

SWASHES

1.00.01

Generated by Doxygen 1.6.3

Wed Dec 14 18:07:58 2011

Contents

1	Class Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	bump Class Reference	7
4.1.1	Detailed Description	8
4.1.2	Constructor & Destructor Documentation	8
4.1.2.1	bump	8
4.1.2.2	~bump	8
4.1.3	Member Function Documentation	8
4.1.3.1	abcd	8
4.1.3.2	compute	9
4.1.3.3	determinant	9
4.1.3.4	function	9
4.1.3.5	height	9
4.1.3.6	p	9
4.1.3.7	param	10
4.1.3.8	q	10
4.1.3.9	RHJump	10
4.2	choice_solution Class Reference	11
4.2.1	Detailed Description	11
4.2.2	Constructor & Destructor Documentation	11
4.2.2.1	choice_solution	11

4.2.2.2	~choice_solution	11
4.2.3	Member Function Documentation	11
4.2.3.1	compute	11
4.3	dam_break Class Reference	12
4.3.1	Detailed Description	12
4.3.2	Constructor & Destructor Documentation	12
4.3.2.1	dam_break	12
4.3.2.2	~dam_break	12
4.3.3	Member Function Documentation	13
4.3.3.1	compute	13
4.3.3.2	function	13
4.3.3.3	param	13
4.4	Dressler_dam Class Reference	14
4.4.1	Detailed Description	14
4.4.2	Constructor & Destructor Documentation	14
4.4.2.1	Dressler_dam	14
4.4.2.2	~Dressler_dam	14
4.4.3	Member Function Documentation	15
4.4.3.1	compute	15
4.4.3.2	param	15
4.5	MacDonald_like Class Reference	16
4.5.1	Detailed Description	16
4.5.2	Constructor & Destructor Documentation	16
4.5.2.1	MacDonald_like	16
4.5.2.2	~MacDonald_like	17
4.5.3	Member Function Documentation	17
4.5.3.1	compute	17
4.5.3.2	Delta_topo_Darcy_Weisbach	17
4.5.3.3	Delta_topo_Manning	17
4.5.3.4	param	17
4.6	MacDonald_like_diffus Class Reference	18
4.6.1	Detailed Description	18
4.6.2	Constructor & Destructor Documentation	18
4.6.2.1	MacDonald_like_diffus	18
4.6.2.2	~MacDonald_like_diffus	18
4.6.3	Member Function Documentation	19

4.6.3.1	compute	19
4.6.3.2	Delta_topo_diffus	19
4.6.3.3	param	19
4.7	MacDonaldB1 Class Reference	20
4.7.1	Detailed Description	20
4.7.2	Constructor & Destructor Documentation	20
4.7.2.1	MacDonaldB1	20
4.7.2.2	~MacDonaldB1	20
4.7.3	Member Function Documentation	21
4.7.3.1	compute	21
4.7.3.2	Delta_topo	21
4.7.3.3	param	21
4.8	MacDonaldB2 Class Reference	22
4.8.1	Detailed Description	22
4.8.2	Constructor & Destructor Documentation	22
4.8.2.1	MacDonaldB2	22
4.8.2.2	~MacDonaldB2	22
4.8.3	Member Function Documentation	23
4.8.3.1	compute	23
4.8.3.2	Delta_topo	23
4.8.3.3	param	23
4.9	parameters Class Reference	24
4.9.1	Detailed Description	24
4.9.2	Constructor & Destructor Documentation	24
4.9.2.1	parameters	24
4.9.2.2	~parameters	24
4.9.3	Member Function Documentation	24
4.9.3.1	get_choice	24
4.9.3.2	get_choicedim	25
4.9.3.3	get_choicedomain	25
4.9.3.4	get_choicetype	25
4.9.3.5	get_Nxex	25
4.9.3.6	get_Nyex	25
4.9.3.7	help	25
4.9.3.8	setparameters	25
4.9.4	Member Data Documentation	25

4.9.4.1	choice	25
4.9.4.2	choicedim	25
4.9.4.3	choicedomain	25
4.9.4.4	choicetype	25
4.9.4.5	Nx_ex	25
4.9.4.6	Ny_ex	26
4.10	Sampson Class Reference	27
4.10.1	Detailed Description	27
4.10.2	Constructor & Destructor Documentation	27
4.10.2.1	Sampson	27
4.10.2.2	~Sampson	27
4.10.3	Member Function Documentation	28
4.10.3.1	compute	28
4.10.3.2	param	28
4.11	solution Class Reference	29
4.11.1	Detailed Description	30
4.11.2	Constructor & Destructor Documentation	30
4.11.2.1	solution	30
4.11.2.2	~solution	31
4.11.3	Member Function Documentation	31
4.11.3.1	allocation	31
4.11.3.2	compute	31
4.11.3.3	desallocation	31
4.11.3.4	head	31
4.11.3.5	savefinal	31
4.11.3.6	savefinal2D	31
4.11.3.7	savefinalcritical	31
4.11.3.8	savefinalmu	32
4.11.4	Member Data Documentation	32
4.11.4.1	dx_ex	32
4.11.4.2	dy_ex	32
4.11.4.3	hex	32
4.11.4.4	l	32
4.11.4.5	L	32
4.11.4.6	Nx_ex	32
4.11.4.7	Ny_ex	32

4.11.4.8	qex	32
4.11.4.9	T	32
4.11.4.10	uex	32
4.11.4.11	xex	33
4.11.4.12	yex	33
4.11.4.13	zex	33
4.12	Thacker Class Reference	34
4.12.1	Detailed Description	34
4.12.2	Constructor & Destructor Documentation	34
4.12.2.1	Thacker	34
4.12.2.2	~Thacker	34
4.12.3	Member Function Documentation	35
4.12.3.1	compute	35
4.12.3.2	param	35
4.13	Thacker2D Class Reference	36
4.13.1	Detailed Description	36
4.13.2	Constructor & Destructor Documentation	36
4.13.2.1	Thacker2D	36
4.13.2.2	~Thacker2D	36
4.13.3	Member Function Documentation	37
4.13.3.1	compute	37
4.13.3.2	param	37
5	File Documentation	39
5.1	Headers/bump.hpp File Reference	39
5.1.1	Detailed Description	39
5.1.2	Define Documentation	40
5.1.2.1	Classe_bump	40
5.2	Headers/choice_solution.hpp File Reference	41
5.2.1	Detailed Description	41
5.2.2	Define Documentation	41
5.2.2.1	Class_choice_solution	41
5.3	Headers/dam_break.hpp File Reference	42
5.3.1	Detailed Description	42
5.4	Headers/Dressler_dam.hpp File Reference	43
5.4.1	Detailed Description	43
5.5	Headers/MacDonald_like.hpp File Reference	44

5.5.1	Detailed Description	44
5.6	Headers/MacDonald_like_diffus.hpp File Reference	45
5.6.1	Detailed Description	45
5.7	Headers/MacDonaldB1.hpp File Reference	46
5.7.1	Detailed Description	46
5.8	Headers/MacDonaldB2.hpp File Reference	47
5.8.1	Detailed Description	47
5.9	Headers/misc.hpp File Reference	48
5.9.1	Define Documentation	48
5.9.1.1	grav	48
5.9.1.2	grav_dem	48
5.9.1.3	max	48
5.9.1.4	min	48
5.9.1.5	PI	49
5.9.1.6	version	49
5.9.1.7	zero	49
5.9.2	Typedef Documentation	49
5.9.2.1	SCALAR	49
5.9.2.2	TAB	49
5.10	Headers/parameters.hpp File Reference	50
5.10.1	Detailed Description	50
5.11	Headers/Sampson.hpp File Reference	51
5.11.1	Detailed Description	51
5.11.2	Define Documentation	51
5.11.2.1	Class_sampson	51
5.12	Headers/solution.hpp File Reference	52
5.12.1	Detailed Description	52
5.13	Headers/Thacker.hpp File Reference	53
5.13.1	Detailed Description	53
5.14	Headers/Thacker2D.hpp File Reference	54
5.14.1	Detailed Description	54
5.15	Sources/bump.cpp File Reference	55
5.16	Sources/choice_solution.cpp File Reference	56
5.17	Sources/dam_break.cpp File Reference	57
5.18	Sources/Dressler_dam.cpp File Reference	58
5.19	Sources/MacDonald_like.cpp File Reference	59

5.20	Sources/MacDonald_like_diffus.cpp File Reference	60
5.21	Sources/MacDonaldB1.cpp File Reference	61
5.22	Sources/MacDonaldB2.cpp File Reference	62
5.23	Sources/parameters.cpp File Reference	63
5.24	Sources/Sampson.cpp File Reference	64
5.25	Sources/solution.cpp File Reference	65
5.26	Sources/swashes.cpp File Reference	66
5.26.1	Detailed Description	66
5.26.2	Function Documentation	66
5.26.2.1	main	66
5.27	Sources/Thacker.cpp File Reference	67
5.28	Sources/Thacker2D.cpp File Reference	68

Chapter 1

Class Index

1.1 Class Hierarchy

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

choice_solution	11
parameters	24
solution	29
bump	7
dam_break	12
Dressler_dam	14
MacDonald_like	16
MacDonald_like_diffus	18
MacDonaldB1	20
MacDonaldB2	22
Sampson	27
Thacker	34
Thacker2D	36

Chapter 2

Class Index

2.1 Class List

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

bump (Class which allows to perform the bump analytic solutions)	7
choice_solution (Class which allows to choose the analytic solution)	11
dam_break (Class which allows to perform the dam break analytic solution with wet and dry soil)	12
Dressler_dam (Class which allows to perform the Dressler dam break analytic solution)	14
MacDonald_like (Class which allows to perform the MacDonald like analytic solution)	16
MacDonald_like_diffus (Class which allows to perform the MacDonald like analytic solution with diffusion)	18
MacDonaldB1 (Class which allows to perform the MacDonald PSEUDO 2D analytic solution) .	20
MacDonaldB2 (Class which allows to perform the MacDonald PSEUDO 2D analytic solution) .	22
parameters (Class that defines the common parameters)	24
Sampson (Class which allows to perform the Sampson analytic solution)	27
solution (Class which allows to perform the analytic solutions)	29
Thacker (Class which allows to perform the Thacker analytic solution)	34
Thacker2D (Class which allows to perform the Thacker 2D analytic solution)	36

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Headers/ bump.hpp (Performs the bump analytic solutions)	39
Headers/ choice_solution.hpp (Allows to choose the analytic solution)	41
Headers/ dam_break.hpp (Performs the dam break analytic solution)	42
Headers/ Dressler_dam.hpp (Performs classical and modified (personnal communication with Valerio Caleffi, illustration in Valiani et al. 1999) Dressler analytic solution)	43
Headers/ MacDonald_like.hpp (Performs the MacDonald like analytic solutions)	44
Headers/ MacDonald_like_diffus.hpp (Performs the MacDonald like analytic solutions with diffusion)	45
Headers/ MacDonaldB1.hpp (Performs the MacDonald PSEUDO 2D analytic solutions)	46
Headers/ MacDonaldB2.hpp (Performs the MacDonald PSEUDO 2D analytic solutions)	47
Headers/ misc.hpp	48
Headers/ parameters.hpp (Defines the common parameters)	50
Headers/ Sampson.hpp (Performs the Sampson analytic solution)	51
Headers/ solution.hpp (Performs the analytic solutions)	52
Headers/ Thacker.hpp (Performs the Thacker analytic solution)	53
Headers/ Thacker2D.hpp (Performs the Thacker 2D analytic solutions)	54
Sources/ bump.cpp	55
Sources/ choice_solution.cpp	56
Sources/ dam_break.cpp	57
Sources/ Dressler_dam.cpp	58
Sources/ MacDonald_like.cpp	59
Sources/ MacDonald_like_diffus.cpp	60
Sources/ MacDonaldB1.cpp	61
Sources/ MacDonaldB2.cpp	62
Sources/ parameters.cpp	63
Sources/ Sampson.cpp	64
Sources/ solution.cpp	65
Sources/ swashes.cpp (Main function. Declares the solution and calculates the chosen analytic solution for 1D Shallow Water equations)	66
Sources/ Thacker.cpp	67
Sources/ Thacker2D.cpp	68

Chapter 4

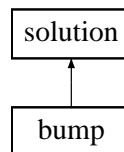
Class Documentation

4.1 bump Class Reference

class which allows to perform the bump analytic solutions

```
#include <bump.hpp>
```

Inheritance diagram for bump:



Public Member Functions

- **bump** (parameters &)
Constructor.
- virtual **~bump** ()
Destructor.
- void **compute** ()
Virtual method which is specific to each analytic solution.
- **SCALAR function** (SCALAR, SCALAR, SCALAR)
• **SCALAR p** (SCALAR a, SCALAR b, SCALAR c)
coefficient Cardano method
- **SCALAR q** (SCALAR a, SCALAR b, SCALAR c, SCALAR d)
coefficient Cardano method
- **SCALAR determinant** (SCALAR p, SCALAR q)
Determinant in the Cardano method/related to number of roots.

- **SCALAR height** (**SCALAR** p, **SCALAR** q, **SCALAR** a, **SCALAR** b, **SCALAR** hnear)
Computation of the 3rd order polynomia roots.
- void **abcd** (**SCALAR** q0, **SCALAR** hfin, **SCALAR** zbx, **SCALAR** zbfm, **SCALAR** &a, **SCALAR** &b, **SCALAR** &c, **SCALAR** &d)
Enters the coefficients of the 3rd order polynomia we want to solve.
- **SCALAR RHJump** (**SCALAR** hplus, **SCALAR** hmois, **SCALAR** q)
steady state RH relation (to check if zero at the shock)
- void **param** (**SCALAR**, **SCALAR**, int)

4.1.1 Detailed Description

class which allows to perform the bump analytic solutions

Definition at line 61 of file bump.hpp.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 bump::bump (parameters & par)

Constructor.

Definition at line 48 of file bump.cpp.

4.1.2.2 bump::~bump () [virtual]

Destructor.

Definition at line 147 of file bump.cpp.

4.1.3 Member Function Documentation

4.1.3.1 void bump::abcd (**SCALAR** q0, **SCALAR** hfin, **SCALAR** zbx, **SCALAR** zbfm, **SCALAR** &a, **SCALAR** &b, **SCALAR** &c, **SCALAR** &d)

Enters the coefficients of the 3rd order polynomia we want to solve.

Parameters

- q0* : Inflow
- hfin* : exit height
- zbx* : bottom of the current cell
- zbfm* : exit bottom
- a* : coeff x3
- b* : coeff x2
- c* : coeff x
- d* : coeff 1

Definition at line 303 of file bump.cpp.

4.1.3.2 void bump::compute () [virtual]

Virtual method which is specific to each analytic solution.

Research of the limit x

PB !!

Computation of the height

Implements [solution](#).

Definition at line 150 of file bump.cpp.

4.1.3.3 SCALAR bump::determinant (SCALAR p , SCALAR q)

Determinant in the Cardano method/related to number of roots.

Parameters

p : computed in function p

q : computed in function q

Definition at line 249 of file bump.cpp.

4.1.3.4 SCALAR bump::function (SCALAR, SCALAR, SCALAR)

4.1.3.5 SCALAR bump::height (SCALAR p , SCALAR q , SCALAR a , SCALAR b , SCALAR h_{near})

Computation of the 3rd order polynomia roots.

Parameters

p : cardano coeff

q : cardano coeff

a :

b :

h_{near} : height of the previous or following cell (depending on the height computation direction)

Definition at line 255 of file bump.cpp.

4.1.3.6 SCALAR bump::p (SCALAR a , SCALAR b , SCALAR c)

coefficient Cardano method

Parameters

a, b, c : coeff of the 3rd order polynomia

Definition at line 237 of file bump.cpp.

4.1.3.7 void bump::param (SCALAR L , SCALAR dx_{ex} , int Nx_{ex})

Definition at line 316 of file bump.cpp.

4.1.3.8 SCALAR bump::q (SCALAR a , SCALAR b , SCALAR c , SCALAR d)

coefficient Cardano method

Parameters

a, b, c, d : coeff of the 3rd order polynomia

Definition at line 243 of file bump.cpp.

4.1.3.9 SCALAR bump::RHJump (SCALAR $hplus$, SCALAR $hminus$, SCALAR q)

steady state RH relation (to check if zero at the shock)

Parameters

$hplus$: height right side

$hminus$: height left side

q : flow

Definition at line 311 of file bump.cpp.

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

- Headers/[bump.hpp](#)
- Sources/[bump.cpp](#)

4.2 choice_solution Class Reference

class which allows to choose the analytic solution.

```
#include <choice_solution.hpp>
```

Public Member Functions

- [choice_solution](#) (parameters &)

Constructor.

- void [compute](#) ()

Performs the solution.

- virtual [~choice_solution](#) ()

Destructor.

4.2.1 Detailed Description

class which allows to choose the analytic solution.

Definition at line 103 of file choice_solution.hpp.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 choice_solution::choice_solution (parameters & par)

Constructor.

Definition at line 45 of file choice_solution.cpp.

4.2.2.2 choice_solution::~~choice_solution () [virtual]

Destructor.

Definition at line 481 of file choice_solution.cpp.

4.2.3 Member Function Documentation

4.2.3.1 void choice_solution::compute ()

Performs the solution.

Definition at line 477 of file choice_solution.cpp.

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

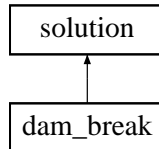
- Headers/[choice_solution.hpp](#)
- Sources/[choice_solution.cpp](#)

4.3 dam_break Class Reference

class which allows to perform the dam break analytic solution with wet and dry soil

```
#include <dam_break.hpp>
```

Inheritance diagram for dam_break:



Public Member Functions

- [dam_break](#) (parameters &)

Constructor.

- virtual [~dam_break](#) ()

Destructor.

- void [compute](#) ()

Virtual method which is specific to each analytic solution.

- [SCALAR function](#) (SCALAR, SCALAR, SCALAR)

*Function $x^6 - 9*cr^2*x^4 + 16*cl*cr^2*x^3 - cr^2*(cr^2 + 8*cl^2)*x^2 + cr^6$ to get the roots by dichotomy.*

- void [param](#) (SCALAR, SCALAR, SCALAR, int, SCALAR)

4.3.1 Detailed Description

class which allows to perform the dam break analytic solution with wet and dry soil

Definition at line 61 of file dam_break.hpp.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 dam_break::dam_break (parameters & par)

Constructor.

Definition at line 48 of file dam_break.cpp.

4.3.2.2 dam_break::~dam_break () [virtual]

Destructor.

Definition at line 90 of file dam_break.cpp.

4.3.3 Member Function Documentation

4.3.3.1 void dam_break::compute () [virtual]

Virtual method which is specific to each analytic solution.

Implements [solution](#).

Definition at line 95 of file dam_break.cpp.

4.3.3.2 SCALAR dam_break::function (SCALAR x , SCALAR v_{left} , SCALAR v_{right})

Function $x^6 - 9cr^2x^4 + 16clcr^2x^3 - cr^2(cr^2 + 8cl^2)x^2 + cr^6$ to get the roots by dichotomy.

Definition at line 167 of file dam_break.cpp.

4.3.3.3 void dam_break::param (SCALAR L , SCALAR x_{dam} , SCALAR dx_{ex} , int Nx_{ex} , SCALAR T)

Definition at line 175 of file dam_break.cpp.

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

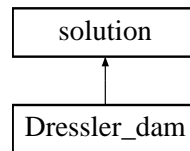
- [Headers/dam_break.hpp](#)
- [Sources/dam_break.cpp](#)

4.4 Dressler_dam Class Reference

class which allows to perform the Dressler dam break analytic solution

```
#include <Dressler_dam.hpp>
```

Inheritance diagram for Dressler_dam:



Public Member Functions

- [Dressler_dam](#) (parameters &)
Constructor.
- virtual [~Dressler_dam](#) ()
Destructor.
- void [compute](#) ()
Virtual method which is specific to each analytic solution.
- void [param](#) (SCALAR, SCALAR, SCALAR, SCALAR, int, SCALAR)

4.4.1 Detailed Description

class which allows to perform the Dressler dam break analytic solution

Definition at line 61 of file Dressler_dam.hpp.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 Dressler_dam::Dressler_dam (parameters & par)

Constructor.

Definition at line 48 of file Dressler_dam.cpp.

4.4.2.2 Dressler_dam::~~Dressler_dam () [virtual]

Destructor.

Definition at line 78 of file Dressler_dam.cpp.

4.4.3 Member Function Documentation

4.4.3.1 void Dressler_dam::compute () [virtual]

Virtual method which is specific to each analytic solution.

Implements [solution](#).

Definition at line 82 of file Dressler_dam.cpp.

4.4.3.2 void Dressler_dam::param (SCALAR *L*, SCALAR *x_{dam}*, SCALAR *C*, SCALAR *dx_{ex}*, int *Nx_{ex}*, SCALAR *T*)

Definition at line 176 of file Dressler_dam.cpp.

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

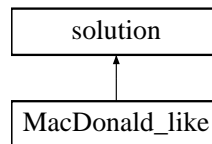
- Headers/[Dressler_dam.hpp](#)
- Sources/[Dressler_dam.cpp](#)

4.5 MacDonald_like Class Reference

class which allows to perform the MacDonald like analytic solution

```
#include <MacDonald_like.hpp>
```

Inheritance diagram for MacDonald_like:



Public Member Functions

- [MacDonald_like](#) (parameters &)

Constructor.

- virtual [~MacDonald_like](#) ()

Destructor.

- void [compute](#) ()

Performs the [MacDonald_like](#) solutions.

- [SCALAR Delta_topo_Manning](#) (SCALAR q, SCALAR h, SCALAR dh, SCALAR Rain, SCALAR n)

Evaluation of the slope variation for Manning friction law.

- [SCALAR Delta_topo_Darcy_Weisbach](#) (SCALAR q, SCALAR h, SCALAR dh, SCALAR Rain, SCALAR n)

Evaluation of the slope variation for Darcy-Weisbach friction law.

- void [param](#) (SCALAR, SCALAR, int)

4.5.1 Detailed Description

class which allows to perform the MacDonald like analytic solution

Definition at line 61 of file MacDonald_like.hpp.

4.5.2 Constructor & Destructor Documentation

4.5.2.1 MacDonald_like::MacDonald_like (parameters & par)

Constructor.

Definition at line 47 of file MacDonald_like.cpp.

4.5.2.2 MacDonald_like::~~MacDonald_like () [virtual]

Destructor.

Definition at line 410 of file MacDonald_like.cpp.

4.5.3 Member Function Documentation

4.5.3.1 void MacDonald_like::compute () [virtual]

Performs the [MacDonald_like](#) solutions.

Implements [solution](#).

Definition at line 415 of file MacDonald_like.cpp.

4.5.3.2 SCALAR MacDonald_like::Delta_topo_Darcy_Weisbach (SCALAR q , SCALAR h , SCALAR dh , SCALAR $Rain$, SCALAR n)

Evaluation of the slope variation for Darcy-Weisbach friction law.

Definition at line 445 of file MacDonald_like.cpp.

4.5.3.3 SCALAR MacDonald_like::Delta_topo_Manning (SCALAR q , SCALAR h , SCALAR dh , SCALAR $Rain$, SCALAR n)

Evaluation of the slope variation for Manning friction law.

Definition at line 441 of file MacDonald_like.cpp.

4.5.3.4 void MacDonald_like::param (SCALAR L , SCALAR dx_{ex} , int Nx_{ex})

Definition at line 450 of file MacDonald_like.cpp.

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

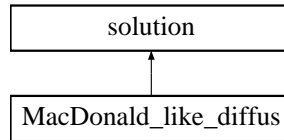
- Headers/[MacDonald_like.hpp](#)
- Sources/[MacDonald_like.cpp](#)

4.6 MacDonald_like_diffus Class Reference

class which allows to perform the MacDonald like analytic solution with diffusion

```
#include <MacDonald_like_diffus.hpp>
```

Inheritance diagram for MacDonald_like_diffus:



Public Member Functions

- [MacDonald_like_diffus](#) (parameters &)
Constructor.
- virtual [~MacDonald_like_diffus](#) ()
Destructor.
- void [compute](#) ()
Performs the MacDonald_like solutions with diffusion.
- [SCALAR](#) [Delta_topo_diffus](#) ([SCALAR](#) q, [SCALAR](#) h, [SCALAR](#) dh, [SCALAR](#) ddh, [SCALAR](#) kt, [SCALAR](#) kl, [SCALAR](#) muv, [SCALAR](#) muh)
Evaluation of the slope variation.
- void [param](#) ([SCALAR](#), [SCALAR](#), int)

4.6.1 Detailed Description

class which allows to perform the MacDonald like analytic solution with diffusion

Definition at line 61 of file MacDonald_like_diffus.hpp.

4.6.2 Constructor & Destructor Documentation

4.6.2.1 MacDonald_like_diffus::MacDonald_like_diffus (parameters & par)

Constructor.

Definition at line 47 of file MacDonald_like_diffus.cpp.

4.6.2.2 MacDonald_like_diffus::~MacDonald_like_diffus () [virtual]

Destructor.

Definition at line 129 of file MacDonald_like_diffus.cpp.

4.6.3 Member Function Documentation

4.6.3.1 void MacDonald_like_diffus::compute () [virtual]

Performs the [MacDonald_like](#) solutions with diffusion.

Implements [solution](#).

Definition at line 135 of file MacDonald_like_diffus.cpp.

4.6.3.2 SCALAR MacDonald_like_diffus::Delta_topo_diffus (SCALAR q , SCALAR h , SCALAR dh , SCALAR ddh , SCALAR kt , SCALAR kl , SCALAR $mu\nu$, SCALAR $mu\eta$)

Evaluation of the slope variation.

Definition at line 154 of file MacDonald_like_diffus.cpp.

4.6.3.3 void MacDonald_like_diffus::param (SCALAR L , SCALAR dx_{ex} , int Nx_{ex})

Definition at line 158 of file MacDonald_like_diffus.cpp.

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

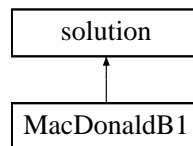
- [Headers/MacDonald_like_diffus.hpp](#)
- [Sources/MacDonald_like_diffus.cpp](#)

4.7 MacDonaldB1 Class Reference

class which allows to perform the MacDonald PSEUDO 2D analytic solution

```
#include <MacDonaldB1.hpp>
```

Inheritance diagram for MacDonaldB1:



Public Member Functions

- [MacDonaldB1](#) ([parameters](#) &)
Constructor.
- virtual [~MacDonaldB1](#) ()
Destructor.
- void [compute](#) ()
Virtual method which is specific to each analytic solution.
- void [param](#) ([SCALAR](#), [SCALAR](#), [SCALAR](#), int)
- [SCALAR](#) [Delta_topo](#) ([SCALAR](#), [SCALAR](#), [SCALAR](#), [SCALAR](#), [SCALAR](#), [SCALAR](#), [SCALAR](#), [SCALAR](#), [SCALAR](#), [SCALAR](#))

4.7.1 Detailed Description

class which allows to perform the MacDonald PSEUDO 2D analytic solution

Definition at line 61 of file MacDonaldB1.hpp.

4.7.2 Constructor & Destructor Documentation

4.7.2.1 MacDonaldB1::MacDonaldB1 ([parameters](#) & *par*)

Constructor.

Definition at line 47 of file MacDonaldB1.cpp.

4.7.2.2 MacDonaldB1::~~MacDonaldB1 () [[virtual](#)]

Destructor.

Definition at line 177 of file MacDonaldB1.cpp.

4.7.3 Member Function Documentation

4.7.3.1 void MacDonaldB1::compute () [virtual]

Virtual method which is specific to each analytic solution.

Implements [solution](#).

Definition at line 142 of file MacDonaldB1.cpp.

4.7.3.2 SCALAR MacDonaldB1::Delta_topo (SCALAR *h*, SCALAR *hp*, SCALAR *b*, SCALAR *bp*, SCALAR *Q*, SCALAR *n*, SCALAR *Z*, SCALAR *exp1*, SCALAR *exp2*)

Definition at line 159 of file MacDonaldB1.cpp.

4.7.3.3 void MacDonaldB1::param (SCALAR *L*, SCALAR *dx_ex*, SCALAR *n*, int *Nx_ex*)

Definition at line 163 of file MacDonaldB1.cpp.

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

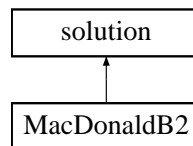
- Headers/[MacDonaldB1.hpp](#)
- Sources/[MacDonaldB1.cpp](#)

4.8 MacDonaldB2 Class Reference

class which allows to perform the MacDonald PSEUDO 2D analytic solution

```
#include <MacDonaldB2.hpp>
```

Inheritance diagram for MacDonaldB2:



Public Member Functions

- [MacDonaldB2](#) (parameters &)
Constructor.
- virtual [~MacDonaldB2](#) ()
Destructor.
- void [compute](#) ()
Virtual method which is specific to each analytic solution.
- void [param](#) (SCALAR, SCALAR, SCALAR, int)
- [SCALAR Delta_topo](#) (SCALAR, SCALAR, SCALAR, SCALAR, SCALAR, SCALAR, SCALAR, SCALAR, SCALAR)

4.8.1 Detailed Description

class which allows to perform the MacDonald PSEUDO 2D analytic solution

Definition at line 61 of file MacDonaldB2.hpp.

4.8.2 Constructor & Destructor Documentation

4.8.2.1 MacDonaldB2::MacDonaldB2 (parameters & par)

Constructor.

Definition at line 47 of file MacDonaldB2.cpp.

4.8.2.2 MacDonaldB2::~~MacDonaldB2 () [virtual]

Destructor.

Definition at line 147 of file MacDonaldB2.cpp.

4.8.3 Member Function Documentation

4.8.3.1 void MacDonaldB2::compute () [virtual]

Virtual method which is specific to each analytic solution.

Implements [solution](#).

Definition at line 112 of file MacDonaldB2.cpp.

4.8.3.2 SCALAR MacDonaldB2::Delta_topo (SCALAR *h*, SCALAR *hp*, SCALAR *b*, SCALAR *bp*, SCALAR *Q*, SCALAR *n*, SCALAR *Z*, SCALAR *exp1*, SCALAR *exp2*)

Definition at line 143 of file MacDonaldB2.cpp.

4.8.3.3 void MacDonaldB2::param (SCALAR *L*, SCALAR *dx_ex*, SCALAR *n*, int *Nx_ex*)

Definition at line 129 of file MacDonaldB2.cpp.

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

- Headers/[MacDonaldB2.hpp](#)
- Sources/[MacDonaldB2.cpp](#)

4.9 parameters Class Reference

class that defines the common parameters.

```
#include <parameters.hpp>
```

Public Member Functions

- [parameters](#) (int, char **)
- void [setparameters](#) (const char *)
- virtual [~parameters](#) ()
- void [help](#) ()
- int [get_Nxex](#) () const
- int [get_Nyex](#) () const
- [SCALAR](#) [get_choicedim](#) () const
- int [get_choicetype](#) () const
- int [get_choice](#) () const
- int [get_choicedomain](#) () const

Protected Attributes

- int [Nx_ex](#)
- int [Ny_ex](#)
- [SCALAR](#) [choicedim](#)
- int [choicetype](#)
- int [choice](#)
- int [choicedomain](#)

4.9.1 Detailed Description

class that defines the common parameters.

Definition at line 57 of file parameters.hpp.

4.9.2 Constructor & Destructor Documentation

4.9.2.1 parameters::parameters (int *argc*, char ** *argv*)

Definition at line 46 of file parameters.cpp.

4.9.2.2 parameters::~~parameters () [virtual]

Definition at line 83 of file parameters.cpp.

4.9.3 Member Function Documentation

4.9.3.1 int parameters::get_choice () const

Definition at line 155 of file parameters.cpp.

4.9.3.2 SCALAR parameters::get_choicedim () const

Definition at line 159 of file parameters.cpp.

4.9.3.3 int parameters::get_choicedomain () const

Definition at line 167 of file parameters.cpp.

4.9.3.4 int parameters::get_choicetype () const

Definition at line 163 of file parameters.cpp.

4.9.3.5 int parameters::get_Nxex () const

Definition at line 147 of file parameters.cpp.

4.9.3.6 int parameters::get_Nyex () const

Definition at line 151 of file parameters.cpp.

4.9.3.7 void parameters::help ()

Definition at line 85 of file parameters.cpp.

4.9.3.8 void parameters::setparameters (const char *)**4.9.4 Member Data Documentation****4.9.4.1 int parameters::choice [protected]**

Definition at line 62 of file parameters.hpp.

4.9.4.2 SCALAR parameters::choicedim [protected]

Definition at line 60 of file parameters.hpp.

4.9.4.3 int parameters::choicedomain [protected]

Definition at line 63 of file parameters.hpp.

4.9.4.4 int parameters::choicetype [protected]

Definition at line 61 of file parameters.hpp.

4.9.4.5 int parameters::Nx_ex [protected]

Definition at line 59 of file parameters.hpp.

4.9.4.6 `int parameters::Ny_ex` [protected]

Definition at line 59 of file `parameters.hpp`.

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

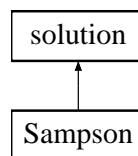
- [Headers/parameters.hpp](#)
- [Sources/parameters.cpp](#)

4.10 Sampson Class Reference

class which allows to perform the [Sampson](#) analytic solution

```
#include <Sampson.hpp>
```

Inheritance diagram for Sampson:



Public Member Functions

- [Sampson](#) (parameters &)

Constructor.

- virtual [~Sampson](#) ()

Destructor.

- void [compute](#) ()

Virtual method which is specific to each analytic solution.

- void [param](#) (SCALAR, SCALAR, SCALAR, SCALAR, SCALAR, SCALAR, SCALAR, int)

4.10.1 Detailed Description

class which allows to perform the [Sampson](#) analytic solution

Definition at line 61 of file Sampson.hpp.

4.10.2 Constructor & Destructor Documentation

4.10.2.1 Sampson::Sampson (parameters & par)

Constructor.

Definition at line 47 of file Sampson.cpp.

4.10.2.2 Sampson::~~Sampson () [virtual]

Destructor.

Definition at line 73 of file Sampson.cpp.

4.10.3 Member Function Documentation

4.10.3.1 void Sampson::compute () [virtual]

Virtual method which is specific to each analytic solution.

Implements [solution](#).

Definition at line 78 of file Sampson.cpp.

4.10.3.2 void Sampson::param (SCALAR *L*, SCALAR *h0*, SCALAR *a*, SCALAR *B*, SCALAR *tau*, SCALAR *dx_ex*, SCALAR *T*, int *Nx_ex*)

Definition at line 98 of file Sampson.cpp.

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

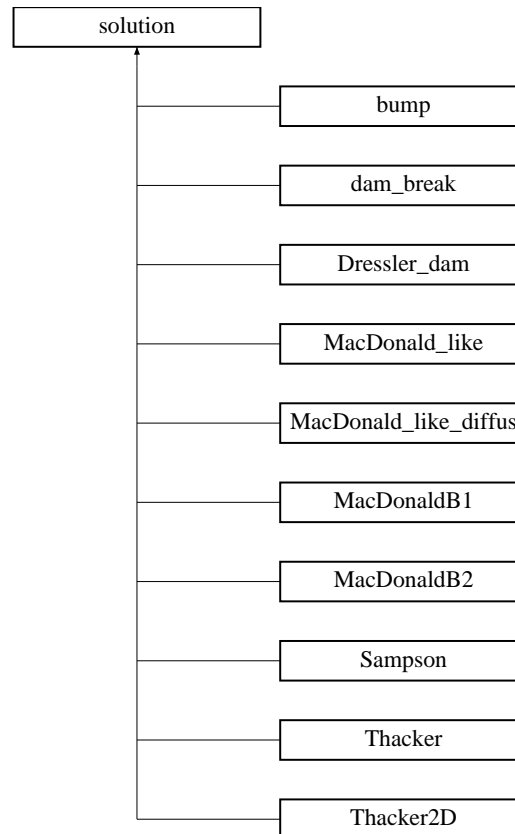
- Headers/[Sampson.hpp](#)
- Sources/[Sampson.cpp](#)

4.11 solution Class Reference

class which allows to perform the analytic solutions.

```
#include <solution.hpp>
```

Inheritance diagram for solution:



Public Member Functions

- [solution](#) (parameters &)
Constructor.
- void [allocation](#) ()
Tables allocation.
- void [desallocation](#) ()
Tables desallocation.
- virtual void [compute](#) ()=0
Virtual method which is specific to each analytic solution.
- void [savefinal](#) (SCALAR *, SCALAR *, SCALAR *, SCALAR *)
Save the analytic solution at the final time.

- void `savefinalmu` (SCALAR *, SCALAR *, SCALAR *)
Save the analytic solution at the final time without u.
- void `savefinal2D` (SCALAR *, SCALAR *, TAB, TAB, TAB, TAB)
Save the analytic solution at the final time in 2D.
- void `savefinalcritical` (SCALAR *, SCALAR *, SCALAR *, SCALAR *)
Save the analytic solution at the final time with the critical height for MacDonald and bumps.
- void `head` (parameters &, string, string)
Write the version of the software and the choice of the solution.
- virtual `~solution` ()
Destructor.

Protected Attributes

- int `Nx_ex`
- int `Ny_ex`
- SCALAR `T`
- SCALAR `L`
- SCALAR `I`
- SCALAR `dx_ex`
- SCALAR `dy_ex`
- SCALAR * `xex`
- SCALAR * `yex`
- SCALAR * `hex`
- SCALAR * `uex`
- SCALAR * `qex`
- SCALAR * `zex`

4.11.1 Detailed Description

class which allows to perform the analytic solutions.

Definition at line 60 of file `solution.hpp`.

4.11.2 Constructor & Destructor Documentation

4.11.2.1 `solution::solution` (parameters & *par*)

Constructor.

Definition at line 47 of file `solution.cpp`.

4.11.2.2 `solution::~~solution () [virtual]`

Destructor.

Definition at line 95 of file solution.cpp.

4.11.3 Member Function Documentation

4.11.3.1 `void solution::allocation ()`

Tables allocation.

Definition at line 100 of file solution.cpp.

4.11.3.2 `virtual void solution::compute () [pure virtual]`

Virtual method which is specific to each analytic solution.

Implemented in [bump](#), [dam_break](#), [Dressler_dam](#), [MacDonald_like](#), [MacDonald_like_diffus](#), [MacDonaldB1](#), [MacDonaldB2](#), [Sampson](#), [Thacker](#), and [Thacker2D](#).

4.11.3.3 `void solution::desallocation ()`

Tables desallocation.

Definition at line 133 of file solution.cpp.

4.11.3.4 `void solution::head (parameters & par, string solutiontype, string solutionchoice)`

Write the version of the software and the choice of the solution.

Definition at line 83 of file solution.cpp.

4.11.3.5 `void solution::savefinal (SCALAR * xex, SCALAR * hex, SCALAR * uex, SCALAR * zex)`

Save the analytic solution at the final time.

Definition at line 53 of file solution.cpp.

4.11.3.6 `void solution::savefinal2D (SCALAR * xex, SCALAR * yex, TAB hex2D, TAB uex2D, TAB vex2D, TAB zex2D)`

Save the analytic solution at the final time in 2D.

Definition at line 67 of file solution.cpp.

4.11.3.7 `void solution::savefinalcritical (SCALAR * xex, SCALAR * hex, SCALAR * qex, SCALAR * zex)`

Save the analytic solution at the final time with the critical height for MacDonald and bumps.

Definition at line 76 of file solution.cpp.

4.11.3.8 void solution::savefinalmu (SCALAR * *xex*, SCALAR * *hex*, SCALAR * *zex*)

Save the analytic solution at the final time without u.

Definition at line 60 of file solution.cpp.

4.11.4 Member Data Documentation**4.11.4.1 SCALAR solution::dx_ex [protected]**

Definition at line 63 of file solution.hpp.

4.11.4.2 SCALAR solution::dy_ex [protected]

Definition at line 63 of file solution.hpp.

4.11.4.3 SCALAR* solution::hex [protected]

Definition at line 67 of file solution.hpp.

4.11.4.4 SCALAR solution::l [protected]

Definition at line 63 of file solution.hpp.

4.11.4.5 SCALAR solution::L [protected]

Definition at line 63 of file solution.hpp.

4.11.4.6 int solution::Nx_ex [protected]

Definition at line 62 of file solution.hpp.

4.11.4.7 int solution::Ny_ex [protected]

Definition at line 62 of file solution.hpp.

4.11.4.8 SCALAR* solution::qex [protected]

Definition at line 69 of file solution.hpp.

4.11.4.9 SCALAR solution::T [protected]

Definition at line 63 of file solution.hpp.

4.11.4.10 SCALAR* solution::uex [protected]

Definition at line 68 of file solution.hpp.

4.11.4.11 SCALAR* solution::xex [protected]

Definition at line 65 of file solution.hpp.

4.11.4.12 SCALAR* solution::yex [protected]

Definition at line 66 of file solution.hpp.

4.11.4.13 SCALAR* solution::zex [protected]

Definition at line 70 of file solution.hpp.

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

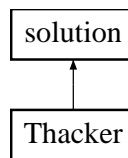
- Headers/[solution.hpp](#)
- Sources/[solution.cpp](#)

4.12 Thacker Class Reference

class which allows to perform the [Thacker](#) analytic solution

```
#include <Thacker.hpp>
```

Inheritance diagram for Thacker:



Public Member Functions

- [Thacker](#) (parameters &)
Constructor.
- virtual [~Thacker](#) ()
Destructor.
- void [compute](#) ()
Virtual method which is specific to each analytic solution.
- void [param](#) (SCALAR, SCALAR, SCALAR, SCALAR, SCALAR, int)

4.12.1 Detailed Description

class which allows to perform the [Thacker](#) analytic solution

Definition at line 61 of file Thacker.hpp.

4.12.2 Constructor & Destructor Documentation

4.12.2.1 Thacker::Thacker (parameters & par)

Constructor.

Definition at line 47 of file Thacker.cpp.

4.12.2.2 Thacker::~~Thacker () [virtual]

Destructor.

Definition at line 72 of file Thacker.cpp.

4.12.3 Member Function Documentation

4.12.3.1 `void Thacker::compute () [virtual]`

Virtual method which is specific to each analytic solution.

Implements [solution](#).

Definition at line 77 of file Thacker.cpp.

4.12.3.2 `void Thacker::param (SCALAR L, SCALAR h0, SCALAR a, SCALAR dx_ex, SCALAR T, int Nx_ex)`

Definition at line 97 of file Thacker.cpp.

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

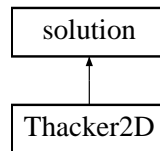
- Headers/[Thacker.hpp](#)
- Sources/[Thacker.cpp](#)

4.13 Thacker2D Class Reference

class which allows to perform the [Thacker](#) 2D analytic solution

```
#include <Thacker2D.hpp>
```

Inheritance diagram for Thacker2D:



Public Member Functions

- [Thacker2D](#) (parameters &)

Constructor.

- virtual [~Thacker2D](#) ()

Destructor.

- void [compute](#) ()

Virtual method which is specific to each analytic solution.

- void [param](#) (SCALAR, SCALAR, SCALAR, SCALAR, SCALAR, SCALAR, SCALAR, int, int)

4.13.1 Detailed Description

class which allows to perform the [Thacker](#) 2D analytic solution

Definition at line 61 of file Thacker2D.hpp.

4.13.2 Constructor & Destructor Documentation

4.13.2.1 Thacker2D::Thacker2D (parameters & par)

Constructor.

Definition at line 47 of file Thacker2D.cpp.

4.13.2.2 Thacker2D::~~Thacker2D () [virtual]

Destructor.

Definition at line 115 of file Thacker2D.cpp.

4.13.3 Member Function Documentation

4.13.3.1 void Thacker2D::compute () [virtual]

Virtual method which is specific to each analytic solution.

Implements [solution](#).

Definition at line 131 of file Thacker2D.cpp.

4.13.3.2 void Thacker2D::param (SCALAR *L*, SCALAR *l*, SCALAR *h0*, SCALAR *a*, SCALAR *dx_ex*, SCALAR *dy_ex*, SCALAR *T*, int *Nx_ex*, int *Ny_ex*)

Definition at line 160 of file Thacker2D.cpp.

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

- Headers/[Thacker2D.hpp](#)
- Sources/[Thacker2D.cpp](#)

Chapter 5

File Documentation

5.1 Headers/bump.hpp File Reference

Performs the bump analytic solutions.

```
#include "solution.hpp"  
#include <vector>  
#include <iomanip>  
#include <iostream>  
#include <cmath>  
#include <stdlib.h>  
#include <fstream>  
#include <complex>  
#include <cstdlib>
```

Classes

- class [bump](#)
class which allows to perform the bump analytic solutions

Defines

- #define [Classe_bump](#)

5.1.1 Detailed Description

Performs the bump analytic solutions.

Author

Anne-Celine Boulanger, Olivier Delestre

Definition in file [bump.hpp](#).

5.1.2 Define Documentation

5.1.2.1 #define Classe_bump

Definition at line 48 of file bump.hpp.

5.2 Headers/choice_solution.hpp File Reference

Allows to choose the analytic solution.

```
#include "solution.hpp"  
#include "dam_break.hpp"
```

Classes

- class [choice_solution](#)
class which allows to choose the analytic solution.

Defines

- #define [Class_choice_solution](#)

5.2.1 Detailed Description

Allows to choose the analytic solution.

Author

Olivier Delestre

Definition in file [choice_solution.hpp](#).

5.2.2 Define Documentation

5.2.2.1 #define Class_choice_solution

Definition at line 90 of file [choice_solution.hpp](#).

5.3 Headers/dam_break.hpp File Reference

Performs the dam break analytic solution.

```
#include "solution.hpp"
```

Classes

- class [dam_break](#)

class which allows to perform the dam break analytic solution with wet and dry soil

5.3.1 Detailed Description

Performs the dam break analytic solution.

Author

Olivier Delestre

Definition in file [dam_break.hpp](#).

5.4 Headers/Dressler_dam.hpp File Reference

Performs classical and modified (personal communication with Valerio Caleffi, illustration in Valiani et al. 1999) Dressler analytic solution.

```
#include "solution.hpp"
```

Classes

- class [Dressler_dam](#)

class which allows to perform the Dressler dam break analytic solution

5.4.1 Detailed Description

Performs classical and modified (personal communication with Valerio Caleffi, illustration in Valiani et al. 1999) Dressler analytic solution.

Author

Olivier Delestre

Definition in file [Dressler_dam.hpp](#).

5.5 Headers/MacDonald_like.hpp File Reference

Performs the MacDonald like analytic solutions.

```
#include "solution.hpp"
```

Classes

- class [MacDonald_like](#)
class which allows to perform the MacDonald like analytic solution

5.5.1 Detailed Description

Performs the MacDonald like analytic solutions.

Author

Olivier Delestre

Definition in file [MacDonald_like.hpp](#).

5.6 Headers/MacDonald_like_diffus.hpp File Reference

Performs the MacDonald like analytic solutions with diffusion.

```
#include "solution.hpp"
```

Classes

- class [MacDonald_like_diffus](#)
class which allows to perform the MacDonald like analytic solution with diffusion

5.6.1 Detailed Description

Performs the MacDonald like analytic solutions with diffusion.

Author

Olivier Delestre

Definition in file [MacDonald_like_diffus.hpp](#).

5.7 Headers/MacDonaldB1.hpp File Reference

Performs the MacDonald PSEUDO 2D analytic solutions.

```
#include "solution.hpp"
```

Classes

- class [MacDonaldB1](#)
class which allows to perform the MacDonald PSEUDO 2D analytic solution

5.7.1 Detailed Description

Performs the MacDonald PSEUDO 2D analytic solutions.

Author

Pierre-Antoine Ksinant and Carine Lucas

Definition in file [MacDonaldB1.hpp](#).

5.8 Headers/MacDonaldB2.hpp File Reference

Performs the MacDonald PSEUDO 2D analytic solutions.

```
#include "solution.hpp"
```

Classes

- class [MacDonaldB2](#)
class which allows to perform the MacDonald PSEUDO 2D analytic solution

5.8.1 Detailed Description

Performs the MacDonald PSEUDO 2D analytic solutions.

Author

Pierre-Antoine Ksinant and Carine Lucas

Definition in file [MacDonaldB2.hpp](#).

5.9 Headers/misc.hpp File Reference

```
#include <vector>
#include <iomanip>
#include <iostream>
#include <cmath>
#include <stdlib.h>
#include <fstream>
#include <complex>
#include <cstdlib>
```

Defines

- #define [max](#)(a, b) (a>=b?a:b)
- #define [min](#)(a, b) (a<=b?a:b)
- #define [grav](#) 9.81
- #define [grav_dem](#) 4.905
- #define [zero](#) 0.
- #define [PI](#) 3.14159265
- #define [version](#) " SWASHES version 1.00.01, November 22, 2011"

Typedefs

- typedef double [SCALAR](#)
- typedef vector< vector< [SCALAR](#) > > [TAB](#)

5.9.1 Define Documentation

5.9.1.1 #define [grav](#) 9.81

Definition at line 55 of file misc.hpp.

5.9.1.2 #define [grav_dem](#) 4.905

Definition at line 56 of file misc.hpp.

5.9.1.3 #define [max](#)(a, b) (a>=b?a:b)

Definition at line 52 of file misc.hpp.

5.9.1.4 #define [min](#)(a, b) (a<=b?a:b)

Definition at line 53 of file misc.hpp.

5.9.1.5 #define PI 3.14159265

Definition at line 58 of file misc.hpp.

5.9.1.6 #define version " SWASHES version 1.00.01, November 22, 2011"

Definition at line 60 of file misc.hpp.

5.9.1.7 #define zero 0.

Definition at line 57 of file misc.hpp.

5.9.2 Typedef Documentation**5.9.2.1 typedef double SCALAR**

Definition at line 64 of file misc.hpp.

5.9.2.2 typedef vector< vector< SCALAR > > TAB

Definition at line 65 of file misc.hpp.

5.10 Headers/parameters.hpp File Reference

Defines the common parameters.

Classes

- class [parameters](#)
class that defines the common parameters.

5.10.1 Detailed Description

Defines the common parameters.

Author

Olivier Delestre

Definition in file [parameters.hpp](#).

5.11 Headers/Sampson.hpp File Reference

Performs the [Sampson](#) analytic solution.

```
#include "solution.hpp"
```

Classes

- class [Sampson](#)
class which allows to perform the [Sampson](#) analytic solution

Defines

- #define [Class_sampson](#)

5.11.1 Detailed Description

Performs the [Sampson](#) analytic solution.

Author

Olivier Delestre

Definition in file [Sampson.hpp](#).

5.11.2 Define Documentation

5.11.2.1 #define Class_sampson

Definition at line 48 of file Sampson.hpp.

5.12 Headers/solution.hpp File Reference

Performs the analytic solutions.

```
#include "misc.hpp"  
#include "parameters.hpp"
```

Classes

- class [solution](#)
class which allows to perform the analytic solutions.

5.12.1 Detailed Description

Performs the analytic solutions.

Author

Olivier Delestre

Definition in file [solution.hpp](#).

5.13 Headers/Thacker.hpp File Reference

Performs the [Thacker](#) analytic solution.

```
#include "solution.hpp"
```

Classes

- class [Thacker](#)
class which allows to perform the [Thacker](#) analytic solution

5.13.1 Detailed Description

Performs the [Thacker](#) analytic solution.

Author

Olivier Delestre

Definition in file [Thacker.hpp](#).

5.14 Headers/Thacker2D.hpp File Reference

Performs the [Thacker](#) 2D analytic solutions.

```
#include "solution.hpp"
```

Classes

- class [Thacker2D](#)
class which allows to perform the [Thacker](#) 2D analytic solution

5.14.1 Detailed Description

Performs the [Thacker](#) 2D analytic solutions.

Author

Pierre-Antoine Ksinant and Carine Lucas

Definition in file [Thacker2D.hpp](#).

5.15 Sources/bump.cpp File Reference

```
#include "bump.hpp"
```

5.16 Sources/choice_solution.cpp File Reference

```
#include "choice_solution.hpp"
```

5.17 Sources/dam_break.cpp File Reference

```
#include "dam_break.hpp"
```

5.18 Sources/Dressler_dam.cpp File Reference

```
#include "Dressler_dam.hpp"
```

5.19 Sources/MacDonald_like.cpp File Reference

```
#include "MacDonald_like.hpp"
```

5.20 Sources/MacDonald_like_diffus.cpp File Reference

```
#include "MacDonald_like_diffus.hpp"
```

5.21 Sources/MacDonaldB1.cpp File Reference

```
#include "MacDonaldB1.hpp"
```

5.22 Sources/MacDonaldB2.cpp File Reference

```
#include "MacDonaldB2.hpp"
```


5.23 Sources/parameters.cpp File Reference

```
#include "misc.hpp"  
#include "parameters.hpp"
```

5.24 Sources/Sampson.cpp File Reference

```
#include "Sampson.hpp"
```

5.25 Sources/solution.cpp File Reference

```
#include "solution.hpp"
```

5.26 Sources/swashes.cpp File Reference

Main function. Declares the solution and calculates the chosen analytic solution for 1D Shallow Water equations.

```
#include "choice_solution.hpp"  
#include "parameters.hpp"
```

Functions

- `int main (int argc, char **argv)`

5.26.1 Detailed Description

Main function. Declares the solution and calculates the chosen analytic solution for 1D Shallow Water equations.

Author

Olivier Delestre

Definition in file [swashes.cpp](#).

5.26.2 Function Documentation

5.26.2.1 `int main (int argc, char ** argv)`

Definition at line 54 of file `swashes.cpp`.

5.27 Sources/Thacker.cpp File Reference

```
#include "Thacker.hpp"
```

5.28 Sources/Thacker2D.cpp File Reference

```
#include "Thacker2D.hpp"
```

Index

- ~Dressler_dam
 - Dressler_dam, 14
- ~MacDonaldB1
 - MacDonaldB1, 20
- ~MacDonaldB2
 - MacDonaldB2, 22
- ~MacDonald_like
 - MacDonald_like, 16
- ~MacDonald_like_diffus
 - MacDonald_like_diffus, 18
- ~Sampson
 - Sampson, 27
- ~Thacker
 - Thacker, 34
- ~Thacker2D
 - Thacker2D, 36
- ~bump
 - bump, 8
- ~choice_solution
 - choice_solution, 11
- ~dam_break
 - dam_break, 12
- ~parameters
 - parameters, 24
- ~solution
 - solution, 30
- abcd
 - bump, 8
- allocation
 - solution, 31
- bump, 7
 - ~bump, 8
 - abcd, 8
 - bump, 8
 - compute, 8
 - determinant, 9
 - function, 9
 - height, 9
 - p, 9
 - param, 9
 - q, 10
 - RHJump, 10
- bump.hpp
 - Classe_bump, 40
- choice
 - parameters, 25
- choice_solution, 11
 - ~choice_solution, 11
 - choice_solution, 11
 - choice_solution, 11
 - compute, 11
- choice_solution.hpp
 - Class_choice_solution, 41
- choicedim
 - parameters, 25
- choicedomain
 - parameters, 25
- choicetype
 - parameters, 25
- Class_choice_solution
 - choice_solution.hpp, 41
- Class_sampson
 - Sampson.hpp, 51
- Classe_bump
 - bump.hpp, 40
- compute
 - bump, 8
 - choice_solution, 11
 - dam_break, 13
 - Dressler_dam, 15
 - MacDonald_like, 17
 - MacDonald_like_diffus, 19
 - MacDonaldB1, 21
 - MacDonaldB2, 23
 - Sampson, 28
 - solution, 31
 - Thacker, 35
 - Thacker2D, 37
- dam_break, 12
 - ~dam_break, 12
 - compute, 13
 - dam_break, 12
 - dam_break, 12
 - function, 13
 - param, 13
- Delta_topo

- MacDonaldB1, 21
- MacDonaldB2, 23
- Delta_topo_Darcy_Weisbach
 - MacDonald_like, 17
- Delta_topo_diffus
 - MacDonald_like_diffus, 19
- Delta_topo_Manning
 - MacDonald_like, 17
- desallocation
 - solution, 31
- determinant
 - bump, 9
- Dressler_dam, 14
 - ~Dressler_dam, 14
 - compute, 15
 - Dressler_dam, 14
 - Dressler_dam, 14
 - param, 15
- dx_ex
 - solution, 32
- dy_ex
 - solution, 32
- function
 - bump, 9
 - dam_break, 13
- get_choice
 - parameters, 24
- get_choicedim
 - parameters, 24
- get_choicedomain
 - parameters, 25
- get_choicetype
 - parameters, 25
- get_Nxex
 - parameters, 25
- get_Nyex
 - parameters, 25
- grav
 - misc.hpp, 48
- grav_dem
 - misc.hpp, 48
- head
 - solution, 31
- Headers/bump.hpp, 39
- Headers/choice_solution.hpp, 41
- Headers/dam_break.hpp, 42
- Headers/Dressler_dam.hpp, 43
- Headers/MacDonald_like.hpp, 44
- Headers/MacDonald_like_diffus.hpp, 45
- Headers/MacDonaldB1.hpp, 46
- Headers/MacDonaldB2.hpp, 47
- Headers/misc.hpp, 48
- Headers/parameters.hpp, 50
- Headers/Sampson.hpp, 51
- Headers/solution.hpp, 52
- Headers/Thacker.hpp, 53
- Headers/Thacker2D.hpp, 54
- height
 - bump, 9
- help
 - parameters, 25
- hex
 - solution, 32
- L
 - solution, 32
- l
 - solution, 32
- MacDonald_like, 16
 - ~MacDonald_like, 16
 - compute, 17
 - Delta_topo_Darcy_Weisbach, 17
 - Delta_topo_Manning, 17
 - MacDonald_like, 16
 - MacDonald_like, 16
 - param, 17
- MacDonald_like_diffus, 18
 - ~MacDonald_like_diffus, 18
 - compute, 19
 - Delta_topo_diffus, 19
 - MacDonald_like_diffus, 18
 - MacDonald_like_diffus, 18
 - param, 19
- MacDonaldB1, 20
 - ~MacDonaldB1, 20
 - compute, 21
 - Delta_topo, 21
 - MacDonaldB1, 20
 - param, 21
- MacDonaldB2, 22
 - ~MacDonaldB2, 22
 - compute, 23
 - Delta_topo, 23
 - MacDonaldB2, 22
 - param, 23
- main
 - swashes.cpp, 66
- max
 - misc.hpp, 48
- min
 - misc.hpp, 48
- misc.hpp
 - grav, 48
 - grav_dem, 48

- max, 48
- min, 48
- PI, 48
- SCALAR, 49
- TAB, 49
- version, 49
- zero, 49
- Nx_ex
 - parameters, 25
 - solution, 32
- Ny_ex
 - parameters, 25
 - solution, 32
- p
 - bump, 9
- param
 - bump, 9
 - dam_break, 13
 - Dressler_dam, 15
 - MacDonald_like, 17
 - MacDonald_like_diffus, 19
 - MacDonaldB1, 21
 - MacDonaldB2, 23
 - Sampson, 28
 - Thacker, 35
 - Thacker2D, 37
- parameters, 24
 - ~parameters, 24
 - choice, 25
 - choicedim, 25
 - choicedomain, 25
 - choicetype, 25
 - get_choice, 24
 - get_choicedim, 24
 - get_choicedomain, 25
 - get_choicetype, 25
 - get_Nxex, 25
 - get_Nyex, 25
 - help, 25
 - Nx_ex, 25
 - Ny_ex, 25
 - parameters, 24
 - setparameters, 25
- PI
 - misc.hpp, 48
- q
 - bump, 10
- qex
 - solution, 32
- RHJump
 - bump, 10
- Sampson, 27
 - ~Sampson, 27
 - compute, 28
 - param, 28
 - Sampson, 27
- Sampson.hpp
 - Class_sampson, 51
- savefinal
 - solution, 31
- savefinal2D
 - solution, 31
- savefinalcritical
 - solution, 31
- savefinalmu
 - solution, 31
- SCALAR
 - misc.hpp, 49
- setparameters
 - parameters, 25
- solution, 29
 - ~solution, 30
 - allocation, 31
 - compute, 31
 - deallocation, 31
 - dx_ex, 32
 - dy_ex, 32
 - head, 31
 - hex, 32
 - L, 32
 - l, 32
 - Nx_ex, 32
 - Ny_ex, 32
 - qex, 32
 - savefinal, 31
 - savefinal2D, 31
 - savefinalcritical, 31
 - savefinalmu, 31
 - solution, 30
 - T, 32
 - uex, 32
 - xex, 32
 - yex, 33
 - zex, 33
- Sources/bump.cpp, 55
- Sources/choice_solution.cpp, 56
- Sources/dam_break.cpp, 57
- Sources/Dressler_dam.cpp, 58
- Sources/MacDonald_like.cpp, 59
- Sources/MacDonald_like_diffus.cpp, 60
- Sources/MacDonaldB1.cpp, 61
- Sources/MacDonaldB2.cpp, 62
- Sources/parameters.cpp, 63

Sources/Sampson.cpp, [64](#)
Sources/solution.cpp, [65](#)
Sources/swashes.cpp, [66](#)
Sources/Thacker.cpp, [67](#)
Sources/Thacker2D.cpp, [68](#)
swashes.cpp
 main, [66](#)

T
 solution, [32](#)

TAB
 misc.hpp, [49](#)

Thacker, [34](#)
 ~Thacker, [34](#)
 compute, [35](#)
 param, [35](#)
 Thacker, [34](#)

Thacker2D, [36](#)
 ~Thacker2D, [36](#)
 compute, [37](#)
 param, [37](#)
 Thacker2D, [36](#)

uex
 solution, [32](#)

version
 misc.hpp, [49](#)

xex
 solution, [32](#)

yex
 solution, [33](#)

zero
 misc.hpp, [49](#)

zex
 solution, [33](#)