

MOHIDLagrangian

0.01

Generated by Doxygen 1.8.14



# Contents

<b>1</b>	<b>Modules Index</b>	<b>1</b>
1.1	Modules List . . . . .	1
<b>2</b>	<b>Data Type Index</b>	<b>3</b>
2.1	Class Hierarchy . . . . .	3
<b>3</b>	<b>Data Type Index</b>	<b>5</b>
3.1	Data Types List . . . . .	5
<b>4</b>	<b>File Index</b>	<b>7</b>
4.1	File List . . . . .	7
<b>5</b>	<b>Module Documentation</b>	<b>9</b>
5.1	commom_modules Module Reference . . . . .	9
5.1.1	Detailed Description . . . . .	9
5.2	geometry Module Reference . . . . .	9
5.2.1	Detailed Description . . . . .	9
5.3	initialize Module Reference . . . . .	10
5.3.1	Function/Subroutine Documentation . . . . .	10
5.3.1.1	initmohidlagrangian() . . . . .	10
5.4	simulation_parameters Module Reference . . . . .	10
5.4.1	Detailed Description . . . . .	11
5.4.2	Function/Subroutine Documentation . . . . .	11
5.4.2.1	setsimbounds() . . . . .	11
5.4.2.2	setsimdp() . . . . .	11

5.4.2.3	<a href="#">setsimgravity()</a>	12
5.4.2.4	<a href="#">setsimparameter()</a>	12
5.4.2.5	<a href="#">setsimrho()</a>	12
5.4.3	<a href="#">Variable Documentation</a>	13
5.4.3.1	<a href="#">cfl</a>	13
5.4.3.2	<a href="#">dp</a>	13
5.4.3.3	<a href="#">gravity</a>	13
5.4.3.4	<a href="#">initfreeze</a>	13
5.4.3.5	<a href="#">integrator</a>	13
5.4.3.6	<a href="#">pointmax</a>	13
5.4.3.7	<a href="#">pointmin</a>	14
5.4.3.8	<a href="#">rho_ref</a>	14
5.4.3.9	<a href="#">timemax</a>	14
5.4.3.10	<a href="#">timeout</a>	14
5.5	<a href="#">source Module Reference</a>	14
5.5.1	<a href="#">Detailed Description</a>	14
5.6	<a href="#">source_emitter Module Reference</a>	14
5.6.1	<a href="#">Detailed Description</a>	14
5.7	<a href="#">source_identity Module Reference</a>	15
5.7.1	<a href="#">Detailed Description</a>	15
5.7.2	<a href="#">Function/Subroutine Documentation</a>	15
5.7.2.1	<a href="#">allocsources()</a>	15
5.7.2.2	<a href="#">initsource()</a>	15
5.7.3	<a href="#">Variable Documentation</a>	16
5.7.3.1	<a href="#">source</a>	16
5.8	<a href="#">tracer Module Reference</a>	16
5.8.1	<a href="#">Detailed Description</a>	16
5.9	<a href="#">tracer2d Module Reference</a>	16
5.9.1	<a href="#">Detailed Description</a>	17
5.9.2	<a href="#">Function/Subroutine Documentation</a>	17

5.9.2.1	tracer2d_init()	17
5.10	tracer3d Module Reference	17
5.10.1	Detailed Description	18
5.10.2	Function/Subroutine Documentation	18
5.10.2.1	tracer_init()	18
5.11	tracer_interp Module Reference	18
5.12	tracer_precision Module Reference	18
5.12.1	Detailed Description	19
5.12.2	Variable Documentation	19
5.12.2.1	err_dist	19
5.12.2.2	err_ind	19
5.12.2.3	missing_value_default	19
5.12.2.4	mv	19
5.12.2.5	mv_int	19
5.12.2.6	prec	19
5.12.2.7	prec_time	19
5.12.2.8	prec_wrt	19
<b>6</b>	<b>Data Type Documentation</b>	<b>21</b>
6.1	geometry::box Type Reference	21
6.2	geometry::line Type Reference	21
6.2.1	Detailed Description	21
6.2.2	Member Data Documentation	21
6.2.2.1	coordinates	22
6.2.2.2	end	22
6.2.2.3	last	22
6.2.2.4	of	22
6.2.2.5	point	22
6.2.2.6	the	22
6.3	geometry::point Type Reference	22
6.4	geometry::shape Type Reference	23
6.4.1	Member Data Documentation	23
6.4.1.1	a	23
6.4.1.2	coordinates	23
6.4.1.3	of	23
6.4.1.4	point	24
6.4.1.5	pt	24
6.5	geometry::sphere Type Reference	24
6.5.1	Member Data Documentation	24
6.5.1.1	radius	24
6.5.1.2	sphere	24

<b>7 File Documentation</b>	<b>25</b>
7.1 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/app/main.f90 File Reference . .	25
7.2 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/common_modules.f90 File Reference . . . . .	25
7.3 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/geometry.f90 File Reference .	25
7.4 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/initialize.f90 File Reference .	26
7.5 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/simulation_parameters.f90 File Reference . . . . .	26
7.6 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source.f90 File Reference . .	27
7.7 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source_emitter.f90 File Reference . . . . .	27
7.8 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source_identity.f90 File Reference . . . . .	27
7.9 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer.f90 File Reference . .	27
7.10 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer2D.f90 File Reference .	28
7.11 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer3D.f90 File Reference .	28
7.12 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_interp.f90 File Reference	28
7.13 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_precision.f90 File Reference . . . . .	28
<b>Index</b>	<b>29</b>

# Chapter 1

## Modules Index

### 1.1 Modules List

Here is a list of all modules with brief descriptions:

<a href="#">common_modules</a>	Module to hold all of the commonly used base modules . . . . .	9
<a href="#">geometry</a>	Module that defines geometry classes and related methods . . . . .	9
<a href="#">initialize</a>	. . . . .	10
<a href="#">simulation_parameters</a>	Module to hold all the simulation related parameters, definitions and methods . . . . .	10
<a href="#">source</a>	Module to hold and wrap all the tracer sources respective modules. Defines a source class and related methods . . . . .	14
<a href="#">source_emitter</a>	Module that defines a source geometry class, emitter class and related methods . . . . .	14
<a href="#">source_identity</a>	Module that defines a source class and related methods . . . . .	15
<a href="#">tracer</a>	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 . . . . .	16
<a href="#">tracer2d</a>	Module that defines a pure Lagrangian 2D tracer class and related methods, as a subset of the tracer3D module . . . . .	16
<a href="#">tracer3d</a>	Module that defines a pure Lagrangian tracer class and related methods . . . . .	17
<a href="#">tracer_interp</a>	. . . . .	18
<a href="#">tracer_precision</a>	Module to control the precision of the variables trough the project . . . . .	18





## Chapter 2

# Data Type Index

### 2.1 Class Hierarchy

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

geometry::shape . . . . .	23
geometry::box . . . . .	21
geometry::line . . . . .	21
geometry::point . . . . .	22
geometry::sphere . . . . .	24



## Chapter 3

# Data Type Index

### 3.1 Data Types List

Here are the data types with brief descriptions:

<a href="#">geometry::box</a> . . . . .	21
<a href="#">geometry::line</a>	
Type - point class . . . . .	21
<a href="#">geometry::point</a> . . . . .	22
<a href="#">geometry::shape</a> . . . . .	23
<a href="#">geometry::sphere</a> . . . . .	24



## 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 . . . . .	25
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/common_modules.f90 . . . . .	25
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/geometry.f90 . . . . .	25
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/initialize.f90 . . . . .	26
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/simulation_parameters.f90 . . . . .	26
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source.f90 . . . . .	27
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source_emitter.f90 . . . . .	27
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/source_identity.f90 . . . . .	27
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer.f90 . . . . .	27
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer2D.f90 . . . . .	28
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer3D.f90 . . . . .	28
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_interp.f90 . . . . .	28
C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/tracer_precision.f90 . . . . .	28



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

#### 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, public [initmohidlagrangian](#) (xmlfilename)  
*Birjukovs Canelas - MARETEC*

### 5.3.1 Function/Subroutine Documentation

#### 5.3.1.1 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	xmlfilename	
----	-------------	--

## 5.4 simulation\_parameters Module Reference

Module to hold all the simulation related parameters, definitions and methods.

### 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
- real(prec), public `cfl` = 0.5
- real(prec), public `initfreeze` = 0.0
- real(prec), public `timemax` = MV
- real(prec), public `timeout` = MV
- real(prec), public `dp` = MV
- type(vector), public `pointmin`
- type(vector), public `pointmax`
- type(vector), public `gravity`
- real(prec), public `rho_ref` = 1000.0

### 5.4.1 Detailed Description

Module to hold all the simulation related parameters, definitions and methods.

#### Author

Ricardo Birjukovs Canelas

### 5.4.2 Function/Subroutine Documentation

#### 5.4.2.1 setsimbounds()

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

Birjukovs Canelas - MARETEC

Public bounding box setting routine.

#### Parameters

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

#### 5.4.2.2 setsimdps()

```
subroutine, public simulation_parameters::setsimdps (
    type(string), intent(in) read_dp )
```

Birjukovs Canelas - MARETEC

Public dp setting routine.

**Parameters**

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

**5.4.2.3 setsimgravity()**

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

Birjukovs Canelas - MARETEC

Public Gravity setting routine.

**Parameters**

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

**5.4.2.4 setsimparameter()**

```
subroutine, public simulation_parameters::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>	

**5.4.2.5 setsimrho()**

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

Birjukovs Canelas - MARETEC

Public Rho\_Ref setting routine.

## Parameters

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

### 5.4.3 Variable Documentation

#### 5.4.3.1 cfl

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

#### 5.4.3.2 dp

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

#### 5.4.3.3 gravity

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

#### 5.4.3.4 initfreeze

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

#### 5.4.3.5 integrator

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

#### 5.4.3.6 pointmax

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

#### 5.4.3.7 pointmin

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

#### 5.4.3.8 rho\_ref

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

#### 5.4.3.9 timemax

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

#### 5.4.3.10 timeout

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

## 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.

### Functions/Subroutines

- subroutine, public [allocsources](#) (nsources)  
*Birjukovs Canelas - MARETEC*
- subroutine, public [initsource](#) (num, id, name, emitting\_rate, source\_geometry, geometry)  
*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>	
----	-----------------	--

##### 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,id,name,emitting_rate,source_geometry</i>	

### 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	<i>trc</i>	
in	<i>filename</i>	

## 5.10 tracer3d Module Reference

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

## 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	<i>trc</i>	
in	<i>filename</i>	

## 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, 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 err\_dist

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

#### 5.12.2.2 err\_ind

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

#### 5.12.2.3 missing\_value\_default

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

#### 5.12.2.4 mv

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

#### 5.12.2.5 mv\_int

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

#### 5.12.2.6 prec

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

#### 5.12.2.7 prec\_time

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

#### 5.12.2.8 prec\_wrt

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



## Chapter 6

# Data Type Documentation

### 6.1 geometry::box Type Reference

Inheritance diagram for geometry::box:

### 6.2 geometry::line Type Reference

Type - point class.

Inheritance diagram for geometry::line:

Collaboration diagram for geometry::line:

#### Public Attributes

- type(vector) [last](#)  
*Type - line class.*
- type(vector) [coordinates](#)
- type(vector) [of](#)
- type(vector) [the](#)
- type(vector) [end](#)
- type(vector) [point](#)

#### 6.2.1 Detailed Description

Type - point class.

#### 6.2.2 Member Data Documentation

#### 6.2.2.1 coordinates

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

#### 6.2.2.2 end

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

#### 6.2.2.3 last

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

Type - line class.

#### 6.2.2.4 of

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

#### 6.2.2.5 point

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

#### 6.2.2.6 the

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

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

Inheritance diagram for geometry::point:

Collaboration diagram for geometry::point:

## Additional Inherited Members

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

Inheritance diagram for geometry::shape:

### Public Attributes

- type(vector) [pt](#)  
*Type - extendable shape class.*
- type(vector) [coordinates](#)
- type(vector) [of](#)
- type(vector) [a](#)
- type(vector) [point](#)

### 6.4.1 Member Data Documentation

#### 6.4.1.1 a

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

#### 6.4.1.2 coordinates

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

#### 6.4.1.3 of

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

#### 6.4.1.4 point

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

#### 6.4.1.5 pt

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

Type - extendable shape class.

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

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

## 6.5 geometry::sphere Type Reference

Inheritance diagram for geometry::sphere:

Collaboration diagram for geometry::sphere:

### Public Attributes

- real(prec) [radius](#)  
*Type - sphere class.*
- real(prec) [sphere](#)

### 6.5.1 Member Data Documentation

#### 6.5.1.1 radius

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

Type - sphere class.

#### 6.5.1.2 sphere

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

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

- C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/[geometry.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 [commom\\_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 [geometry::point](#)
- type [geometry::line](#)  
*Type - point class.*
- type [geometry::sphere](#)
- type [geometry::box](#)

#### 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, public [initialize::initmohidlagrangian](#) (xmlfilename)  
*Birjukovs Canelas - MARETEC*

## 7.5 C:/Users/administrator/Documents/GitHub/MOHID-Lagrangian/src/lib/simulation\_parameters.f90 File Reference

### Modules

- module [simulation\\_parameters](#)  
*Module to hold all the simulation related parameters, definitions and methods.*

### Functions/Subroutines

- subroutine, public [simulation\\_parameters::setsimpparameter](#) (parmkey, parmvalue)  
*Birjukovs Canelas - MARETEC*
- subroutine, public [simulation\\_parameters::setsimgravity](#) (grav)  
*Birjukovs Canelas - MARETEC*
- subroutine, public [simulation\\_parameters::setsimrho](#) (read\_rho)  
*Birjukovs Canelas - MARETEC*
- subroutine, public [simulation\\_parameters::setsimdp](#) (read\_dp)  
*Birjukovs Canelas - MARETEC*
- subroutine, public [simulation\\_parameters::setsimbounds](#) (point\_, coords)  
*Birjukovs Canelas - MARETEC*

### Variables

- integer, public [simulation\\_parameters::integrator](#) = 1
- real(prec), public [simulation\\_parameters::cfl](#) = 0.5
- real(prec), public [simulation\\_parameters::initfreeze](#) = 0.0
- real(prec), public [simulation\\_parameters::timemax](#) = MV
- real(prec), public [simulation\\_parameters::timeout](#) = MV
- real(prec), public [simulation\\_parameters::dp](#) = MV
- type(vector), public [simulation\\_parameters::pointmin](#)
- type(vector), public [simulation\\_parameters::pointmax](#)
- type(vector), public [simulation\\_parameters::gravity](#)
- real(prec), public [simulation\\_parameters::rho\\_ref](#) = 1000.0



## 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

### 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*

### 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

### 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, 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

# Index

tracer, [5](#)  
tracer2d, [5](#)  
    tracer2d\_init, [6](#)  
tracer2d\_init  
    tracer2d, [6](#)  
tracer3d, [6](#)  
    tracer\_init, [7](#)  
tracer3d::tracer\_class, [9](#)  
tracer3d::tracer\_dep\_class, [9](#)  
tracer3d::tracer\_par\_class, [10](#)  
tracer3d::tracer\_par\_trans\_class, [10](#)  
tracer3d::tracer\_state\_class, [11](#)  
tracer3d::tracer\_stats\_class, [11](#)  
tracer\_init  
    tracer3d, [7](#)  
tracer\_precision, [7](#)