to model the desired behaviour. Basic tracer data (parameters, variables) are implemented. Tracer methods such as I/O, integration and interpolation routines are implemented	26
tracer2d	Module that defines a pure Lagrangian 2D tracer class and related methods, as a subset of the tracer3D module	26
tracer3d	Module that defines a pure Lagrangian tracer class and related methods	27
tracer_interp	28
tracer_precision	Module to control the precision of the variables trough the project	28

Chapter 2

Data Type Index

2.1 Class Hierarchy

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

geometry::shape	35
geometry::box	31
geometry::line	32
geometry::point	34
geometry::sphere	41
source_identity::source_class	36
source_identity::source_par_class	37
source_identity::source_state_class	38
source_identity::source_stats_class	40
tracer3d::tracer_class	43
tracer3d::tracer_par_class	44
tracer3d::tracer_par_trans_class	46
tracer3d::tracer_state_class	46
tracer3d::tracer_stats_class	48

Chapter 3

Data Type Index

3.1 Data Types List

Here are the data types with brief descriptions:

geometry::box	
Type - point class	31
geometry::line	
Type - line class	32
geometry::point	
Type - point class	34
geometry::shape	
Type - extendable shape class	35
source_identity::source_class	
Type - The source class	36
source_identity::source_par_class	
.	37
source_identity::source_state_class	
Type - state variables of a source object	38
source_identity::source_stats_class	
Type - statistical variables of a source object	40
geometry::sphere	
Type - sphere class	41
tracer3d::tracer_class	
Type - The pure Lagrangian tracer class	43
tracer3d::tracer_par_class	
Type - parameters of a pure Lagrangian tracer object	44
tracer3d::tracer_par_trans_class	
.	46
tracer3d::tracer_state_class	
Type - state variables of a pure Lagrangian tracer object	46
tracer3d::tracer_stats_class	
Type - statistical variables of a pure Lagrangian tracer object	48

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/app/main.f90	51
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/common_modules.f90	51
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/geometry.f90	51
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/initialize.f90	52
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/simulation_globals.f90	52
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source.f90	53
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source_emitter.f90	53
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source_identity.f90	53
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer.f90	54
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer2D.f90	54
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer3D.f90	55
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_interp.f90	55
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_precision.f90	55

Chapter 5

Module Documentation

5.1 commom_modules Module Reference

Module to hold all of the commonly used base modules.

5.1.1 Detailed Description

Module to hold all of the commonly used base modules.

Author

Ricardo Birjukovs Canelas

5.2 geometry Module Reference

Module that defines geometry classes and related methods.

Data Types

- type [box](#)
Type - point class.
- type [line](#)
Type - line class.
- type [point](#)
Type - point class.
- type [shape](#)
Type - extendable shape class.
- type [sphere](#)
Type - sphere class.

5.2.1 Detailed Description

Module that defines geometry classes and related methods.

Author

Ricardo Birjukovs Canelas

5.3 initialize Module Reference

Functions/Subroutines

- subroutine [readxmlatt](#) (xmlnode, tag, att_name, att_value)
Birjukovs Canelas - MARETEC
- subroutine [readxmlvector](#) (xmlnode, tag, vec)
Birjukovs Canelas - MARETEC
- subroutine [read_xml_geometry](#) (source, source_detail, geometry)
Birjukovs Canelas - MARETEC
- subroutine [init_sources](#) (parsedxml)
Birjukovs Canelas - MARETEC
- subroutine [init_simdefs](#) (parsedxml)
Birjukovs Canelas - MARETEC
- subroutine [init_caseconstants](#) (parsedxml)
Birjukovs Canelas - MARETEC
- subroutine [init_parameters](#) (parsedxml)
Birjukovs Canelas - MARETEC
- subroutine, public [initmohidlagrangian](#) (xmlfilename)
Birjukovs Canelas - MARETEC

5.3.1 Function/Subroutine Documentation

5.3.1.1 [init_caseconstants\(\)](#)

```
subroutine initialize::init_caseconstants (
    type(node), intent(in), pointer parsedxml ) [private]
```

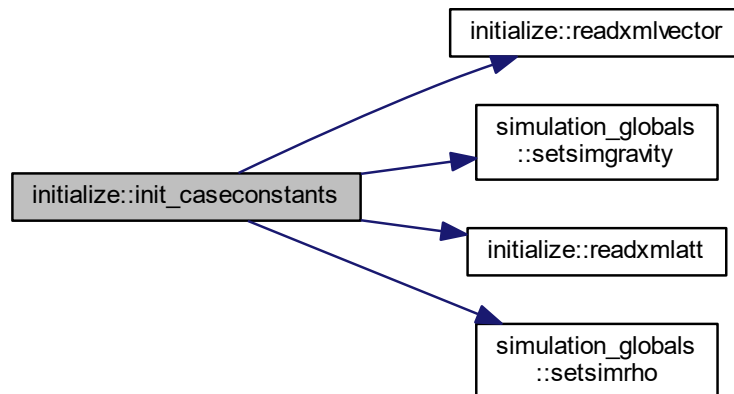
Birjukovs Canelas - MARETEC

Private case constant parser routine. Builds the simulation parametric space from the input xml case file.

Parameters

in	<i>parsedxml</i>	
----	------------------	--

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.1.2 init_parameters()

```

subroutine initialize::init_parameters (
    type(node), intent(in), pointer parsedxml ) [private]

```

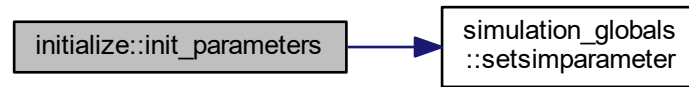
Birjukovs Canelas - MARETEC

Private parameter parser routine. Builds the simulation parametric space from the input xml case file.

Parameters

in	<i>parsedxml</i>	
----	------------------	--

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.1.3 init_simdefs()

```

subroutine initialize::init_simdefs (
    type(node), intent(in), pointer parsedxml ) [private]
  
```

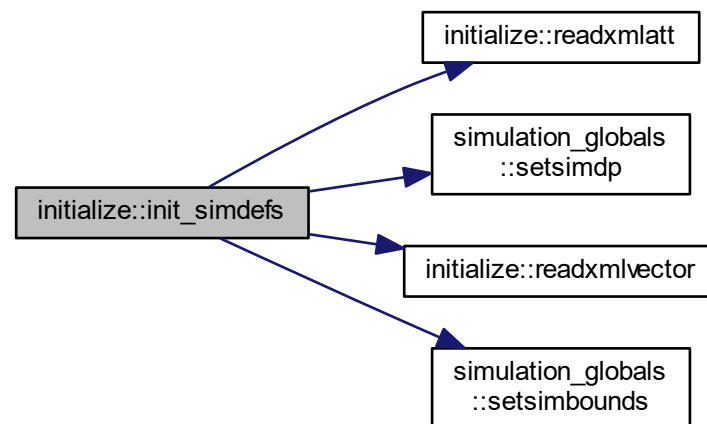
Birjukovs Canelas - MARETEC

Private simulation definitions parser routine. Builds the simulation geometric space from the input xml case file.

Parameters

in	<i>parsedxml</i>	
----	------------------	--

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.1.4 init_sources()

```

subroutine initialize::init_sources (
    type(node), intent(in), pointer parsedxml ) [private]
  
```

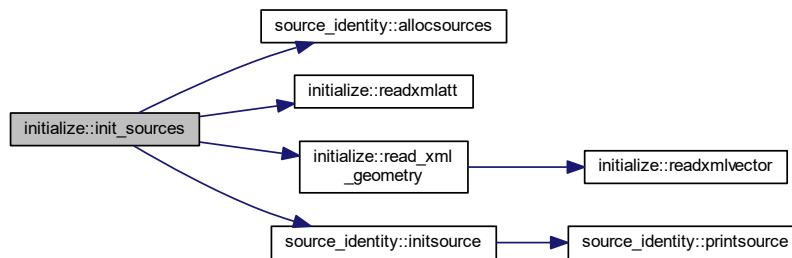
Birjukovs Canelas - MARETEC

Private source definitions parser routine. Builds the tracer sources from the input xml case file.

Parameters

in	<i>parsedxml</i>	
----	------------------	--

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.1.5 initmohidlagrangian()

```

subroutine, public initialize::initmohidlagrangian (
    type(string), intent(in) xmlfilename )
  
```

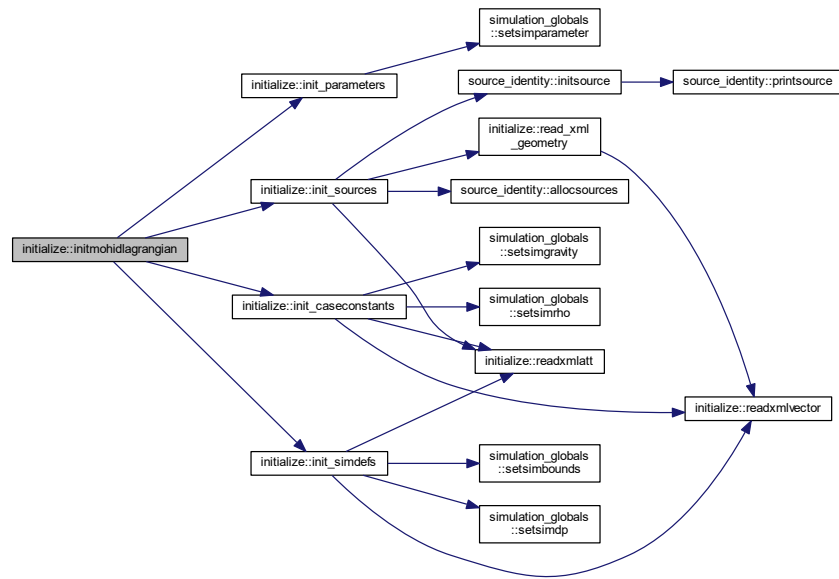
Birjukovs Canelas - MARETEC

Public xml parser routine. Builds the simulation space from the input xml case file.

Parameters

in	<i>xmlfilename</i>	
----	--------------------	--

Here is the call graph for this function:



5.3.1.6 read_xml_geometry()

```

subroutine initialize::read_xml_geometry (
    type(node), intent(in), pointer source,
    type(node), intent(in), pointer source_detail,
    class(shape), intent(inout) geometry ) [private]

```

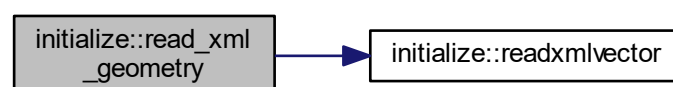
Birjukovs Canelas - MARETEC

Private geometry xml parser routine.

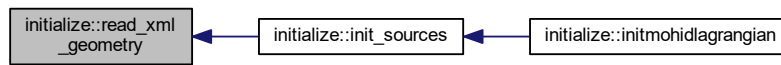
Parameters

in	source, geometry	
----	------------------	--

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.1.7 readxmlatt()

```

subroutine initialize::readxmlatt (
    type(node), intent(in), pointer xmlnode,
    type(string), intent(in) tag,
    type(string), intent(in) att_name,
    type(string), intent(out) att_value ) [private]
  
```

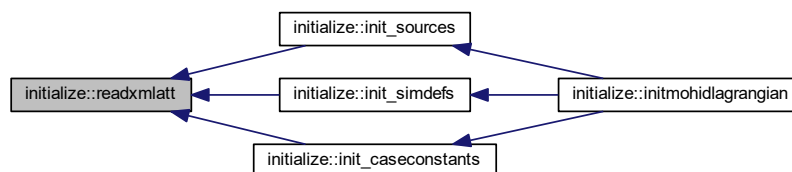
Birjukovs Canelas - MARETEC

Private attribute xml parser routine.

Parameters

in	<i>xmlnode, tag, vec</i>	
----	--------------------------	--

Here is the caller graph for this function:



5.3.1.8 readxmlvector()

```

subroutine initialize::readxmlvector (
    type(node), intent(in), pointer xmlnode,
    type(string), intent(in) tag,
    type(vector), intent(out) vec ) [private]
  
```

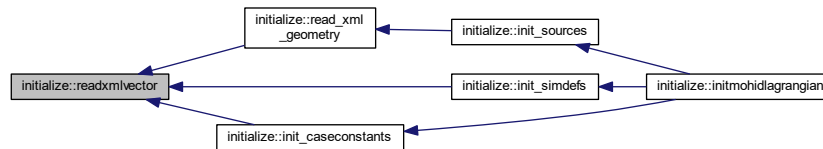
Birjukovs Canelas - MARETEC

Private vector xml parser routine.

Parameters

in	xmlnode,tag,vec	
----	-----------------	--

Here is the caller graph for this function:



5.4 simulation_globals Module Reference

Module to finalize the simulation. This presents a public routine that is in charge of deallocating all global variables, closing all files and print some simulation-related statistics.

Functions/Subroutines

- subroutine, public [setsimparameter](#) (parmkey, parmvalue)
Birjukovs Canelas - MARETEC
- subroutine, public [setsimgravity](#) (grav)
Birjukovs Canelas - MARETEC
- subroutine, public [setsimrho](#) (read_rho)
Birjukovs Canelas - MARETEC
- subroutine, public [setsimdp](#) (read_dp)
Birjukovs Canelas - MARETEC
- subroutine, public [setsimbounds](#) (point_, coords)
Birjukovs Canelas - MARETEC

Variables

- integer, public [integrator](#) = 1
Integration Algorithm 1:Verlet, 2:Symplectic, 3:RK4 (default=1)
- real(prec), public [cfl](#) = 0.5
Courant Friedrichs Lewy condition number.
- real(prec), public [initfreeze](#) = 0.0
Time to freeze the tracers at simulation start (warmup) (default=0.0)
- real(prec), public [timemax](#) = MV
Simulation duration.
- real(prec), public [timeout](#) = MV
Time out data (1/Hz)
- real(prec), public [dp](#) = MV
Initial particle spacing at source generation.

- type(vector), public [pointmin](#)
Point that defines the lowest corner of the simulation bounding box.
- type(vector), public [pointmax](#)
Point that defines the upper corner of the simulation bounding box.
- type(vector), public [gravity](#)
Gravitational acceleration vector (default=(0 0 -9.81)) (m s-2)
- real(prec), public [rho_ref](#) = 1000.0
Reference density of the medium (default=1000.0) (kg m-3)

5.4.1 Detailed Description

Module to finalize the simulation. This presents a public routine that is in charge of deallocating all global variables, closing all files and print some simulation-related statistics.

Author

Ricardo Birjukovs Canelas

5.4.2 Function/Subroutine Documentation

5.4.2.1 setsimbounds()

```
subroutine, public simulation_globals::setsimbounds (
    type(string), intent(in) point_,
    type(vector) coords )
```

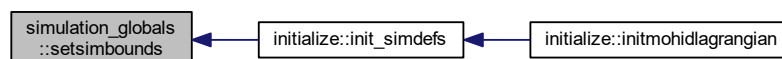
Birjukovs Canelas - MARETEC

Public bounding box setting routine.

Parameters

in	<i>point_,coords</i>	
----	----------------------	--

Here is the caller graph for this function:



5.4.2.2 setsimdp()

```
subroutine, public simulation_globals::setsimdp (  
    type(string), intent(in) read_dp )
```

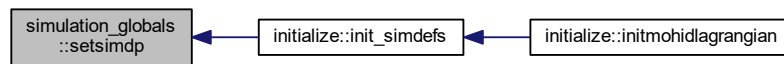
Birjukovs Canelas - MARETEC

Public dp setting routine.

Parameters

in	<i>read_dp</i>	
----	----------------	--

Here is the caller graph for this function:



5.4.2.3 setsimgravity()

```
subroutine, public simulation_globals::setsimgravity (  
    type(vector) grav )
```

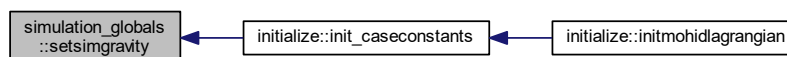
Birjukovs Canelas - MARETEC

Public Gravity setting routine.

Parameters

in	<i>grav</i>	
----	-------------	--

Here is the caller graph for this function:



5.4.2.4 setsimparameter()

```
subroutine, public simulation_globals::setsimparameter (
    type(string), intent(in) parmkey,
    type(string), intent(in) parmvalue )
```

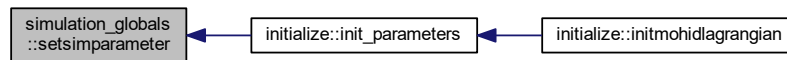
Birjukovs Canelas - MARETEC

Public parameter setting routine. Builds the simulation parametric space from the input xml case file.

Parameters

in	<i>parmkey</i>	
in	<i>parmvalue</i>	

Here is the caller graph for this function:



5.4.2.5 setsimrho()

```
subroutine, public simulation_globals::setsimrho (
    type(string), intent(in) read_rho )
```

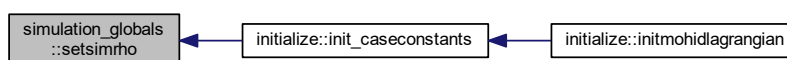
Birjukovs Canelas - MARETEC

Public Rho_Ref setting routine.

Parameters

in	<i>rho</i>	
----	------------	--

Here is the caller graph for this function:



5.4.3 Variable Documentation

5.4.3.1 cfl

```
real(prec), public simulation_globals::cfl = 0.5
```

Courant Friedrichs Lewy condition number.

5.4.3.2 dp

```
real(prec), public simulation_globals::dp = MV
```

Initial particle spacing at source generation.

5.4.3.3 gravity

```
type(vector), public simulation_globals::gravity
```

Gravitational acceleration vector (default=(0 0 -9.81)) (m s-2)

5.4.3.4 initfreeze

```
real(prec), public simulation_globals::initfreeze = 0.0
```

Time to freeze the tracers at simulation start (warmup) (default=0.0)

5.4.3.5 integrator

```
integer, public simulation_globals::integrator = 1
```

Integration Algorithm 1:Verlet, 2:Symplectic, 3:RK4 (default=1)

5.4.3.6 pointmax

```
type(vector), public simulation_globals::pointmax
```

Point that defines the upper corner of the simulation bounding box.

5.4.3.7 pointmin

```
type(vector), public simulation_globals::pointmin
```

Point that defines the lowest corner of the simulation bounding box.

5.4.3.8 rho_ref

```
real(prec), public simulation_globals::rho_ref = 1000.0
```

Reference density of the medium (default=1000.0) (kg m-3)

5.4.3.9 timemax

```
real(prec), public simulation_globals::timemax = MV
```

Simulation duration.

5.4.3.10 timeout

```
real(prec), public simulation_globals::timeout = MV
```

Time out data (1/Hz)

5.5 source Module Reference

Module to hold and wrap all the tracer sources respective modules. Defines a source class and related methods.

5.5.1 Detailed Description

Module to hold and wrap all the tracer sources respective modules. Defines a source class and related methods.

Author

Ricardo Birjukovs Canelas

5.6 `source_emitter` Module Reference

Module that defines a source geometry class, emitter class and related methods.

5.6.1 Detailed Description

Module that defines a source geometry class, emitter class and related methods.

Author

Ricardo Birjukovs Canelas

5.7 `source_identity` Module Reference

Module that defines a source class and related methods.

Data Types

- type `source_class`
Type - The source class.
- type `source_par_class`
- type `source_state_class`
Type - state variables of a source object.
- type `source_stats_class`
Type - statistical variables of a source object.

Functions/Subroutines

- subroutine, public `allocsources` (nsources)
Birjukovs Canelas - MARETEC
- subroutine, public `initsource` (num, id, name, emitting_rate, source_geometry, geometry)
Birjukovs Canelas - MARETEC
- subroutine `printsource` (src)
Birjukovs Canelas - MARETEC

Variables

- type(`source_class`), `dimension(:)`, allocatable, public `source`

5.7.1 Detailed Description

Module that defines a source class and related methods.

Author

Ricardo Birjukovs Canelas

5.7.2 Function/Subroutine Documentation

5.7.2.1 allocsources()

```
subroutine, public source_identity::allocsources (
    integer, intent(in) nsources )
```

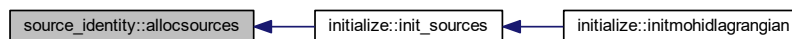
Birjukovs Canelas - MARETEC

source allocation routine - allocates the sources objects

Parameters

in	<i>nsources</i>	
----	-----------------	--

Here is the caller graph for this function:



5.7.2.2 initsource()

```
subroutine, public source_identity::initsource (
    integer, intent(in) num,
    integer, intent(in) id,
    type(string), intent(in) name,
    real(prec), intent(in) emitting_rate,
    type(string), intent(in) source_geometry,
    class(shape), intent(in) geometry )
```

Birjukovs Canelas - MARETEC

source initialization routine - Generates a source and initializes its variables

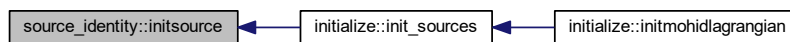
Parameters

out	<i>source</i>	
in	<i>num</i> , <i>id</i> , <i>name</i> , <i>emitting_rate</i> , <i>source_geometry</i>	

Here is the call graph for this function:



Here is the caller graph for this function:



5.7.2.3 printsource()

```

subroutine source_identity::printsource (
    type(source_class), intent(in) src ) [private]
  
```

Birjukovs Canelas - MARETEC

source print routine - prints a source info on console/log

Parameters

in	src	
----	-----	--

Here is the caller graph for this function:



5.7.3 Variable Documentation

5.7.3.1 source

```

type(source_class), dimension(:), allocatable, public source_identity::source
  
```

5.8 tracer Module Reference

Module to hold and wrap all the tracer respective modules. Defines a pure Lagrangian tracer class. This is intended to serve as the base class for every type of tracer class needed, that should be built as derived of this class, with the necessary modifiers to model the desired behaviour. Basic tracer data (parameters, variables) are implemented. Tracer methods such as I/O, integration and interpolation routines are implemented.

5.8.1 Detailed Description

Module to hold and wrap all the tracer respective modules. Defines a pure Lagrangian tracer class. This is intended to serve as the base class for every type of tracer class needed, that should be built as derived of this class, with the necessary modifiers to model the desired behaviour. Basic tracer data (parameters, variables) are implemented. Tracer methods such as I/O, integration and interpolation routines are implemented.

Author

Ricardo Birjukovs Canelas

5.9 tracer2d Module Reference

Module that defines a pure Lagrangian 2D tracer class and related methods, as a subset of the tracer3D module.

Functions/Subroutines

- subroutine [tracer2d_init](#) (trc, filename, time, x, is_sigma)
Birjukovs Canelas - MARETEC Routine Author Name and Affiliation.

5.9.1 Detailed Description

Module that defines a pure Lagrangian 2D tracer class and related methods, as a subset of the tracer3D module.

Author

Ricardo Birjukovs Canelas

5.9.2 Function/Subroutine Documentation

5.9.2.1 tracer2d_init()

```
subroutine tracer2d::tracer2d_init (
    type(tracer_class), intent(out) trc,
    character(len=*), intent(in) filename,
    real(prec_time) time,
    real(prec), dimension(:), intent(in) x,
    logical, intent(in) is_sigma )
```

Birjukovs Canelas - MARETEC Routine Author Name and Affiliation.

Brief description of routine.

2D Tracer initialization routine - Generates a tracer collection and initializes their variables

Parameters

out	trc	
in	filename	

5.10 tracer3d Module Reference

Module that defines a pure Lagrangian tracer class and related methods.

Data Types

- type [tracer_class](#)
Type - The pure Lagrangian tracer class.
- type [tracer_par_class](#)
Type - parameters of a pure Lagrangian tracer object.
- type [tracer_par_trans_class](#)
- type [tracer_state_class](#)
Type - state variables of a pure Lagrangian tracer object.
- type [tracer_stats_class](#)
Type - statistical variables of a pure Lagrangian tracer object.

Functions/Subroutines

- subroutine, public [tracer_init](#) (trc, id, time, x, y, z)
Birjukovs Canelas - MARETEC

5.10.1 Detailed Description

Module that defines a pure Lagrangian tracer class and related methods.

Author

Ricardo Birjukovs Canelas

5.10.2 Function/Subroutine Documentation

5.10.2.1 tracer_init()

```
subroutine, public tracer3d::tracer_init (
    type(tracer\_class), intent(inout) trc,
    integer, intent(in) id,
    real(prec_time), intent(in) time,
    real(prec), intent(in) x,
    real(prec), intent(in) y,
    real(prec), intent(in) z )
```

Birjukovs Canelas - MARETEC

Tracer initialization routine - Generates a tracer and initializes its variables

Parameters

out	trc	
in	filename	

5.11 tracer_interp Module Reference**5.12 tracer_precision Module Reference**

Module to control the precision of the variables trough the project.

Variables

- integer, parameter `sp` = kind(1._R4P)
Simple precision definition switch.
- integer, parameter `dp` = kind(1._R8P)
Double precision definition switch.
- integer, parameter, public `prec` = `sp`
- integer, parameter, public `prec_time` = `sp`
- integer, parameter, public `prec_wrt` = `sp`
- real(`prec`), parameter, public `missing_value_default` = -9999.0_dp
- real(`prec`), parameter, public `mv` = MISSING_VALUE_DEFAULT
- real(`prec`), parameter, public `mv_int` = int(MISSING_VALUE_DEFAULT)
- real(`prec`), parameter, public `err_dist` = 1E8_dp
- integer, parameter, public `err_ind` = -1

5.12.1 Detailed Description

Module to control the precision of the variables trough the project.

Author

Ricardo Birjukovs Canelas

5.12.2 Variable Documentation**5.12.2.1 dp**

```
integer, parameter tracer_precision::dp = kind(1._R8P) [private]
```

Double precision definition switch.

5.12.2.2 err_dist

```
real(prec), parameter, public tracer_precision::err_dist = 1E8_dp
```

5.12.2.3 err_ind

```
integer, parameter, public tracer_precision::err_ind = -1
```

5.12.2.4 missing_value_default

```
real(prec), parameter, public tracer_precision::missing_value_default = -9999.0_dp
```

5.12.2.5 mv

```
real(prec), parameter, public tracer_precision::mv = MISSING_VALUE_DEFAULT
```

5.12.2.6 mv_int

```
real(prec), parameter, public tracer_precision::mv_int = int(MISSING_VALUE_DEFAULT)
```

5.12.2.7 prec

```
integer, parameter, public tracer_precision::prec = sp
```

5.12.2.8 prec_time

```
integer, parameter, public tracer_precision::prec_time = sp
```

5.12.2.9 prec_wrt

```
integer, parameter, public tracer_precision::prec_wrt = sp
```

5.12.2.10 sp

```
integer, parameter tracer_precision::sp = kind(1._R4P) [private]
```

Simple precision definition switch.

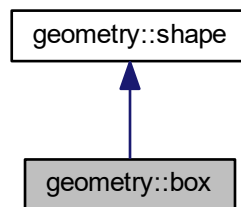
Chapter 6

Data Type Documentation

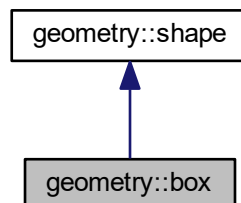
6.1 geometry::box Type Reference

Type - point class.

Inheritance diagram for geometry::box:



Collaboration diagram for geometry::box:



Public Attributes

- type(vector) [size](#)
Box size.

6.1.1 Detailed Description

Type - point class.

6.1.2 Member Data Documentation

6.1.2.1 size

```
type(vector) geometry::box::size
```

Box size.

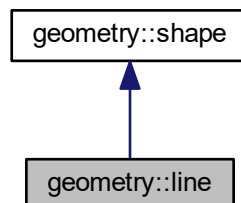
The documentation for this type was generated from the following file:

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/[geometry.f90](#)

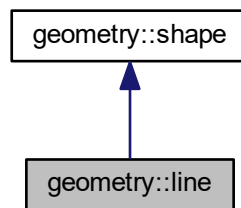
6.2 geometry::line Type Reference

Type - line class.

Inheritance diagram for geometry::line:



Collaboration diagram for geometry::line:



Public Attributes

- type(vector) [last](#)
Coordinates of the end point.

6.2.1 Detailed Description

Type - line class.

6.2.2 Member Data Documentation

6.2.2.1 last

```
type(vector) geometry::line::last
```

Coordinates of the end point.

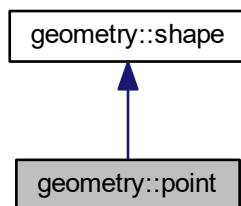
The documentation for this type was generated from the following file:

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/[geometry.f90](#)

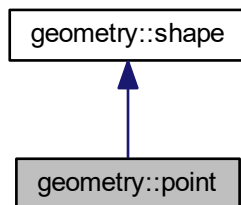
6.3 geometry::point Type Reference

Type - point class.

Inheritance diagram for geometry::point:



Collaboration diagram for geometry::point:



Additional Inherited Members

6.3.1 Detailed Description

Type - point class.

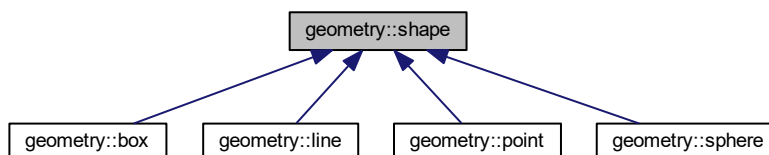
The documentation for this type was generated from the following file:

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/[geometry.f90](#)

6.4 geometry::shape Type Reference

Type - extendable shape class.

Inheritance diagram for geometry::shape:



Public Attributes

- `type(vector)` `pt`
Coordinates of a point.

6.4.1 Detailed Description

Type - extendable shape class.

6.4.2 Member Data Documentation

6.4.2.1 pt

`type(vector)` `geometry::shape::pt`

Coordinates of a point.

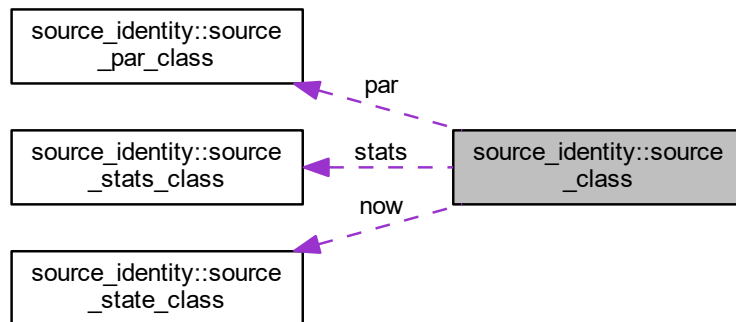
The documentation for this type was generated from the following file:

- `C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/geometry.f90`

6.5 source_identity::source_class Type Reference

Type - The source class.

Collaboration diagram for source_identity::source_class:



Private Attributes

- `type(source_par_class) par`
To access parameters.
- `type(source_state_class) now`
To access state variables.
- `type(source_stats_class) stats`
To access statistics.

6.5.1 Detailed Description

Type - The source class.

6.5.2 Member Data Documentation

6.5.2.1 now

```
type(source_state_class) source_identity::source_class::now [private]
```

To access state variables.

6.5.2.2 par

```
type(source_par_class) source_identity::source_class::par [private]
```

To access parameters.

6.5.2.3 stats

```
type(source_stats_class) source_identity::source_class::stats [private]
```

To access statistics.

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/[source_identity.f90](#)

6.6 source_identity::source_par_class Type Reference

Private Attributes

- integer [id](#)
unique source identification (integer)
- real(prec) [emitting_rate](#)
Emitting rate of the source (Hz)
- type(string) [name](#)
source name
- type(string) [property_name](#)
source property name
- type(string) [source_geometry](#)
Source type : 'point', 'line', 'sphere', 'box'.
- class([shape](#)), allocatable [geometry](#)
Source geometry.

6.6.1 Member Data Documentation

6.6.1.1 emitting_rate

```
real(prec) source_identity::source_par_class::emitting_rate [private]
```

Emitting rate of the source (Hz)

6.6.1.2 geometry

```
class(shape), allocatable source_identity::source_par_class::geometry [private]
```

Source geometry.

6.6.1.3 id

```
integer source_identity::source_par_class::id [private]
```

unique source identification (integer)

6.6.1.4 name

```
type(string) source_identity::source_par_class::name [private]
```

source name

6.6.1.5 property_name

```
type(string) source_identity::source_par_class::property_name [private]
```

source property name

6.6.1.6 source_geometry

```
type(string) source_identity::source_par_class::source_geometry [private]
```

Source type : 'point', 'line', 'sphere', 'box'.

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/[source_identity.f90](#)

6.7 source_identity::source_state_class Type Reference

Type - state variables of a source object.

Private Attributes

- real(prec_time) [age](#)
- logical [active](#)
active switch
- type(vector) [pos](#)
Position of the source baricenter (m)
- type(vector) [vel](#)
Velocity of the source (m s-1)
- real(prec) [depth](#)
Depth of the source baricenter (m)
- real(prec) [t](#)
Temperature of the source (Celcius)

6.7.1 Detailed Description

Type - state variables of a source object.

6.7.2 Member Data Documentation

6.7.2.1 active

```
logical source_identity::source_state_class::active [private]
```

active switch

6.7.2.2 age

```
real(prec_time) source_identity::source_state_class::age [private]
```

6.7.2.3 depth

```
real(prec) source_identity::source_state_class::depth [private]
```

Depth of the source baricenter (m)

6.7.2.4 pos

```
type(vector) source_identity::source_state_class::pos [private]
```

Position of the source baricenter (m)

6.7.2.5 t

```
real(prec) source_identity::source_state_class::t [private]
```

Temperature of the source (Celcius)

6.7.2.6 vel

```
type(vector) source_identity::source_state_class::vel [private]
```

Velocity of the source (m s-1)

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/[source_identity.f90](#)

6.8 source_identity::source_stats_class Type Reference

Type - statistical variables of a source object.

Private Attributes

- integer [particles_emitted](#)
Number of emitted particles by this source.
- real(prec_wrt) [acc_t](#)
Accumulated temperature of the tracer (Celcius)
- integer [ns](#)
Number of sampling steps.

6.8.1 Detailed Description

Type - statistical variables of a source object.

6.8.2 Member Data Documentation

6.8.2.1 acc_t

```
real(prec_wrt) source_identity::source_stats_class::acc_t [private]
```

Accumulated temperature of the tracer (Celcius)

6.8.2.2 ns

```
integer source_identity::source_stats_class::ns [private]
```

Number of sampling steps.

6.8.2.3 particles_emitted

```
integer source_identity::source_stats_class::particles_emitted [private]
```

Number of emitted particles by this source.

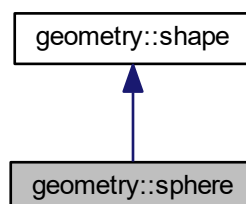
The documentation for this type was generated from the following file:

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/[source_identity.f90](#)

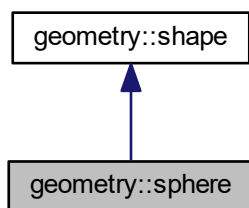
6.9 geometry::sphere Type Reference

Type - sphere class.

Inheritance diagram for geometry::sphere:



Collaboration diagram for geometry::sphere:



Public Attributes

- real(prec) [radius](#)
Sphere radius.

6.9.1 Detailed Description

Type - sphere class.

6.9.2 Member Data Documentation

6.9.2.1 radius

```
real(prec) geometry::sphere::radius
```

Sphere radius.

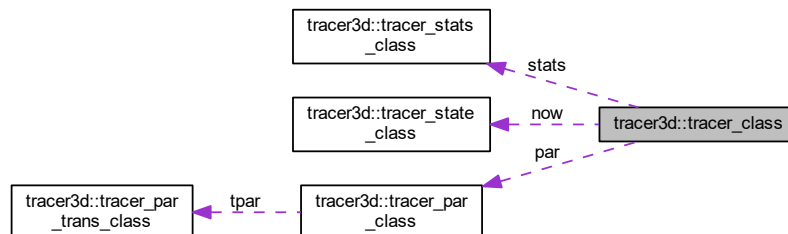
The documentation for this type was generated from the following file:

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/[geometry.f90](#)

6.10 tracer3d::tracer_class Type Reference

Type - The pure Lagrangian tracer class.

Collaboration diagram for tracer3d::tracer_class:



Private Attributes

- `type(tracer_par_class) par`
To access parameters.
- `type(tracer_state_class) now`
To access state variables.
- `type(tracer_stats_class) stats`
To access statistics.

6.10.1 Detailed Description

Type - The pure Lagrangian tracer class.

6.10.2 Member Data Documentation

6.10.2.1 now

```
type(tracer_state_class) tracer3d::tracer_class::now [private]
```

To access state variables.

6.10.2.2 par

```
type(tracer\_par\_class) tracer3d::tracer_class::par [private]
```

To access parameters.

6.10.2.3 stats

```
type(tracer\_stats\_class) tracer3d::tracer_class::stats [private]
```

To access statistics.

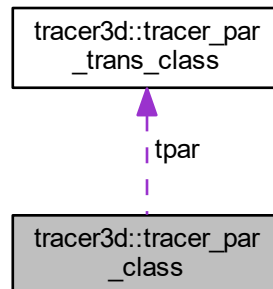
The documentation for this type was generated from the following file:

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/[tracer3D.f90](#)

6.11 tracer3d::tracer_par_class Type Reference

Type - parameters of a pure Lagrangian tracer object.

Collaboration diagram for tracer3d::tracer_par_class:



Private Attributes

- integer [id](#)
unique tracer identification (integer)
- integer [group](#)
Group to which the tracer belongs (usually by source)
- real(prec) [vel_max](#)
Maximum velocity of tracer to track (m/s)
- logical [noise](#)
- character(len=56) [interp_method](#)
interpolation method this tracer calls
- type([tracer_par_trans_class](#)) [tpar](#)
access to the transient parameters is done through this

6.11.1 Detailed Description

Type - parameters of a pure Lagrangian tracer object.

6.11.2 Member Data Documentation

6.11.2.1 group

```
integer tracer3d::tracer_par_class::group [private]
```

Group to which the tracer belongs (usually by source)

6.11.2.2 id

```
integer tracer3d::tracer_par_class::id [private]
```

unique tracer identification (integer)

6.11.2.3 interp_method

```
character(len=56) tracer3d::tracer_par_class::interp_method [private]
```

interpolation method this tracer calls

6.11.2.4 noise

```
logical tracer3d::tracer_par_class::noise [private]
```

6.11.2.5 tpar

```
type(tracer_par_trans_class) tracer3d::tracer_par_class::tpar [private]
```

access to the transient parameters is done through this

6.11.2.6 vel_max

```
real(prec) tracer3d::tracer_par_class::vel_max [private]
```

Maximum velocity of tracer to track (m/s)

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/[tracer3D.f90](#)

6.12 tracer3d::tracer_par_trans_class Type Reference

Private Attributes

- character(len=512) [par_trans_file](#)
- logical [use_par_trans](#)

6.12.1 Member Data Documentation

6.12.1.1 par_trans_file

```
character(len=512) tracer3d::tracer_par_trans_class::par_trans_file [private]
```

6.12.1.2 use_par_trans

```
logical tracer3d::tracer_par_trans_class::use_par_trans [private]
```

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/[tracer3D.f90](#)

6.13 tracer3d::tracer_state_class Type Reference

Type - state variables of a pure Lagrangian tracer object.

Private Attributes

- real(prec_time) [age](#)
- logical [active](#)
active switch
- type(vector) [pos](#)
Position of the tracer (m)
- type(vector) [vel](#)
Velocity of the tracer (m s-1)
- type(vector) [acc](#)
Acceleration of the tracer (m s-2)
- real(prec) [depth](#)
Depth of the tracer (m)
- real(prec) [t](#)
Temperature of the tracer (Celcius)

6.13.1 Detailed Description

Type - state variables of a pure Lagrangian tracer object.

6.13.2 Member Data Documentation

6.13.2.1 [acc](#)

```
type(vector) tracer3d::tracer_state_class::acc [private]
```

Acceleration of the tracer (m s-2)

6.13.2.2 [active](#)

```
logical tracer3d::tracer_state_class::active [private]
```

active switch

6.13.2.3 [age](#)

```
real(prec_time) tracer3d::tracer_state_class::age [private]
```

6.13.2.4 depth

```
real(prec) tracer3d::tracer_state_class::depth [private]
```

Depth of the tracer (m)

6.13.2.5 pos

```
type(vector) tracer3d::tracer_state_class::pos [private]
```

Position of the tracer (m)

6.13.2.6 t

```
real(prec) tracer3d::tracer_state_class::t [private]
```

Temperature of the tracer (Celcius)

6.13.2.7 vel

```
type(vector) tracer3d::tracer_state_class::vel [private]
```

Velocity of the tracer (m s-1)

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/[tracer3D.f90](#)

6.14 tracer3d::tracer_stats_class Type Reference

Type - statistical variables of a pure Lagrangian tracer object.

Private Attributes

- type(vector) [acc_pos](#)
Accumulated position of the tracer (m)
- type(vector) [acc_vel](#)
Accumulated velocity of the tracer (m s-1)
- real(prec_wrt) [acc_depth](#)
Accumulated depth of the tracer (m)
- real(prec_wrt) [acc_t](#)
Accumulated temperature of the tracer (Celcius)
- integer [ns](#)
Number of sampling steps.

6.14.1 Detailed Description

Type - statistical variables of a pure Lagrangian tracer object.

6.14.2 Member Data Documentation

6.14.2.1 acc_depth

```
real(prec_wrt) tracer3d::tracer_stats_class::acc_depth [private]
```

Accumulated depth of the tracer (m)

6.14.2.2 acc_pos

```
type(vector) tracer3d::tracer_stats_class::acc_pos [private]
```

Accumulated position of the tracer (m)

6.14.2.3 acc_t

```
real(prec_wrt) tracer3d::tracer_stats_class::acc_t [private]
```

Accumulated temperature of the tracer (Celcius)

6.14.2.4 acc_vel

```
type(vector) tracer3d::tracer_stats_class::acc_vel [private]
```

Accumulated velocity of the tracer (m s-1)

6.14.2.5 ns

```
integer tracer3d::tracer_stats_class::ns [private]
```

Number of sampling steps.

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/[tracer3D.f90](#)

Chapter 7

File Documentation

7.1 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/app/main.f90 File Reference

7.2 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/common_↵modules.f90 File Reference

Modules

- module [common_modules](#)
Module to hold all of the commonly used base modules.

7.3 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/geometry.f90 File Reference

Data Types

- type [geometry::shape](#)
Type - extendable shape class.
- type [geometry::point](#)
Type - point class.
- type [geometry::line](#)
Type - line class.
- type [geometry::sphere](#)
Type - sphere class.
- type [geometry::box](#)
Type - point class.

Modules

- module [geometry](#)
Module that defines geometry classes and related methods.

7.4 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/initialize.f90 File Reference

Modules

- module [initialize](#)

Functions/Subroutines

- subroutine [initialize::readxmlatt](#) (xmlnode, tag, att_name, att_value)
Birjukovs Canelas - MARETEC
- subroutine [initialize::readxmlvector](#) (xmlnode, tag, vec)
Birjukovs Canelas - MARETEC
- subroutine [initialize::read_xml_geometry](#) (source, source_detail, geometry)
Birjukovs Canelas - MARETEC
- subroutine [initialize::init_sources](#) (parsedxml)
Birjukovs Canelas - MARETEC
- subroutine [initialize::init_simdefs](#) (parsedxml)
Birjukovs Canelas - MARETEC
- subroutine [initialize::init_caseconstants](#) (parsedxml)
Birjukovs Canelas - MARETEC
- subroutine [initialize::init_parameters](#) (parsedxml)
Birjukovs Canelas - MARETEC
- subroutine, public [initialize::initmohidlagrangian](#) (xmlfilename)
Birjukovs Canelas - MARETEC

7.5 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/simulation_globals.f90 File Reference

Modules

- module [simulation_globals](#)
Module to finalize the simulation. This presents a public routine that is in charge of deallocating all global variables, closing all files and print some simulation-related statistics.

Functions/Subroutines

- subroutine, public [simulation_globals::setsimparameter](#) (parmkey, parmvalue)
Birjukovs Canelas - MARETEC
- subroutine, public [simulation_globals::setsimgravity](#) (grav)
Birjukovs Canelas - MARETEC
- subroutine, public [simulation_globals::setsimrho](#) (read_rho)
Birjukovs Canelas - MARETEC
- subroutine, public [simulation_globals::setsimdp](#) (read_dp)
Birjukovs Canelas - MARETEC
- subroutine, public [simulation_globals::setsimbounds](#) (point_, coords)
Birjukovs Canelas - MARETEC

Variables

- integer, public `simulation_globals::integrator` = 1
Integration Algorithm 1:Verlet, 2:Symplectic, 3:RK4 (default=1)
- real(prec), public `simulation_globals::cfl` = 0.5
Courant Friedrichs Lewy condition number.
- real(prec), public `simulation_globals::initfreeze` = 0.0
Time to freeze the tracers at simulation start (warmup) (default=0.0)
- real(prec), public `simulation_globals::timemax` = MV
Simulation duration.
- real(prec), public `simulation_globals::timeout` = MV
Time out data (1/Hz)
- real(prec), public `simulation_globals::dp` = MV
Initial particle spacing at source generation.
- type(vector), public `simulation_globals::pointmin`
Point that defines the lowest corner of the simulation bounding box.
- type(vector), public `simulation_globals::pointmax`
Point that defines the upper corner of the simulation bounding box.
- type(vector), public `simulation_globals::gravity`
Gravitational acceleration vector (default=(0 0 -9.81)) (m s⁻²)
- real(prec), public `simulation_globals::rho_ref` = 1000.0
Reference density of the medium (default=1000.0) (kg m⁻³)

7.6 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source.f90 File Reference

Modules

- module `source`
Module to hold and wrap all the tracer sources respective modules. Defines a source class and related methods.

7.7 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source_↵ emitter.f90 File Reference

Modules

- module `source_emitter`
Module that defines a source geometry class, emitter class and related methods.

7.8 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source_↵ identity.f90 File Reference

Data Types

- type `source_identity::source_par_class`
- type `source_identity::source_state_class`
Type - state variables of a source object.
- type `source_identity::source_stats_class`
Type - statistical variables of a source object.
- type `source_identity::source_class`
Type - The source class.

Modules

- module [source_identity](#)
Module that defines a source class and related methods.

Functions/Subroutines

- subroutine, public [source_identity::allocsources](#) (nsources)
Birjukovs Canelas - MARETEC
- subroutine, public [source_identity::initsource](#) (num, id, name, emitting_rate, source_geometry, geometry)
Birjukovs Canelas - MARETEC
- subroutine [source_identity::printsources](#) (src)
Birjukovs Canelas - MARETEC

Variables

- type(source_class), dimension(:), allocatable, public [source_identity::source](#)

7.9 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer.f90 File Reference

Modules

- module [tracer](#)
Module to hold and wrap all the tracer respective modules. Defines a pure Lagrangian tracer class. This is intended to serve as the base class for every type of tracer class needed, that should be built as derived of this class, with the necessary modifiers to model the desired behaviour. Basic tracer data (parameters, variables) are implemented. Tracer methods such as I/O, integration and interpolation routines are implemented.

7.10 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer2D.f90 File Reference

Modules

- module [tracer2d](#)
Module that defines a pure Lagrangian 2D tracer class and related methods, as a subset of the tracer3D module.

Functions/Subroutines

- subroutine [tracer2d::tracer2d_init](#) (trc, filename, time, x, is_sigma)
Birjukovs Canelas - MARETEC Routine Author Name and Affiliation.

7.11 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer3D.f90 File Reference

Data Types

- type [tracer3d::tracer_par_trans_class](#)
- type [tracer3d::tracer_par_class](#)
Type - parameters of a pure Lagrangian tracer object.
- type [tracer3d::tracer_state_class](#)
Type - state variables of a pure Lagrangian tracer object.
- type [tracer3d::tracer_stats_class](#)
Type - statistical variables of a pure Lagrangian tracer object.
- type [tracer3d::tracer_class](#)
Type - The pure Lagrangian tracer class.

Modules

- module [tracer3d](#)
Module that defines a pure Lagrangian tracer class and related methods.

Functions/Subroutines

- subroutine, public [tracer3d::tracer_init](#) (trc, id, time, x, y, z)
Birjukovs Canelas - MARETEC

7.12 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_↔interp.f90 File Reference

Modules

- module [tracer_interp](#)

7.13 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_↔precision.f90 File Reference

Modules

- module [tracer_precision](#)
Module to control the precision of the variables trough the project.

Variables

- integer, parameter `tracer_precision::sp` = kind(1._R4P)
Simple precision definition switch.
- integer, parameter `tracer_precision::dp` = kind(1._R8P)
Double precision definition switch.
- integer, parameter, public `tracer_precision::prec` = sp
- integer, parameter, public `tracer_precision::prec_time` = sp
- integer, parameter, public `tracer_precision::prec_wrt` = sp
- real(prec), parameter, public `tracer_precision::missing_value_default` = -9999.0_dp
- real(prec), parameter, public `tracer_precision::mv` = MISSING_VALUE_DEFAULT
- real(prec), parameter, public `tracer_precision::mv_int` = int(MISSING_VALUE_DEFAULT)
- real(prec), parameter, public `tracer_precision::err_dist` = 1E8_dp
- integer, parameter, public `tracer_precision::err_ind` = -1