Dikes Overtopping

Generated by Doxygen 1.8.9.1

Thu Aug 27 2015 10:24:32

Contents

1 Modules Index **1**

1.1 Modules List 1

2 Data Type Index **3**

2.1 Class List 3

3 File Index **5**

3.1 File List 5

4 Module Documentation **7**

4.1 dllovertopping Module Reference 7

4.1.1 Detailed Description 7

4.1.2 Function/Subroutine Documentation 7

4.1.2.1 calculateqo 7

4.1.2.2 calculateqof 8

4.1.2.3 calczvalue 9

4.1.2.4 versionnumber 10

4.2 factormodulertoovertopping Module Reference 10

4.2.1 Function/Subroutine Documentation 10

4.2.1.1 calculategammab 10

4.2.1.2 calculategammabeta 11

4.2.1.3 calculategammaf 11

4.2.1.4 calculatetanalpha 12

4.3 formulamodulertoovertopping Module Reference 13

4.3.1 Function/Subroutine Documentation 14

4.3.1.1 adjustinfluencefactors 14

4.3.1.2 calculateanglewaveattack 14

4.3.1.3 calculatebreakerlimit 15

4.3.1.4 calculatebreakerparameter 15

4.3.1.5 calculatewavelength 15

4.3.1.6 calculatewaveovertoppingdischarge 16

4.3.1.7 calculatewaverunup 16

4.3.1.8 calculatewavesteepness 17
4.3.1.9 cubicroots 18
4.3.1.10 isequalreal 18
4.3.1.11 isequalzero 19
4.3.1.12 realrootscubicfunction 19
4.3.1.13 rootsdepressedcubic 19
4.3.1.14 rootsgeneralcubic 20
4.4 geometrymodulertoovertopping Module Reference 20
4.4.1 Function/Subroutine Documentation 21
4.4.1.1 adjustnonhorizontalberms 21
4.4.1.2 allocatevectorsgeometry 22
4.4.1.3 calculatehorzdistance 22
4.4.1.4 calculatehorzlengths 23
4.4.1.5 calculatesegmentslopes 23
4.4.1.6 checkcrosssection 24
4.4.1.7 copygeometry 25
4.4.1.8 deallocategeometry 25
4.4.1.9 determinesegmenttypes 26
4.4.1.10 initializegeometry 26
4.4.1.11 isequalgeometry 27
4.4.1.12 mergesequentialberms 28
4.4.1.13 removeberms 29
4.4.1.14 removedikesegments 30
4.4.1.15 splitcrosssection 30
4.4.1.16 writecrosssection 31
4.5 mainmodulertoovertopping Module Reference 31
4.5.1 Function/Subroutine Documentation 32
4.5.1.1 calculateovertopping 32
4.5.1.2 calculateovertoppingnegativefreeboard 33
4.5.1.3 calculateovertoppingsection 33
4.5.1.4 calculatewaveovertopping 34
4.5.1.5 checkinputdata 35
4.5.1.6 convertmodelfactorsrto 36
4.5.1.7 fillmodelfactorsrto 37
4.5.1.8 interpolateresultssections 37
4.5.1.9 iterationwaverunup 38
4.6 overtoppinginterface Module Reference 38
4.6.1 Variable Documentation 39
4.6.1.1 varmodelfactorcriticalovertopping 39
4.7 typedefinitionsrtoovertopping Module Reference 39

4.7.1 Variable Documentation 40
4.7.1.1 berm_max 40
4.7.1.2 berm_min 40
4.7.1.3 fb_max 40
4.7.1.4 fb_min 40
4.7.1.5 fn_max 40
4.7.1.6 fn_min 41
4.7.1.7 frunup1_max 41
4.7.1.8 frunup1_min 41
4.7.1.9 frunup2_max 41
4.7.1.10 frunup2_min 41
4.7.1.11 frunup3_max 41
4.7.1.12 frunup3_min 41
4.7.1.13 fs_max 41
4.7.1.14 fs_min 41
4.7.1.15 margindiff 42
4.7.1.16 margingrad 42
4.7.1.17 rfactor_max 42
4.7.1.18 rfactor_min 42
4.7.1.19 slope_max 42
4.7.1.20 slope_min 42
4.7.1.21 xdiff_min 42
4.7.1.22 z2_iter_max 42
4.7.1.23 z2_margin 42
4.8 zfunctionswtiovertopping Module Reference 43
4.8.1 Detailed Description 43
4.8.2 Function/Subroutine Documentation 43
4.8.2.1 adjustprofile 43
4.8.2.2 calculateqorto 44
4.8.2.3 profileinstructure 45
4.8.2.4 zfunclogratios 46
5 Data Type Documentation 49
5.1 overtoppinginterface::overtoppinggeometrytype Type Reference 49
5.1.1 Detailed Description 50
5.1.2 Member Data Documentation 50
5.1.2.1 normal 50
5.1.2.2 npoints 50
5.1.2.3 roughness 50
5.1.2.4 xcoords 50

5.1.2.5 ycoords 50
5.2 overtoppinginterface::overtoppinggeometrytypef Type Reference 51
5.2.1 Detailed Description 51
5.2.2 Member Data Documentation 51
5.2.2.1 normal 51
5.2.2.2 npoints 51
5.2.2.3 roughness 52
5.2.2.4 xcoords 52
5.2.2.5 ycoords 52
5.3 typedefinitionsrtoovertopping::overtoppingmodelfactors Type Reference 52
5.3.1 Detailed Description 53
5.3.2 Member Data Documentation 53
5.3.2.1 computedovertopping 53
5.3.2.2 criticalovertopping 53
5.3.2.3 factordeterminationq_b_f_b 53
5.3.2.4 factordeterminationq_n_f_n 53
5.3.2.5 frunup1 54
5.3.2.6 frunup2 54
5.3.2.7 frunup3 54
5.3.2.8 fshallow 54
5.4 typedefinitionsrtoovertopping::tpgeometry Type Reference 54
5.4.1 Detailed Description 56
5.4.2 Member Data Documentation 56
5.4.2.1 nbermsegments 56
5.4.2.2 ncoordinates 56
5.4.2.3 psi 56
5.4.2.4 roughnessfactors 56
5.4.2.5 segmentslopes 56
5.4.2.6 segmenttypes 56
5.4.2.7 xcoorddiff 56
5.4.2.8 xcoordinates 57
5.4.2.9 ycoorddiff 57
5.4.2.10 ycoordinates 57
5.5 typedefinitionsrtoovertopping::tpload Type Reference 57
5.5.1 Detailed Description 58
5.5.2 Member Data Documentation 58
5.5.2.1 h 58
5.5.2.2 hm0 58
5.5.2.3 phi 58
5.5.2.4 tm_10 58

5.6 typedefinitionsrtoovertopping::tpmodelfactors Type Reference 58
5.6.1 Detailed Description 59
5.6.2 Member Data Documentation 60
5.6.2.1 fb 60
5.6.2.2 fn 60
5.6.2.3 frunup1 60
5.6.2.4 frunup2 60
5.6.2.5 frunup3 60
5.6.2.6 fs 60
5.7 typedefinitionsrtoovertopping::tpovertopping Type Reference 60
5.7.1 Detailed Description 61
5.7.2 Member Data Documentation 61
5.7.2.1 qo 61
5.7.2.2 z2 61
5.8 overtoppinginterface::tpprofilecoordinate Type Reference 62
5.8.1 Detailed Description 62
5.8.2 Member Data Documentation 62
5.8.2.1 roughness 62
5.8.2.2 xcoordinate 63
5.8.2.3 zcoordinate 63

6 File Documentation 65
6.1 dllOvertopping.f90 File Reference 65
6.1.1 Detailed Description 65
6.2 factorModuleRTOovertopping.f90 File Reference 65
6.2.1 Detailed Description 66
6.3 formulaModuleRTOovertopping.f90 File Reference 66
6.3.1 Detailed Description 67
6.4 geometryModuleRTOovertopping.f90 File Reference 67
6.4.1 Detailed Description 68
6.5 mainModuleRTOovertopping.f90 File Reference 68
6.5.1 Detailed Description 69
6.6 overtoppingInterface.f90 File Reference 69
6.6.1 Detailed Description 69
6.7 typeDefinitionsRTOovertopping.f90 File Reference 69
6.7.1 Detailed Description 71
6.8 zFunctionsWTIOvertopping.f90 File Reference 71
6.8.1 Detailed Description 71

Index 73

Chapter 1

Modules Index

1.1 Modules List

Here is a list of all modules with brief descriptions:

dllovertopping

Calculate one type of overtopping . . . 7

factormodulertoovertopping . . . 10

formulamodulertoovertopping . . . 13

geometrymodulertoovertopping . . . 20

mainmodulertoovertopping . . . 31

overtoppinginterface . . . 38

typedefinitionsrtoovertopping . . . 39

zfunctionswtiovertopping

Module for the Limit State Functions (Z-functions) for wave overtopping . . . 43

Chapter 2

Data Type Index

2.1 Class List

Here are the data types with brief descriptions:

overtoppinginterface::overtoppinggeometrytype 49
overtoppinginterface::overtoppinggeometrytypef 51
typedefinitionsrtoovertopping::overtoppingmodelfactors
OvertoppingModelFactors: C-structure with model factors 52
typedefinitionsrtoovertopping::tpgeometry
TpGeometry: structure with geometry data 54
typedefinitionsrtoovertopping::tpload
TpLoad: structure with load parameters 57
typedefinitionsrtoovertopping::tpmodelfactors
TpModelFactors: structure with model factors 58
typedefinitionsrtoovertopping::tpovertopping
TpOvertopping: structure with overtopping results 60
overtoppinginterface::tpprofilecoordinate 62

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

dllOvertopping.f90

Main entry for the dll DikesOvertopping FUNCTIONS/SUBROUTINES exported from dll↩ Overtopping.dll: 65

factorModuleRTOovertopping.f90

This file contains a module with functions for the slope angle and influence factors 65

formulaModuleRTOovertopping.f90

This file contains a module with the core computations for Dikes Overtopping 66

geometryModuleRTOovertopping.f90

This file contains a module with the core computations for Dikes Overtopping related to the geometry 67

mainModuleRTOovertopping.f90

This file contains a module with the core computations for Dikes Overtopping 68

overtoppingInterface.f90

This file contains the parameters and types (structs) as part of the interface to and from dll↩ Overtopping 69

typeDefinitionsRTOovertopping.f90

This file contains a module with the type definitions for Dikes Overtopping 69

zFunctionsWTIOvertopping.f90

This file contains the limit state functions for wave overtopping within WTI 71

Chapter 4

Module Documentation

4.1 dllovertopping Module Reference

Calculate one type of overtopping.

Functions/Subroutines

- subroutine, public calculateqo (load, geometryInput, dikeHeight, modelFactors, overtopping, success, error↩ Text)

  Subroutine that calculates the discharge needed for the Z-function DikesOvertopping Wrapper for calculateQoF↩ : covert C-like input structures to Fortran input structures.
- subroutine, public calculateqof (load, geometryF, dikeHeight, modelFactors, overtopping, success, errorText)

  Subroutine that calculates the discharge needed for the Z-function DikesOvertopping.
- subroutine, public calczvalue (criticalOvertoppingRate, modelFactors, Qo, z, success, errorMessage)

  Subroutine that calculates the Z-function DikesOvertopping based on the discharge calculated with calculateQoF.
- subroutine, public versionnumber (version)

4.1.1 Detailed Description

Calculate one type of overtopping.

4.1.2 Function/Subroutine Documentation

4.1.2.1 subroutine, public dllovertopping::calculateqo (type(tpload), intent(in) *load,* type(overtoppinggeometrytype), intent(in) *geometryInput,* real(kind=wp), intent(in) *dikeHeight,* type(overtoppingmodelfactors), intent(in) *modelFactors,* type (tpovertopping), intent(out) *overtopping,* logical, intent(out) *success,* character(len=∗), intent(out) *errorText*)

Subroutine that calculates the discharge needed for the Z-function DikesOvertopping Wrapper for calculateQoF: covert C-like input structures to Fortran input structures.

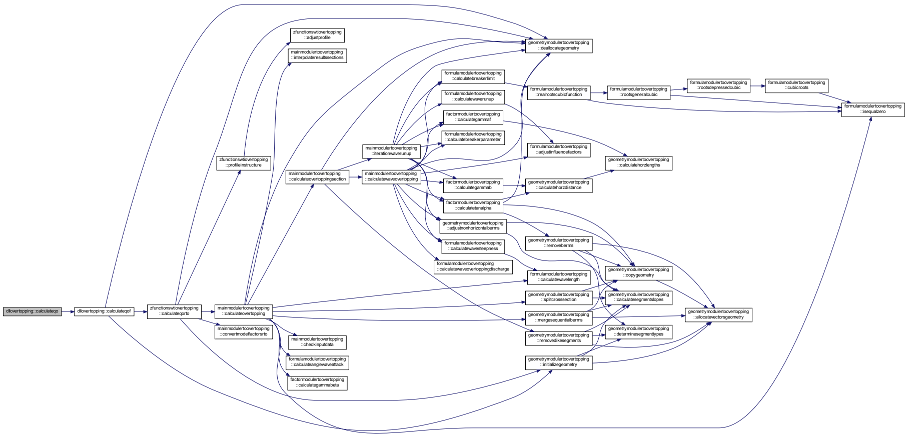
Parameters

`in`	*geometryinput*	struct with geometry and roughness as c-pointers
`in`	*load*	struct with waterlevel and wave parameters
`in`	*modelfactors*	struct with modelfactors

out	*overtopping*	structure with overtopping results
out	*success*	flag for success
out	*errortext*	error message (only set if not successful)

Definition at line 38 of file dllOvertopping.f90.

Here is the call graph for this function:


dllovertopping::calculateqo
dllovertopping::calculateqof
zfunctionswtiovertopping::calculateqorto
mainmodulertoovertopping::calculateovertopping
mainmodulertoovertopping::convertmodelfactorsrto
zfunctionswtiovertopping::profileinstructure
zfunctionswtiovertopping::adjustprofile
mainmodulertoovertopping::interpolateresultssections
mainmodulertoovertopping::calculateovertoppingsection
mainmodulertoovertopping::iterationwaverunup
mainmodulertoovertopping::calculatewaveovertopping
mainmodulertoovertopping::checkinputdata
formulamodulertoovertopping::calculateanglewaveattack
factormodulertoovertopping::calculategammabeta
geometrymodulertoovertopping::deallocategeometry
formulamodulertoovertopping::calculatebreakerlimit
formulamodulertoovertopping::calculatewaverunup
factormodulertoovertopping::calculategammaf
formulamodulertoovertopping::calculatebreakerparameter
factormodulertoovertopping::calculategammab
factormodulertoovertopping::calculatetanalpha
geometrymodulertoovertopping::adjustnonhorizontalberms
formulamodulertoovertopping::calculatewavesteepness
formulamodulertoovertopping::calculatewaveovertoppingdischarge
formulamodulertoovertopping::realrootscubicfunction
formulamodulertoovertopping::rootsgeneralcubic
formulamodulertoovertopping::rootsdepressedcubic
formulamodulertoovertopping::cubicroots
formulamodulertoovertopping::isequalzero
formulamodulertoovertopping::adjustinfluencefactors
geometrymodulertoovertopping::calculatehorzlengths
geometrymodulertoovertopping::calculatehorzdistance
geometrymodulertoovertopping::removeberms
formulamodulertoovertopping::calculatewavelength
geometrymodulertoovertopping::copygeometry
geometrymodulertoovertopping::splitcrosssection
geometrymodulertoovertopping::calculatesegmentslopes
geometrymodulertoovertopping::mergesequentialberms
geometrymodulertoovertopping::allocatevectorsgeometry
geometrymodulertoovertopping::removedikesegments
geometrymodulertoovertopping::determinesegmenttypes
geometrymodulertoovertopping::initializegeometry

4.1.2.2 subroutine, public dllovertopping::calculateqof (type(tpload), intent(in) *load,* type(overtoppinggeometrytypef), intent(in) *geometryF,* real(kind=wp), intent(in) *dikeHeight,* type(overtoppingmodelfactors), intent(in) *modelFactors,* type (tpovertopping), intent(out) *overtopping,* logical, intent(out) *success,* character(len=∗), intent(out) *errorText*)

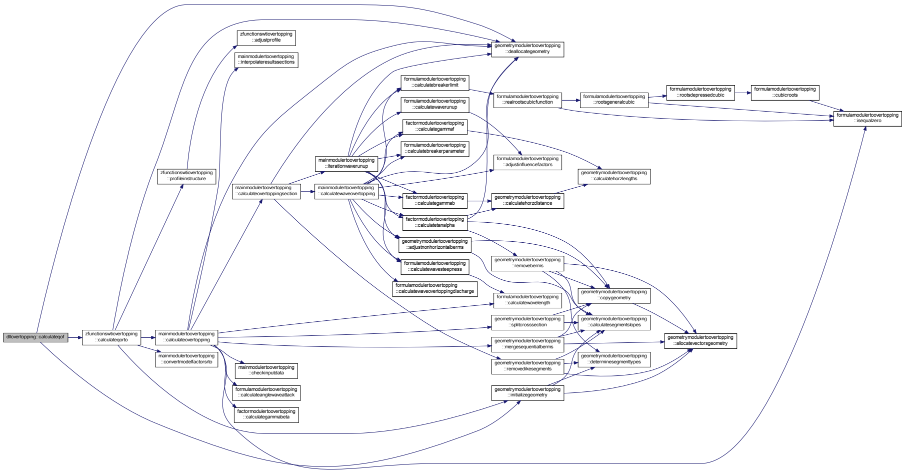
Subroutine that calculates the discharge needed for the Z-function DikesOvertopping.

Parameters

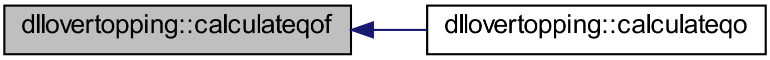
in	*geometryf*	struct with geometry and roughness
in	*load*	struct with waterlevel and wave parameters
in	*modelfactors*	struct with modelFactors
out	*overtopping*	structure with overtopping results
out	*success*	flag for success
out	*errortext*	error message (only set if not successful)

Definition at line 70 of file dllOvertopping.f90.

Here is the call graph for this function:


dllovertopping::calculateqof
zfunctionswtiovertopping ::calculateqorto
zfunctionswtiovertopping ::adjustprofile
mainmodulertoovertopping ::interpolateresultssections
zfunctionswtiovertopping ::profileinstructure
mainmodulertoovertopping ::calculateovertoppingsection
mainmodulertoovertopping ::iterationwaverunup
mainmodulertoovertopping ::calculatewaveovertopping
mainmodulertoovertopping ::calculateovertopping
mainmodulertoovertopping ::convertmodelfactorsrto
mainmodulertoovertopping ::checkinputdata
formulamodulertoovertopping ::calculateanglewaveattack
factormodulertoovertopping ::calculategammabeta
formulamodulertoovertopping ::calculatebreakerlimit
formulamodulertoovertopping ::calculatewaverunup
factormodulertoovertopping ::calculategammaf
formulamodulertoovertopping ::calculatebreakerparameter
factormodulertoovertopping ::calculategammab
factormodulertoovertopping ::calculatetanalpha
geometrymodulertoovertopping ::adjustnonhorizontalberms
formulamodulertoovertopping ::calculatewavesteepness
formulamodulertoovertopping ::calculatewaveovertoppingdischarge
geometrymodulertoovertopping ::deallocategeometry
formulamodulertoovertopping ::realrootscubicfunction
formulamodulertoovertopping ::rootsgeneralcubic
formulamodulertoovertopping ::rootsdepressedcubic
formulamodulertoovertopping ::cubicroots
formulamodulertoovertopping ::isequalzero
formulamodulertoovertopping ::adjustinfluencefactors
geometrymodulertoovertopping ::calculatehorzlengths
geometrymodulertoovertopping ::calculatehorzdistance
geometrymodulertoovertopping ::removeberms
formulamodulertoovertopping ::calculatewavelength
geometrymodulertoovertopping ::splitcrosssection
geometrymodulertoovertopping ::mergesequentialberms
geometrymodulertoovertopping ::removedikesegments
geometrymodulertoovertopping ::initializegeometry
geometrymodulertoovertopping ::copygeometry
geometrymodulertoovertopping ::calculatesegmentslopes
geometrymodulertoovertopping ::determinesegmenttypes
geometrymodulertoovertopping ::allocatevectorsgeometry

Here is the caller graph for this function:



4.1.2.3 subroutine, public dllovertopping::calczvalue (real(kind=wp), intent(in) *criticalOvertoppingRate,* type(overtoppingmodelfactors), intent(in) *modelFactors,* real(kind=wp), intent(in) *Qo,* real(kind=wp), intent(out) *z,* logical, intent(out) *success,* character(len=∗), intent(out) *errorMessage*)

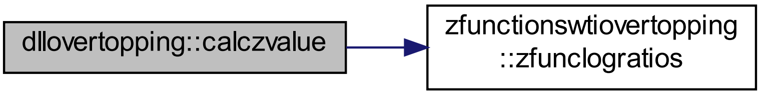
Subroutine that calculates the Z-function DikesOvertopping based on the discharge calculated with calculateQoF.

Parameters

in	*criticalovertop-pingrate*	critical overtoppingrate
in	*modelfactors*	struct with modelfactors
in	*qo*	calculated discharge
out	*z*	z value
out	*errormessage*	error message (only if not successful)
out	*success*	flag for success

Definition at line 98 of file dllOvertopping.f90.

Here is the call graph for this function:


dllovertopping::calczvalue
zfunctionswtiovertopping ::zfunclogratios

4.1.2.4 subroutine, public dllovertopping::versionnumber (character(len=∗), intent(out) *version*)

Parameters

out	*version*	version number

Definition at line 113 of file dllOvertopping.f90.

4.2 factormodulertoovertopping Module Reference

Functions/Subroutines

- subroutine, public calculatetanalpha (h, Hm0, z2, geometry, tanAlpha, succes, errorMessage)

  calculateTanAlpha representative slope angle
- subroutine, public calculategammabeta (Hm0, Tm_10, beta, gammaBeta_z, gammaBeta_o)

  calculateGammaBeta influence factor angle of wave attack
- subroutine, public calculategammaf (h, ksi0, ksi0Limit, gammaB, z2, geometry, gammaF, succes, error↩ Message)

  calculateGammaF influence factor roughness
- subroutine, public calculategammab (h, Hm0, z2, geometry, NbermSegments, gammaB, succes, error↩ Message)

  calculateGammaB influence factor berms

4.2.1 Function/Subroutine Documentation

4.2.1.1 subroutine, public factormodulertoovertopping::calculategammab (real(wp), intent(in) *h,* real(wp), intent(in) *Hm0,* real(wp), intent(in) *z2,* type(tpgeometry), intent(in) *geometry,* integer, intent(in) *NbermSegments,* real(wp), intent(out) *gammaB,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

calculateGammaB influence factor berms

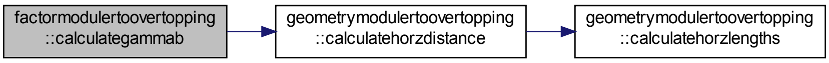
Parameters

in	*h*	local water level (m+NAP)
in	*hm0*	significant wave height (m)
in	*z2*	2% wave run-up (m)
in	*geometry*	structure with geometry data

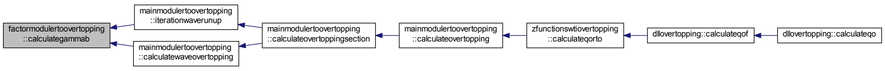
in	*nbermsegments*	number of berm segments
out	*gammab*	influence factor berms
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 271 of file factorModuleRTOovertopping.f90.

Here is the call graph for this function:


factormodulertoovertopping ::calculategammab
geometrymodulertoovertopping ::calculatehorzdistance
geometrymodulertoovertopping ::calculatehorzlengths

Here is the caller graph for this function:


factormodulertoovertopping ::calculategammab
mainmodulertoovertopping ::iterationwaverunup
mainmodulertoovertopping ::calculatewaveovertopping
mainmodulertoovertopping ::calculateovertoppingsection
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.2.1.2 subroutine, public factormodulertoovertopping::calculategammabeta (real(wp), intent(inout) *Hm0,* real(wp), intent(inout) *Tm_10,* real(wp), intent(in) *beta,* real(wp), intent(out) *gammaBeta_z,* real(wp), intent(out) *gammaBeta_o*)

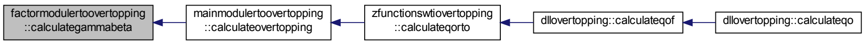
calculateGammaBeta influence factor angle of wave attack

Parameters

in,out	*hm0*	significant wave height (m)
in,out	*tm_10*	spectral wave period (s)
in	*beta*	angle of wave attack (degree)
out	*gammabeta_z*	influence factor angle of wave attack 2% wave run-up
out	*gammabeta_o*	influence factor angle of wave attack overtopping

Definition at line 114 of file factorModuleRTOovertopping.f90.

Here is the caller graph for this function:


factormodulertoovertopping ::calculategammabeta
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.2.1.3 subroutine, public factormodulertoovertopping::calculategammaf (real(wp), intent(in) *h,* real(wp), intent(in) *ksi0,* real(wp), intent(in) *ksi0Limit,* real(wp), intent(in) *gammaB,* real(wp), intent(in) *z2,* type(tpgeometry), intent(in) *geometry,* real(wp), intent(out) *gammaF,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

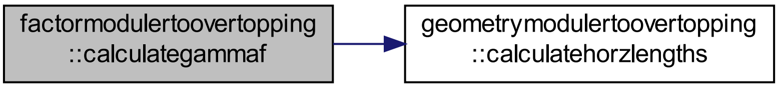
calculateGammaF influence factor roughness

Parameters

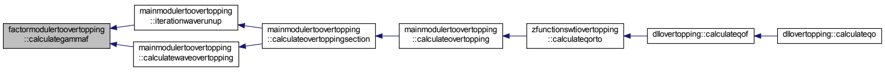
in	*h*	local water level (m+NAP)
in	*ksi0*	breaker parameter
in	*ksi0limit*	limit value breaker parameter
in	*gammab*	influence factor berms
in	*z2*	2% wave run-up (m)
in	*geometry*	structure with geometry data
out	*gammaf*	influence factor roughness
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 154 of file factorModuleRTOovertopping.f90.

Here is the call graph for this function:



Here is the caller graph for this function:



4.2.1.4 subroutine, public factormodulertoovertopping::calculatetanalpha (real(wp), intent(in) *h*, real(wp), intent(in) *Hm0*, real(wp), intent(in) *z2*, type(tpgeometry), intent(in) *geometry*, real(wp), intent(out) *tanAlpha*, logical, intent(out) *succes*, character(len=∗), intent(out) *errorMessage*)

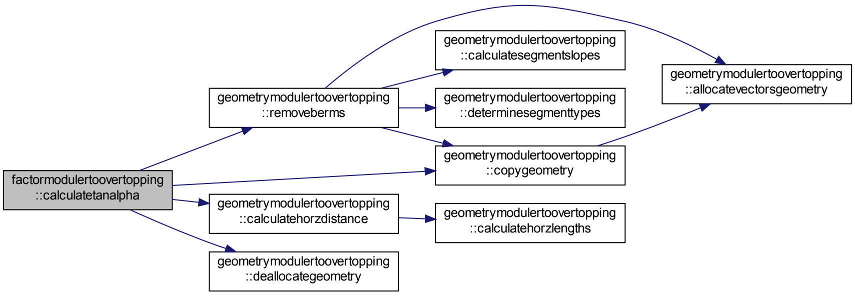
calculateTanAlpha representative slope angle

Parameters

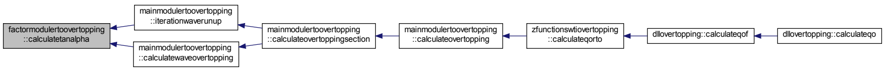
in	*h*	local water level (m+NAP)
in	*hm0*	significant wave height (m)
in	*z2*	2% wave run-up (m)
in	*geometry*	structure with geometry data
out	*tanalpha*	representative slope angle
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 35 of file factorModuleRTOovertopping.f90.

Here is the call graph for this function:


factormodulertoovertopping
::calculatetanalpha
geometrymodulertoovertopping
::removeberms
geometrymodulertoovertopping
::calculatesegmentslopes
geometrymodulertoovertopping
::determinesegmenttypes
geometrymodulertoovertopping
::copygeometry
geometrymodulertoovertopping
::allocatevectorsgeometry
geometrymodulertoovertopping
::calculatehorzdistance
geometrymodulertoovertopping
::calculatehorzlengths
geometrymodulertoovertopping
::deallocategeometry

Here is the caller graph for this function:



4.3 formulamodulertoovertopping Module Reference

Functions/Subroutines

- subroutine, public calculatewaverunup (Hm0, s0, ksi0, ksi0Limit, gammaB, gammaF, gammaBeta, model←↩Factors, z2, succes, errorMessage)

  calculateWaveRunup: calculate wave runup
- subroutine, public calculatewaveovertoppingdischarge (h, Hm0, tanAlpha, gammaB, gammaF, gammaBeta, ksi0, hCrest, modelFactors, Qo, succes, errorMessage)

  calculateWaveOvertoppingDischarge: calculate the wave overtopping discharge
- subroutine, public calculatewavelength (Tm_10, L0)

  calculateWaveLength: calculate the wave length
- subroutine, public calculatewavesteepness (Hm0, Tm_10, s0, succes, errorMessage)

  calculateWaveSteepness: calculate the wave steepness
- subroutine, public calculatebreakerparameter (tanAlpha, s0, ksi0, succes, errorMessage)

  calculateBreakerParameter: calculate the breaker parameter
- subroutine, public calculateanglewaveattack (phi, psi, beta)

  calculateAngleWaveAttack: calculate the angle of wave attack
- subroutine, public calculatebreakerlimit (modelFactors, gammaB, ksi0Limit, succes, errorMessage)

  calculateBreakerLimit: calculate the breaker limit
- subroutine, public adjustinfluencefactors (gammaB, gammaF, gammaBeta, gammaBetaType, ksi0, ksi0Limit, succes, errorMessage)

  adjustInfluenceFactors: adjust the influence factors
- subroutine, public realrootscubicfunction (a, b, c, d, N, x, succes, errorMessage)

  realRootsCubicFunction: calculate the roots of a cubic function
- subroutine, public rootsgeneralcubic (a, b, c, d, z, succes, errorMessage)

  rootsGeneralCubic: calculate the roots of a generic cubic function
- subroutine, public rootsdepressedcubic (p, q, z)

  rootsDepressedCubic: calculate the roots of a depressed cubic function

- subroutine, public cubicroots (z, roots)

  cubicRoots: calculate the roots of a cubic function

- logical function, public isequalreal (x1, x2)

  isEqualReal: are two reals (almost) equal

- logical function, public isequalzero (x)

  isEqualZero: is a real (almost) zero

4.3.1 Function/Subroutine Documentation

4.3.1.1 subroutine, public formulamodulertoovertopping::adjustinfluencefactors (real(wp), intent(inout) *gammaB*, real(wp), intent(inout) *gammaF*, real(wp), intent(inout) *gammaBeta*, integer, intent(in) *gammaBetaType*, real(wp), intent(in) *ksi0*, real(wp), intent(in) *ksi0Limit*, logical, intent(out) *succes*, character(len=∗), intent(out) *errorMessage*)

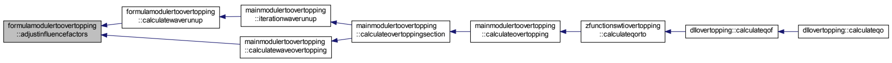
adjustInfluenceFactors: adjust the influence factors

Parameters

`in,out`	*gammab*	influence factor berms
`in,out`	*gammaf*	influence factor roughness
`in,out`	*gammabeta*	influence factor angle of wave attack
`in`	*gammabetatype*	type influence factor angle of wave attack: 1 = wave run-up, 2 = overtopping
`in`	*ksi0*	breaker parameter
`in`	*ksi0limit*	limit value breaker parameter
`out`	*succes*	flag for succes
`out`	*errormessage*	error message

Definition at line 405 of file formulaModuleRTOovertopping.f90.

Here is the caller graph for this function:


formulamodulertoovertopping ::adjustinfluencefactors
formulamodulertoovertopping ::calculatewaverunup
mainmodulertoovertopping ::iterationwaverunup
mainmodulertoovertopping ::calculatewaveovertopping
mainmodulertoovertopping ::calculateovertoppingsection
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.3.1.2 subroutine, public formulamodulertoovertopping::calculateanglewaveattack (real(wp), intent(in) *phi*, real(wp), intent(in) *psi*, real(wp), intent(out) *beta*)

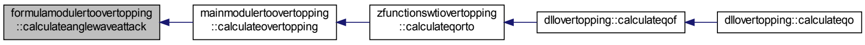
calculateAngleWaveAttack: calculate the angle of wave attack

Parameters

`in`	*phi*	wave direction (degree)
`in`	*psi*	dike normal (degree)
`out`	*beta*	angle of wave attack (degree)

Definition at line 308 of file formulaModuleRTOovertopping.f90.

Here is the caller graph for this function:

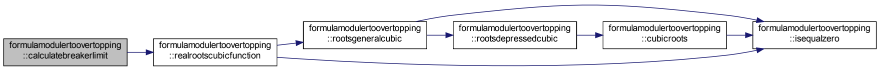

formulamodulertoovertopping ::calculateanglewaveattack
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.3.1.3 subroutine, public formulamodulertoovertopping::calculatebreakerlimit (type (tpmodelfactors), intent(in) *modelFactors,* real(wp), intent(in) *gammaB,* real(wp), intent(out) *ksi0Limit,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

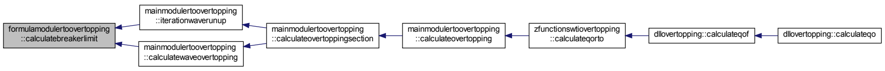
calculateBreakerLimit: calculate the breaker limit

Definition at line 331 of file formulaModuleRTOovertopping.f90.

Here is the call graph for this function:


formulamodulertoovertopping ::calculatebreakerlimit
formulamodulertoovertopping ::realrootscubicfunction
formulamodulertoovertopping ::rootsgeneralcubic
formulamodulertoovertopping ::rootsdepressedcubic
formulamodulertoovertopping ::cubicroots
formulamodulertoovertopping ::isequalzero

Here is the caller graph for this function:


formulamodulertoovertopping ::calculatebreakerlimit
mainmodulertoovertopping ::iterationwaverunup
mainmodulertoovertopping ::calculatewaveovertopping
mainmodulertoovertopping ::calculateovertoppingsection
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.3.1.4 subroutine, public formulamodulertoovertopping::calculatebreakerparameter (real(wp), intent(in) *tanAlpha,* real(wp), intent(in) *s0,* real(wp), intent(out) *ksi0,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

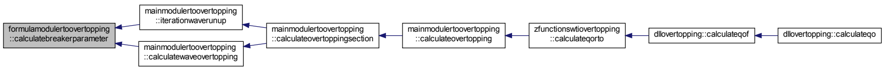
calculateBreakerParameter: calculate the breaker parameter

Parameters

in	*tanalpha*	representative slope angle
in	*s0*	wave steepness
out	*ksi0*	breaker parameter
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 267 of file formulaModuleRTOovertopping.f90.

Here is the caller graph for this function:


formulamodulertoovertopping ::calculatebreakerparameter
mainmodulertoovertopping ::iterationwaverunup
mainmodulertoovertopping ::calculatewaveovertopping
mainmodulertoovertopping ::calculateovertoppingsection
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.3.1.5 subroutine, public formulamodulertoovertopping::calculatewavelength (real(wp), intent(in) *Tm_10,* real(wp), intent(out) *L0*)

calculateWaveLength: calculate the wave length

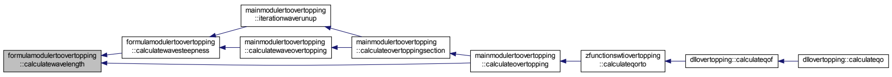
Parameters

in	*tm_10*	spectral wave period (s)

out	*l0*	wave length (m)

Definition at line 204 of file formulaModuleRTOovertopping.f90.

Here is the caller graph for this function:


mainmodulertoovertopping ::iterationwaverunup
formulamodulertoovertopping ::calculatewavelength
formulamodulertoovertopping ::calculatewavesteepness
mainmodulertoovertopping ::calculatewaveovertopping
mainmodulertoovertopping ::calculateovertoppingsection
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.3.1.6 subroutine, public formulamodulertoovertopping::calculatewaveovertoppingdischarge (real(wp), intent(in) *h*, real(wp), intent(in) *Hm0*, real(wp), intent(in) *tanAlpha*, real(wp), intent(in) *gammaB*, real(wp), intent(in) *gammaF*, real(wp), intent(in) *gammaBeta*, real(wp), intent(in) *ksi0*, real(wp), intent(in) *hCrest*, type(tpmodelfactors), intent(in) *modelFactors*, real(wp), intent(out) *Qo*, logical, intent(out) *succes*, character(len=∗), intent(out) *errorMessage*)

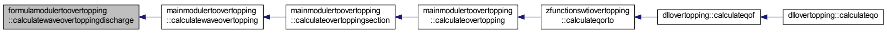
calculateWaveOvertoppingDischarge: calculate the wave overtopping discharge

Parameters

in	*h*	local water level (m+NAP)
in	*hm0*	significant wave height (m)
in	*tanalpha*	representative slope angle
in	*gammab*	influence factor berms
in	*gammaf*	influence factor roughness
in	*gammabeta*	influence factor angle of wave attack
in	*ksi0*	breaker parameter
in	*hcrest*	crest level (m+NAP)
in	*modelfactors*	structure with model factors
out	*qo*	wave overtopping discharge (l/m per s)
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 106 of file formulaModuleRTOovertopping.f90.

Here is the caller graph for this function:


formulamodulertoovertopping ::calculatewaveovertoppingdischarge
mainmodulertoovertopping ::calculatewaveovertopping
mainmodulertoovertopping ::calculateovertoppingsection
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.3.1.7 subroutine, public formulamodulertoovertopping::calculatewaverunup (real(wp), intent(in) *Hm0*, real(wp), intent(in) *s0*, real(wp), intent(in) *ksi0*, real(wp), intent(in) *ksi0Limit*, real(wp), intent(inout) *gammaB*, real(wp), intent(inout) *gammaF*, real(wp), intent(inout) *gammaBeta*, type (tpmodelfactors), intent(in) *modelFactors*, real(wp), intent(out) *z2*, logical, intent(out) *succes*, character(len=∗), intent(out) *errorMessage*)

calculateWaveRunup: calculate wave runup

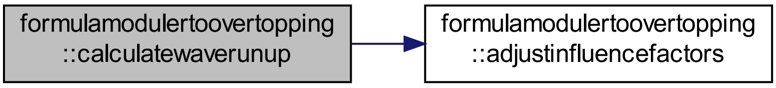
Parameters

in	*hm0*	significant wave height (m)
in	*s0*	wave steepness

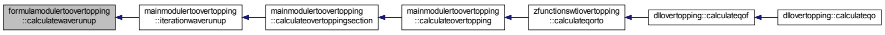
in	*ksi0*	breaker parameter
in	*ksi0limit*	limit value breaker parameter
in,out	*gammab*	influence factor berms
in,out	*gammaf*	influence factor roughness
in,out	*gammabeta*	influence factor angle of wave attack
in	*modelfactors*	structure with model factors
out	*z2*	2% wave run-up (m)
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 34 of file formulaModuleRTOovertopping.f90.

Here is the call graph for this function:


formulamodulertoovertopping ::calculatewaverunup
formulamodulertoovertopping ::adjustinfluencefactors

Here is the caller graph for this function:


formulamodulertoovertopping ::calculatewaverunup
mainmodulertoovertopping ::iterationwaverunup
mainmodulertoovertopping ::calculateovertoppingsection
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.3.1.8 subroutine, public formulamodulertoovertopping::calculatewavesteepness (real(wp), intent(in) *Hm0*, real(wp), intent(in) *Tm_10*, real(wp), intent(out) *s0*, logical, intent(out) *succes*, character(len=∗), intent(out) *errorMessage*)

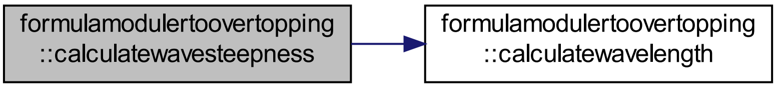
calculateWaveSteepness: calculate the wave steepness

Parameters

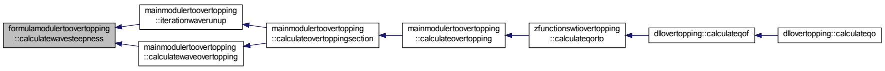
in	*hm0*	significant wave height (m)
in	*tm_10*	spectral wave period (s)
out	*s0*	wave steepness
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 225 of file formulaModuleRTOovertopping.f90.

Here is the call graph for this function:


formulamodulertoovertopping ::calculatewavesteepness
formulamodulertoovertopping ::calculatewavelength

Here is the caller graph for this function:



4.3.1.9 subroutine, public formulamodulertoovertopping::cubicroots (double complex, intent(in) *z*, double complex, dimension(3), intent(out) *roots*)

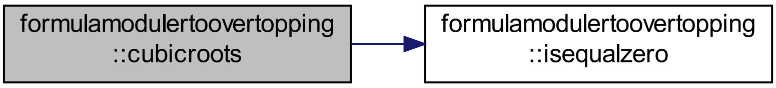
cubicRoots: calculate the roots of a cubic function

Parameters

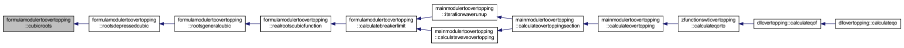
`in`	*z*	complex number
`out`	*roots*	cubic roots

Definition at line 651 of file formulaModuleRTOovertopping.f90.

Here is the call graph for this function:



Here is the caller graph for this function:



4.3.1.10 logical function, public formulamodulertoovertopping::isequalreal (real(wp), intent(in) *x1*, real(wp), intent(in) *x2*)

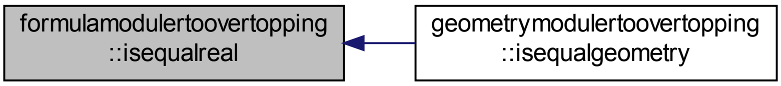
isEqualReal: are two reals (almost) equal

Parameters

`in`	*x1*	first real
`in`	*x2*	second real

Definition at line 692 of file formulaModuleRTOovertopping.f90.

Here is the caller graph for this function:



4.3.1.11 logical function, public formulamodulertoovertopping::isequalzero (real(wp), intent(in) *x*)

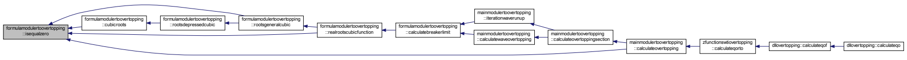
isEqualZero: is a real (almost) zero

Parameters

in	*x*	real number

Definition at line 716 of file formulaModuleRTOovertopping.f90.

Here is the caller graph for this function:


formulamodulertoovertopping ::isequalzero
formulamodulertoovertopping ::cubicroots
formulamodulertoovertopping ::rootsdepressedcubic
formulamodulertoovertopping ::rootsgeneralcubic
formulamodulertoovertopping ::realrootscubicfunction
formulamodulertoovertopping ::calculatebreakerlimit
mainmodulertoovertopping ::iterationwaverunup
mainmodulertoovertopping ::calculatewaveovertopping
mainmodulertoovertopping ::calculateovertoppingsection
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.3.1.12 subroutine, public formulamodulertoovertopping::realrootscubicfunction (real(wp), intent(in) *a,* real(wp), intent(in) *b,* real(wp), intent(in) *c,* real(wp), intent(in) *d,* integer, intent(out) *N,* real(wp), dimension(3), intent(out) *x,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

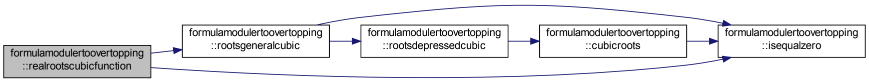
realRootsCubicFunction: calculate the roots of a cubic function

Parameters

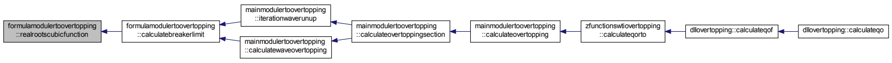
in	*d*	coefficients cubic function
out	*n*	number of real roots cubic function
out	*x*	real roots cubic function
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 503 of file formulaModuleRTOovertopping.f90.

Here is the call graph for this function:


formulamodulertoovertopping ::realrootscubicfunction
formulamodulertoovertopping ::rootsgeneralcubic
formulamodulertoovertopping ::rootsdepressedcubic
formulamodulertoovertopping ::cubicroots
formulamodulertoovertopping ::isequalzero

Here is the caller graph for this function:


formulamodulertoovertopping ::realrootscubicfunction
formulamodulertoovertopping ::calculatebreakerlimit
mainmodulertoovertopping ::iterationwaverunup
mainmodulertoovertopping ::calculatewaveovertopping
mainmodulertoovertopping ::calculateovertoppingsection
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.3.1.13 subroutine, public formulamodulertoovertopping::rootsdepressedcubic (real(wp), intent(in) *p,* real(wp), intent(in) *q,* double complex, dimension(3), intent(out) *z*)

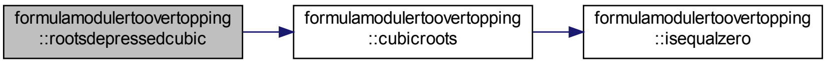
rootsDepressedCubic: calculate the roots of a depressed cubic function

Parameters

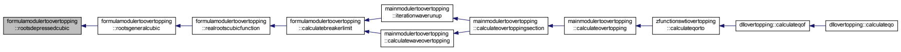
in	*q*	coefficients depressed cubic
out	*z*	roots depressed cubic

Definition at line 611 of file formulaModuleRTOovertopping.f90.

Here is the call graph for this function:


formulamodulertoovertopping ::rootsdepressedcubic
formulamodulertoovertopping ::cubicroots
formulamodulertoovertopping ::isequalzero

Here is the caller graph for this function:


formulamodulertoovertopping ::rootsdepressedcubic
formulamodulertoovertopping ::rootsgeneralcubic
formulamodulertoovertopping ::realrootscubicfunction
formulamodulertoovertopping ::calculatebreakerlimit
mainmodulertoovertopping ::iterationwaverunup
mainmodulertoovertopping ::calculatewaveovertopping
mainmodulertoovertopping ::calculateovertoppingsection
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.3.1.14 subroutine, public formulamodulertoovertopping::rootsgeneralcubic (real(wp), intent(in) *a,* real(wp), intent(in) *b,* real(wp), intent(in) *c,* real(wp), intent(in) *d,* double complex, dimension(3), intent(out) *z,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

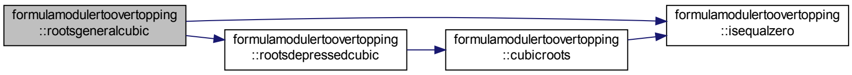
rootsGeneralCubic: calculate the roots of a generic cubic function

Parameters

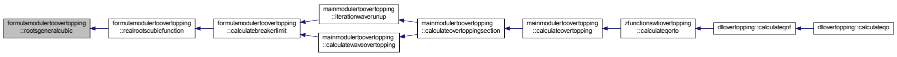
in	*d*	coefficients cubic function
out	*z*	roots cubic function
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 561 of file formulaModuleRTOovertopping.f90.

Here is the call graph for this function:


formulamodulertoovertopping ::rootsgeneralcubic
formulamodulertoovertopping ::rootsdepressedcubic
formulamodulertoovertopping ::cubicroots
formulamodulertoovertopping ::isequalzero

Here is the caller graph for this function:


formulamodulertoovertopping ::rootsgeneralcubic
formulamodulertoovertopping ::realrootscubicfunction
formulamodulertoovertopping ::calculatebreakerlimit
mainmodulertoovertopping ::iterationwaverunup
mainmodulertoovertopping ::calculatewaveovertopping
mainmodulertoovertopping ::calculateovertoppingsection
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.4 geometrymodulertoovertopping Module Reference

Functions/Subroutines

- subroutine, public checkcrosssection (psi, nCoordinates, xCoordinates, yCoordinates, roughnessFactors, succes, errorMessage) *checkCrossSection: check cross section*

- subroutine, public initializegeometry (psi, nCoordinates, xCoordinates, yCoordinates, roughnessFactors, geometry, succes, errorMessage)

  initializeGeometry: initialize the geometry
- subroutine, public allocatevectorsgeometry (nCoordinates, geometry)

  allocateVectorsGeometry: allocate the geometry vectors
- subroutine, public deallocategeometry (geometry)

  deallocateGeometry: deallocate the geometry vectors
- subroutine, public calculatesegmentslopes (geometry, succes, errorMessage)

  calculateSegmentSlopes: calculate the segment slopes
- subroutine, public determinesegmenttypes (geometry)

  determineSegmentTypes: determine the segment types
- subroutine, public copygeometry (geometry, geometryCopy)

  copyGeometry: copy a geometry structure
- subroutine, public isequalgeometry (geometry1, geometry2, succes, errorMessage)

  isEqualGeometry: are two geometries equal
- subroutine, public mergesequentialberms (geometry, geometryMergedBerms, succes, errorMessage)

  mergeSequentialBerms: merge sequential berms
- subroutine, public adjustnonhorizontalberms (geometry, geometryFlatBerms, succes, errorMessage)

  adjustNonHorizontalBerms: adjust non-horizontal berms
- subroutine, public removeberms (geometry, geometryNoBerms, succes, errorMessage)

  removeBerms: remove berms
- subroutine, public removedikesegments (geometry, index, geometryAdjusted, succes, errorMessage)

  removeDikeSegments: remove dike segments
- subroutine, public splitcrosssection (geometry, L0, NwideBerms, geometrysectionB, geometrysectionF, succes, errorMessage)

  splitCrossSection: split a cross section
- subroutine, public calculatehorzlengths (geometry, yLower, yUpper, horzLengths, succes, errorMessage)

  calculateHorzLengths: calculate horizontal lengths
- subroutine, public calculatehorzdistance (geometry, yLower, yUpper, dx, succes, errorMessage)

  calculateHorzDistance: calculate horizontal distance
- subroutine, public writecrosssection (geometry, geometryName)

  writeCrossSection: write a cross section

4.4.1 Function/Subroutine Documentation

4.4.1.1 subroutine, public geometrymodulertoovertopping::adjustnonhorizontalberms (type (tpgeometry), intent(in) *geometry,* type (tpgeometry), intent(out) *geometryFlatBerms,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

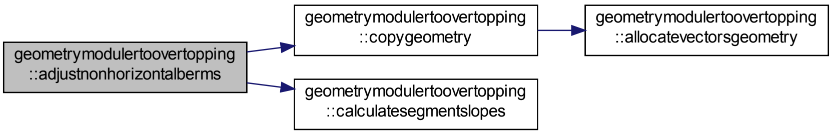
adjustNonHorizontalBerms: adjust non-horizontal berms

Parameters

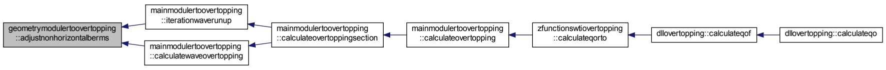
in	*geometry*	structure with geometry data
out	*geometryflatberms*	geometry data with horizontal berms
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 578 of file geometryModuleRTOovertopping.f90.

Here is the call graph for this function:


geometrymodulertoovertopping
::adjustnonhorizontalberms
geometrymodulertoovertopping
::copygeometry
geometrymodulertoovertopping
::allocatevectorsgeometry
geometrymodulertoovertopping
::calculatesegmentslopes

Here is the caller graph for this function:



4.4.1.2 subroutine, public geometrymodulertoovertopping::allocatevectorsgeometry (integer, intent(in) *nCoordinates,* type (tpgeometry), intent(inout) *geometry*)

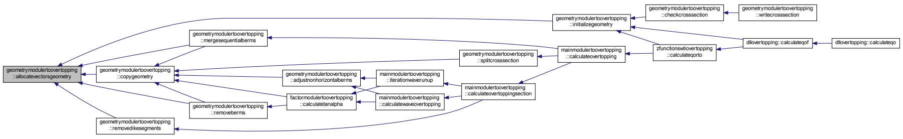
allocateVectorsGeometry: allocate the geometry vectors

Parameters

`in`	*ncoordinates*	number of coordinates
`in,out`	*geometry*	structure with geometry data

Definition at line 199 of file geometryModuleRTOovertopping.f90.

Here is the caller graph for this function:


geometrymodulertoovertopping
::allocatevectorsgeometry
geometrymodulertoovertopping
::copygeometry
geometrymodulertoovertopping
::mergesequentialberms
geometrymodulertoovertopping
::removeberms
geometrymodulertoovertopping
::removedikesegments
geometrymodulertoovertopping
::adjustnonhorizontalberms
factormodulertoovertopping
::calculatetanalpha
mainmodulertoovertopping
::iterationwaverunup
mainmodulertoovertopping
::calculatewaveovertopping
geometrymodulertoovertopping
::splitcrosssection
mainmodulertoovertopping
::calculateovertoppingsection
geometrymodulertoovertopping
::initializegeometry
mainmodulertoovertopping
::calculateovertopping
zfunctionswtiovertopping
::calculateqorto
geometrymodulertoovertopping
::checkcrosssection
geometrymodulertoovertopping
::writecrosssection
dllovertopping::calculateqof
dllovertopping::calculateqo

4.4.1.3 subroutine, public geometrymodulertoovertopping::calculatehorzdistance (type (tpgeometry), intent(in) *geometry,* real(wp), intent(in) *yLower,* real(wp), intent(in) *yUpper,* real(wp), intent(out) *dx,* logical, intent(out) *succes,* character(len=$*$), intent(out) *errorMessage*)

calculateHorzDistance: calculate horizontal distance

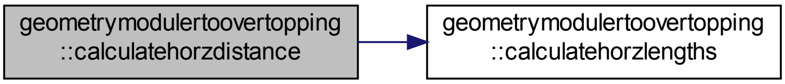
Parameters

`in`	*geometry*	structure with geometry data
`in`	*ylower*	y-coordinate lower bound (m+NAP)
`in`	*yupper*	y-coordinate upper bound (m+NAP)
`out`	*dx*	horizontal distance between bounds (m)

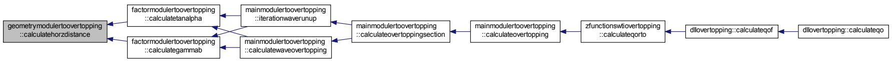
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 1023 of file geometryModuleRTOovertopping.f90.

Here is the call graph for this function:



Here is the caller graph for this function:



4.4.1.4 subroutine, public geometrymodulertoovertopping::calculatehorzlengths (type (tpgeometry), intent(in) *geometry,* real(wp), intent(in) *yLower,* real(wp), intent(in) *yUpper,* real(wp), dimension(geometry%ncoordinates-1), intent(out) *horzLengths,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

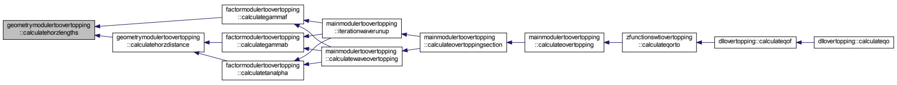
calculateHorzLengths: calculate horizontal lengths

Parameters

in	*geometry*	structure with geometry data
in	*ylower*	y-coord. lower bound (m+NAP)
in	*yupper*	y-coord. upper bound (m+NAP)
out	*horzlengths*	horizontal lengths segments (m)
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 927 of file geometryModuleRTOovertopping.f90.

Here is the caller graph for this function:



4.4.1.5 subroutine, public geometrymodulertoovertopping::calculatesegmentslopes (type (tpgeometry), intent(inout) *geometry,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

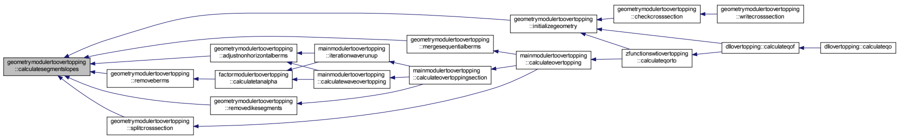
calculateSegmentSlopes: calculate the segment slopes

Parameters

`in,out`	*geometry*	structure with geometry data
`out`	*succes*	flag for succes
`out`	*errormessage*	error message

Definition at line 248 of file geometryModuleRTOovertopping.f90.

Here is the caller graph for this function:


geometrymodulertoovertopping ::calculatesegmentslopes
geometrymodulertoovertopping ::initializegeometry
geometrymodulertoovertopping ::checkcrosssection
geometrymodulertoovertopping ::writecrosssection
geometrymodulertoovertopping ::mergesequentialberms
dllovertopping::calculateqof
dllovertopping::calculateqo
geometrymodulertoovertopping ::adjustnonhorizontalberms
mainmodulertoovertopping ::iterationwaverunup
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
geometrymodulertoovertopping ::removeberms
factormodulertoovertopping ::calculatetanalpha
mainmodulertoovertopping ::calculatewaveovertopping
mainmodulertoovertopping ::calculateovertoppingsection
geometrymodulertoovertopping ::removedikesegments
geometrymodulertoovertopping ::splitcrosssection

4.4.1.6 subroutine, public geometrymodulertoovertopping::checkcrosssection (real(wp), intent(in) *psi,* integer, intent(in) *nCoordinates,* real(wp), dimension (ncoordinates), intent(in) *xCoordinates,* real(wp), dimension (ncoordinates), intent(in) *yCoordinates,* real(wp), dimension(ncoordinates-1), intent(in) *roughnessFactors,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

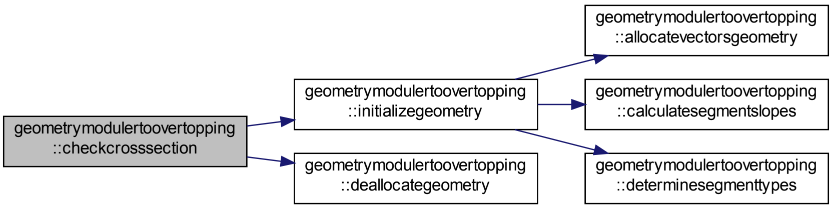
checkCrossSection: check cross section

Parameters

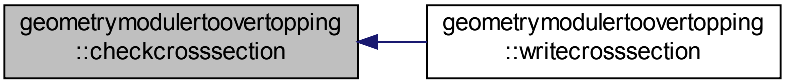
`in`	*psi*	dike normal (degree)
`in`	*ncoordinates*	number of coordinates
`in`	*xcoordinates*	x-coordinates (m)
`in`	*ycoordinates*	y-coordinates (m+NAP)
`in`	*roughnessfactors*	roughness factors
`out`	*succes*	flag for succes
`out`	*errormessage*	error message

Definition at line 34 of file geometryModuleRTOovertopping.f90.

Here is the call graph for this function:


geometrymodulertoovertopping ::checkcrosssection
geometrymodulertoovertopping ::initializegeometry
geometrymodulertoovertopping ::deallocategeometry
geometrymodulertoovertopping ::allocatevectorsgeometry
geometrymodulertoovertopping ::calculatesegmentslopes
geometrymodulertoovertopping ::determinesegmenttypes

Here is the caller graph for this function:



4.4.1.7 subroutine, public geometrymodulertoovertopping::copygeometry (type (tpgeometry), intent(in) *geometry,* type (tpgeometry), intent(inout) *geometryCopy*)

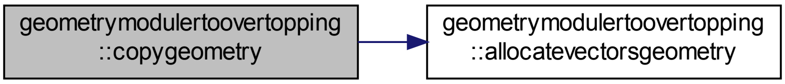
copyGeometry: copy a geometry structure

Parameters

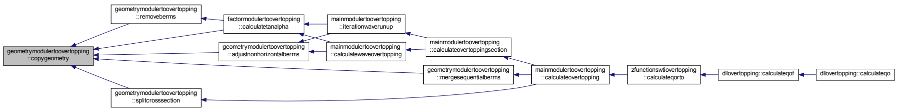
`in`	*geometry*	structure with geometry data
`in,out`	*geometrycopy*	structure with geometry data copy

Definition at line 330 of file geometryModuleRTOovertopping.f90.

Here is the call graph for this function:


geometrymodulertoovertopping
::copygeometry
geometrymodulertoovertopping
::allocatevectorsgeometry

Here is the caller graph for this function:


geometrymodulertoovertopping
::copygeometry
geometrymodulertoovertopping
::removeberms
factormodulertoovertopping
::calculatetanalpha
mainmodulertoovertopping
::iterationwaverunup
geometrymodulertoovertopping
::adjustnonhorizontalberms
mainmodulertoovertopping
::calculatewaveovertopping
mainmodulertoovertopping
::calculateovertoppingsection
geometrymodulertoovertopping
::mergesequentialberms
mainmodulertoovertopping
::calculateovertopping
zfunctionswtiovertopping
::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo
geometrymodulertoovertopping
::splitcrosssection

4.4.1.8 subroutine, public geometrymodulertoovertopping::deallocategeometry (type (tpgeometry), intent(inout) *geometry*)

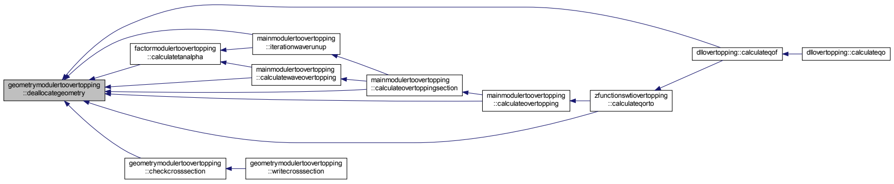
deallocateGeometry: deallocate the geometry vectors

Parameters

`in,out`	*geometry*	structure with geometry data

Definition at line 225 of file geometryModuleRTOovertopping.f90.

Here is the caller graph for this function:



4.4.1.9 subroutine, public geometrymodulertoovertopping::determinesegmenttypes (type (tpgeometry), intent(inout) *geometry*)

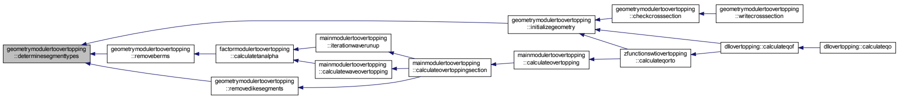
determineSegmentTypes: determine the segment types

Parameters

in,out	*geometry*	structure with geometry data

Definition at line 287 of file geometryModuleRTOovertopping.f90.

Here is the caller graph for this function:



4.4.1.10 subroutine, public geometrymodulertoovertopping::initializegeometry (real(wp), intent(in) *psi,* integer, intent(in) *nCoordinates,* real(wp), dimension (ncoordinates), intent(in) *xCoordinates,* real(wp), dimension (ncoordinates), intent(in) *yCoordinates,* real(wp), dimension(ncoordinates-1), intent(in) *roughnessFactors,* type (tpgeometry), intent(out) *geometry,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

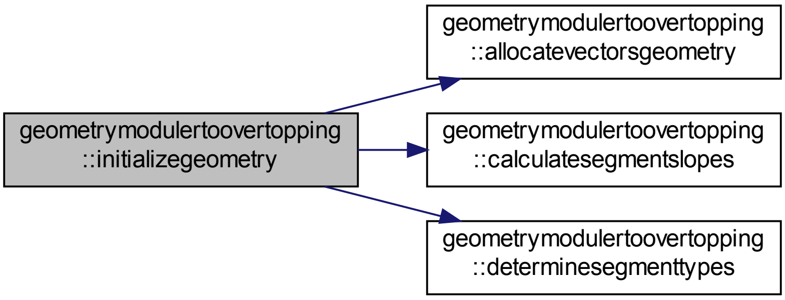
initializeGeometry: initialize the geometry

Parameters

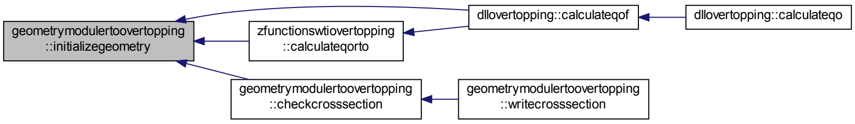
in	*psi*	dike normal (degree)
in	*ncoordinates*	number of coordinates
in	*xcoordinates*	x-coordinates (m)
in	*ycoordinates*	y-coordinates (m+NAP)
in	*roughnessfactors*	roughness factors
out	*geometry*	structure with geometry data
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 142 of file geometryModuleRTOovertopping.f90.

Here is the call graph for this function:


geometrymodulertoovertopping ::initializegeometry
geometrymodulertoovertopping ::allocatevectorsgeometry
geometrymodulertoovertopping ::calculatesegmentslopes
geometrymodulertoovertopping ::determinesegmenttypes

Here is the caller graph for this function:


geometrymodulertoovertopping ::initializegeometry
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo
geometrymodulertoovertopping ::checkcrosssection
geometrymodulertoovertopping ::writecrosssection

4.4.1.11 subroutine, public geometrymodulertoovertopping::isequalgeometry (type (tpgeometry), intent(in) *geometry1,* type (tpgeometry), intent(in) *geometry2,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

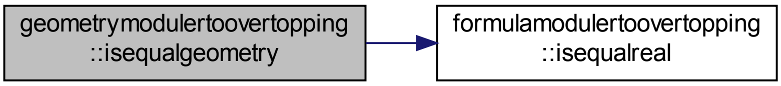
isEqualGeometry: are two geometries equal

Parameters

in	*geometry1*	structure with geometry data 1
in	*geometry2*	structure with geometry data 2
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 377 of file geometryModuleRTOovertopping.f90.

Here is the call graph for this function:


geometrymodulertoovertopping ::isequalgeometry
formulamodulertoovertopping ::isequalreal

4.4.1.12 subroutine, public geometrymodulertoovertopping::mergesequentialberms (type (tpgeometry), intent(in) *geometry,* type (tpgeometry), intent(inout) *geometryMergedBerms,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

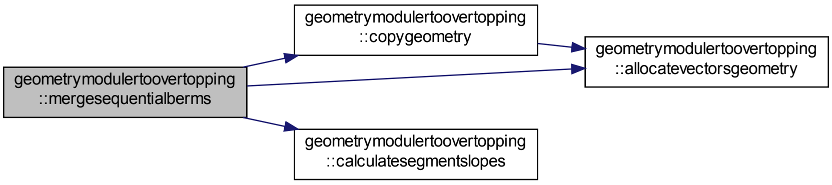
mergeSequentialBerms: merge sequential berms

Parameters

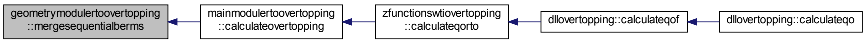
`in`	*geometry*	structure with geometry data
`in,out`	*geometrymergedberms*	geometry data with merged sequential berms
`out`	*succes*	flag for succes
`out`	*errormessage*	error message

Definition at line 471 of file geometryModuleRTOovertopping.f90.

Here is the call graph for this function:


geometrymodulertoovertopping ::mergesequentialberms
geometrymodulertoovertopping ::copygeometry
geometrymodulertoovertopping ::allocatevectorsgeometry
geometrymodulertoovertopping ::calculatesegmentslopes

Here is the caller graph for this function:


geometrymodulertoovertopping ::mergesequentialberms
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.4.1.13 subroutine, public geometrymodulertoovertopping::removeberms (type (tpgeometry), intent(in) *geometry,* type (tpgeometry), intent(out) *geometryNoBerms,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

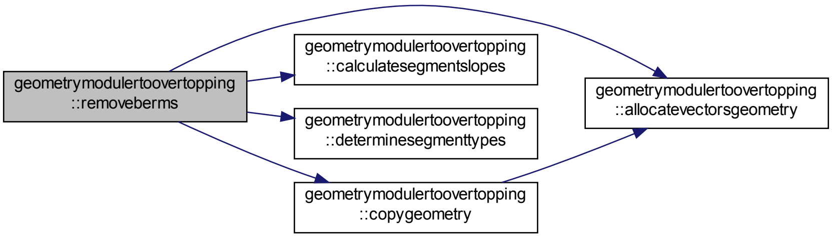
removeBerms: remove berms

Parameters

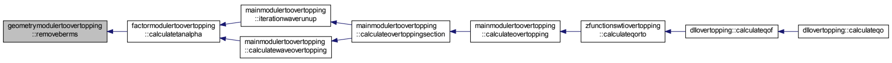
`in`	*geometry*	structure with geometry data
`out`	*geometrynoberms*	geometry data withouth berms
`out`	*succes*	flag for succes
`out`	*errormessage*	error message

Definition at line 664 of file geometryModuleRTOovertopping.f90.

Here is the call graph for this function:


geometrymodulertoovertopping ::removeberms
geometrymodulertoovertopping ::calculatesegmentslopes
geometrymodulertoovertopping ::determinesegmenttypes
geometrymodulertoovertopping ::copygeometry
geometrymodulertoovertopping ::allocatevectorsgeometry

Here is the caller graph for this function:


geometrymodulertoovertopping ::removeberms
factormodulertoovertopping ::calculatetanalpha
mainmodulertoovertopping ::iterationwaverunup
mainmodulertoovertopping ::calculatewaveovertopping
mainmodulertoovertopping ::calculateovertoppingsection
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.4.1.14 subroutine, public geometrymodulertoovertopping::removedikesegments (type (tpgeometry), intent(in) *geometry,* integer, intent(in) *index,* type (tpgeometry), intent(out) *geometryAdjusted,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

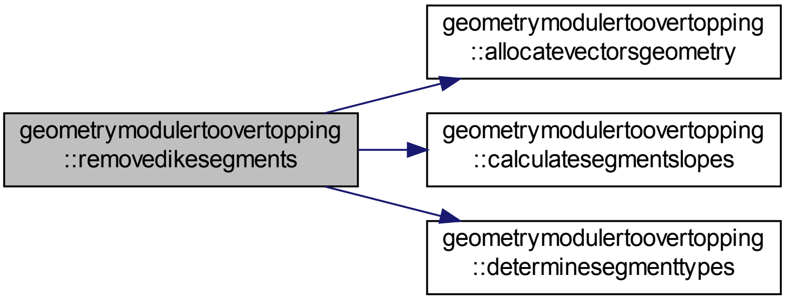
removeDikeSegments: remove dike segments

Parameters

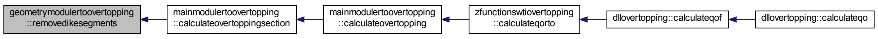
in	*geometry*	structure with geometry data
in	*index*	index starting point new cross section
out	*geometryadjusted*	geometry data with removed dike segments
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 761 of file geometryModuleRTOovertopping.f90.

Here is the call graph for this function:


geometrymodulertoovertopping ::removedikesegments
geometrymodulertoovertopping ::allocatevectorsgeometry
geometrymodulertoovertopping ::calculatesegmentslopes
geometrymodulertoovertopping ::determinesegmenttypes

Here is the caller graph for this function:


geometrymodulertoovertopping ::removedikesegments
mainmodulertoovertopping ::calculateovertoppingsection
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.4.1.15 subroutine, public geometrymodulertoovertopping::splitcrosssection (type (tpgeometry), intent(in) *geometry,* real(wp), intent(in) *L0,* integer, intent(out) *NwideBerms,* type (tpgeometry), intent(out) *geometrysectionB,* type (tpgeometry), intent(out) *geometrysectionF,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

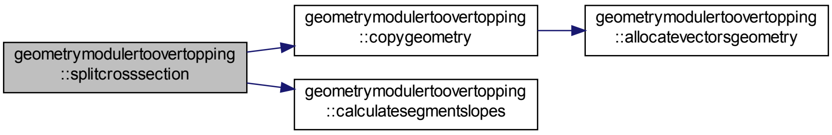
splitCrossSection: split a cross section

Parameters

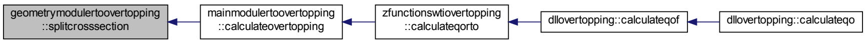
in	*geometry*	structure with geometry data
in	*l0*	wave length (m)
out	*nwideberms*	number of wide berms
out	*geometrysectionb*	geometry data with wide berms to ordinary berms
out	*geometrysectionf*	geometry data with wide berms to foreshores
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 825 of file geometryModuleRTOovertopping.f90.

Here is the call graph for this function:


geometrymodulertoovertopping ::splitcrosssection
geometrymodulertoovertopping ::copygeometry
geometrymodulertoovertopping ::allocatevectorsgeometry
geometrymodulertoovertopping ::calculatesegmentslopes

Here is the caller graph for this function:


geometrymodulertoovertopping ::splitcrosssection
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.4.1.16 subroutine, public geometrymodulertoovertopping::writecrosssection (type (tpgeometry), intent(in) *geometry,* character(len=∗), intent(in) *geometryName*)

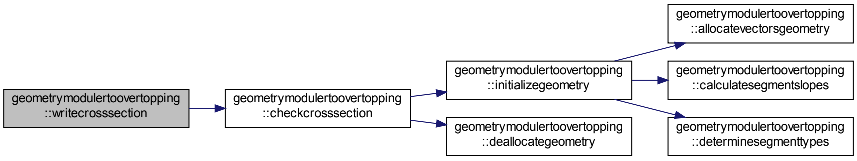
writeCrossSection: write a cross section

Parameters

in	*geometry*	structure with geometry data
in	*geometryname*	description of geometry data

Definition at line 1066 of file geometryModuleRTOovertopping.f90.

Here is the call graph for this function:


geometrymodulertoovertopping ::writecrosssection
geometrymodulertoovertopping ::checkcrosssection
geometrymodulertoovertopping ::initializegeometry
geometrymodulertoovertopping ::deallocategeometry
geometrymodulertoovertopping ::allocatevectorsgeometry
geometrymodulertoovertopping ::calculatesegmentslopes
geometrymodulertoovertopping ::determinesegmenttypes

4.5 mainmodulertoovertopping Module Reference

Functions/Subroutines

- subroutine, public calculateovertopping (geometry, load, modelFactors, probContext, overtopping, succes, errorMessage)

  calculateOvertopping: calculate the overtopping

- subroutine, public calculateovertoppingsection (geometry, h, Hm0, Tm_10, L0, gammaBeta_z, gammaBeta↩_o, modelFactors, overtopping, succes, errorMessage)

  calculateOvertoppingSection: calculate the overtopping for a section

- subroutine, public iterationwaverunup (geometry, h, Hm0, Tm_10, L0, gammaBeta_z, modelFactors, z2, succes, errorMessage)

  iterationWaveRunup: iteration for the wave runup

- subroutine, public calculatewaveovertopping (geometry, h, Hm0, Tm_10, z2, gammaBeta_o, modelFactors, Qo, succes, errorMessage)

  calculateWaveOvertopping: calculate wave overtopping

- subroutine calculateovertoppingnegativefreeboard (load, geometry, overtopping, succes, errorMessage)

  calculateOvertoppingNegativeFreeboard: calculate overtopping in case of negative freeboard

- subroutine, public interpolateresultssections (geometry, L0, NwideBerms, overtoppingB, overtoppingF, overtopping, succes, errorMessage)

  interpolateResultsSections: interpolate results for split cross sections

- subroutine, public checkinputdata (geometry, load, modelFactors, probContext, succes, errorMessage)

  checkInputdata: check the input data

- subroutine, public convertmodelfactorsrto (modelFactorsF, modelFactorsC)

  convertModelfactorsRTO: convert the model factors

- subroutine, public fillmodelfactorsrto (modelFactors, fn, fb, frunup1, frunup2, frunup3, fshallow)

  fillModelfactorsRTO: fill the model factors

4.5.1 Function/Subroutine Documentation

4.5.1.1 subroutine, public mainmodulertoovertopping::calculateovertopping (type (tpgeometry), intent(in) *geometry,* type (tpload), intent(in) *load,* type (tpmodelfactors), intent(in) *modelFactors,* logical, intent(in) *probContext,* type (tpovertopping), intent(out) *overtopping,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

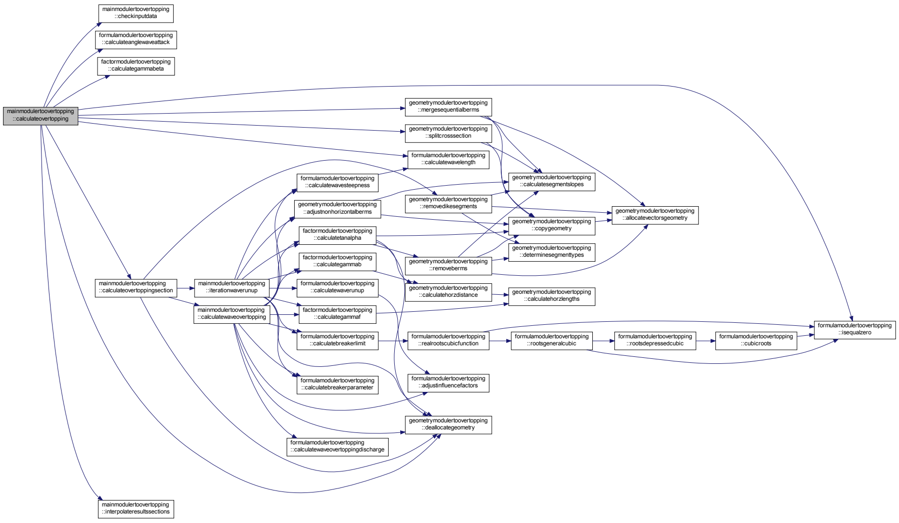
calculateOvertopping: calculate the overtopping

Parameters

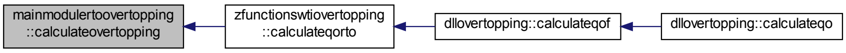
in	*geometry*	structure with geometry data
in	*load*	structure with load parameters
in	*modelfactors*	structure with model factors
in	*probcontext*	flag for using the overtopping module in a probabilistic context
out	*overtopping*	structure with overtopping results
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 35 of file mainModuleRTOovertopping.f90.

Here is the call graph for this function:


mainmodulertoovertopping ::calculateovertopping
mainmodulertoovertopping ::checkinputdata
formulamodulertoovertopping ::calculateanglewaveattack
factormodulertoovertopping ::calculategammabeta
geometrymodulertoovertopping ::mergesequentialberms
geometrymodulertoovertopping ::splitcrosssection
formulamodulertoovertopping ::calculatewavelength
formulamodulertoovertopping ::calculatewavesteepness
geometrymodulertoovertopping ::adjustnonhorizontalberms
factormodulertoovertopping ::calculatetanalpha
factormodulertoovertopping ::calculategammab
mainmodulertoovertopping ::calculateovertoppingsection
mainmodulertoovertopping ::iterationwaverunup
mainmodulertoovertopping ::calculatewaveovertopping
formulamodulertoovertopping ::calculatewaverunup
factormodulertoovertopping ::calculategammaf
formulamodulertoovertopping ::calculatebreakerlimit
formulamodulertoovertopping ::calculatebreakerparameter
formulamodulertoovertopping ::calculatewaveovertoppingdischarge
geometrymodulertoovertopping ::removedikesegments
geometrymodulertoovertopping ::removeberms
geometrymodulertoovertopping ::calculatehorzdistance
formulamodulertoovertopping ::realrootscubicfunction
formulamodulertoovertopping ::adjustinfluencefactors
geometrymodulertoovertopping ::deallocategeometry
geometrymodulertoovertopping ::calculatesegmentslopes
geometrymodulertoovertopping ::copygeometry
geometrymodulertoovertopping ::determinesegmenttypes
geometrymodulertoovertopping ::calculatehorzlengths
geometrymodulertoovertopping ::allocatevectorsgeometry
formulamodulertoovertopping ::rootsgeneralcubic
formulamodulertoovertopping ::rootsdepressedcubic
formulamodulertoovertopping ::cubicroots
formulamodulertoovertopping ::isequalzero
mainmodulertoovertopping ::interpolateresultssections

Here is the caller graph for this function:


mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.5.1.2 subroutine mainmodulertoovertopping::calculateovertoppingnegativefreeboard (type (tpload), intent(in) *load*, type (tpgeometry), intent(in) *geometry*, type (tpovertopping), intent(inout) *overtopping*, logical, intent(out) *succes*, character(len=∗), intent(out) *errorMessage*) [private]

calculateOvertoppingNegativeFreeboard: calculate overtopping in case of negative freeboard

Parameters

in	*geometry*	structure with geometry data
in	*load*	structure with load parameters
in,out	*overtopping*	structure with overtopping results
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 693 of file mainModuleRTOovertopping.f90.

4.5.1.3 subroutine, public mainmodulertoovertopping::calculateovertoppingsection (type (tpgeometry), intent(in) *geometry*, real(wp), intent(in) *h*, real(wp), intent(in) *Hm0*, real(wp), intent(in) *Tm_10*, real(wp), intent(in) *L0*, real(wp), intent(inout) *gammaBeta_z*, real(wp), intent(inout) *gammaBeta_o*, type (tpmodelfactors), intent(in) *modelFactors*, type (tpovertopping), intent(out) *overtopping*, logical, intent(out) *succes*, character(len=∗), intent(out) *errorMessage*)

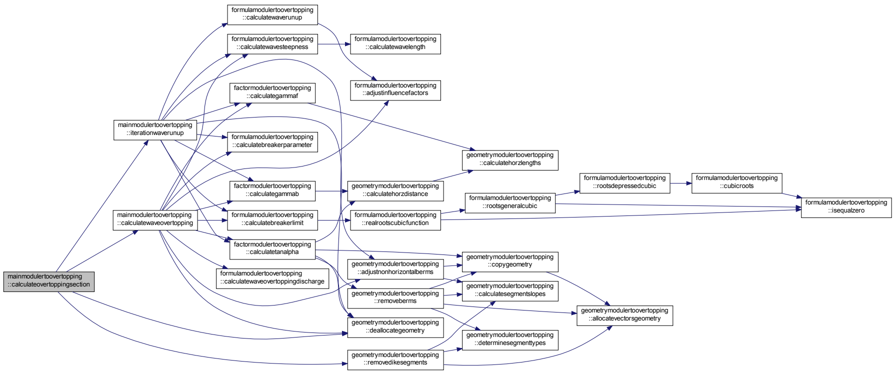
calculateOvertoppingSection: calculate the overtopping for a section

Parameters

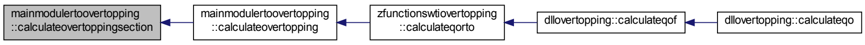
in	*geometry*	structure with geometry data
in	*h*	local water level (m+NAP)
in	*hm0*	significant wave height (m)
in	*tm_10*	spectral wave period (s)
in	*l0*	wave length (m)
in,out	*gammabeta_z*	influence angle wave attack wave run-up
in,out	*gammabeta_o*	influence angle wave attack overtopping
in	*modelfactors*	structure with model factors
out	*overtopping*	structure with overtopping results
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 167 of file mainModuleRTOovertopping.f90.

Here is the call graph for this function:


formulamodulertoovertopping ::calculatewaverunup
formulamodulertoovertopping ::calculatewavesteepness
formulamodulertoovertopping ::calculatewavelength
factormodulertoovertopping ::calculategammaf
formulamodulertoovertopping ::adjustinfluencefactors
mainmodulertoovertopping ::iterationwaverunup
formulamodulertoovertopping ::calculatebreakerparameter
geometrymodulertoovertopping ::calculatehorzlengths
formulamodulertoovertopping ::rootsdepressedcubic
formulamodulertoovertopping ::cubicroots
factormodulertoovertopping ::calculategammab
geometrymodulertoovertopping ::calculatehorzdistance
formulamodulertoovertopping ::rootsgeneralcubic
formulamodulertoovertopping ::isequalzero
mainmodulertoovertopping ::calculatewaveovertopping
formulamodulertoovertopping ::calculatebreakerlimit
formulamodulertoovertopping ::realrootscubicfunction
factormodulertoovertopping ::calculatetanalpha
geometrymodulertoovertopping ::copygeometry
mainmodulertoovertopping ::calculateovertoppingsection
formulamodulertoovertopping ::calculatewaveovertoppingdischarge
geometrymodulertoovertopping ::adjustnonhorizontalberms
geometrymodulertoovertopping ::calculatesegmentslopes
geometrymodulertoovertopping ::removeberms
geometrymodulertoovertopping ::allocatevectorsgeometry
geometrymodulertoovertopping ::deallocategeometry
geometrymodulertoovertopping ::determinesegmenttypes
geometrymodulertoovertopping ::removedikesegments

Here is the caller graph for this function:


mainmodulertoovertopping ::calculateovertoppingsection
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.5.1.4 subroutine, public mainmodulertoovertopping::calculatewaveovertopping (type (tpgeometry), intent(in) *geometry,* real(wp), intent(in) *h,* real(wp), intent(in) *Hm0,* real(wp), intent(in) *Tm_10,* real(wp), intent(in) *z2,* real(wp), intent(inout) *gammaBeta_o,* type (tpmodelfactors), intent(in) *modelFactors,* real(wp), intent(out) *Qo,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

calculateWaveOvertopping: calculate wave overtopping

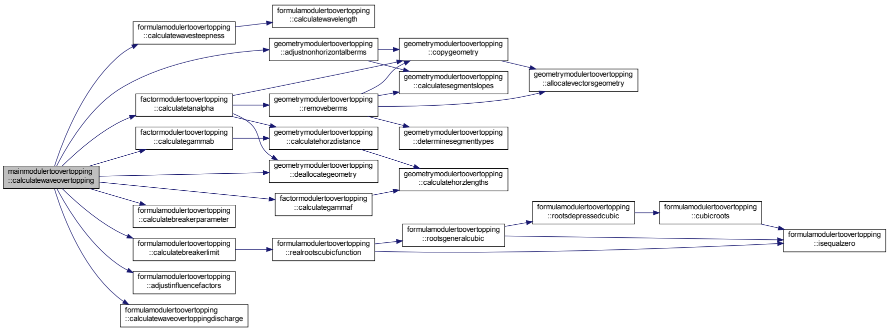
Parameters

in	*geometry*	structure with geometry data
in	*h*	local water level (m+NAP)
in	*hm0*	significant wave height (m)
in	*tm_10*	spectral wave period (s)

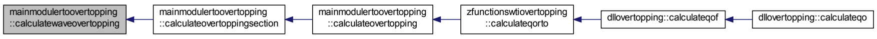
in	*z2*	2% wave run-up (m)
in,out	*gammabeta_o*	influence angle wave attack overtopping
in	*modelfactors*	structure with model factors
out	*qo*	wave overtopping discharge (m3/m per s)
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 605 of file mainModuleRTOovertopping.f90.

Here is the call graph for this function:


mainmodulertoovertopping ::calculatewaveovertopping
formulamodulertoovertopping ::calculatewavesteepness
formulamodulertoovertopping ::calculatewavelength
geometrymodulertoovertopping ::adjustnonhorizontalberms
geometrymodulertoovertopping ::copygeometry
geometrymodulertoovertopping ::calculatesegmentslopes
geometrymodulertoovertopping ::allocatevectorsgeometry
factormodulertoovertopping ::calculatetanalpha
geometrymodulertoovertopping ::removeberms
factormodulertoovertopping ::calculategammab
geometrymodulertoovertopping ::calculatehorzdistance
geometrymodulertoovertopping ::determinesegmenttypes
geometrymodulertoovertopping ::deallocategeometry
geometrymodulertoovertopping ::calculatehorzlengths
factormodulertoovertopping ::calculategammaf
formulamodulertoovertopping ::calculatebreakerparameter
formulamodulertoovertopping ::rootsdepressedcubic
formulamodulertoovertopping ::cubicroots
formulamodulertoovertopping ::rootsgeneralcubic
formulamodulertoovertopping ::isequalzero
formulamodulertoovertopping ::calculatebreakerlimit
formulamodulertoovertopping ::realrootscubicfunction
formulamodulertoovertopping ::adjustinfluencefactors
formulamodulertoovertopping ::calculatewaveovertoppingdischarge

Here is the caller graph for this function:


mainmodulertoovertopping ::calculatewaveovertopping
mainmodulertoovertopping ::calculateovertoppingsection
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.5.1.5 subroutine, public mainmodulertoovertopping::checkinputdata (type (tpgeometry), intent(in) *geometry,* type (tpload), intent(in) *load,* type (tpmodelfactors), intent(in) *modelFactors,* logical, intent(in) *probContext,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

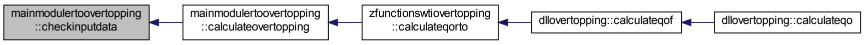
checkInputdata: check the input data

Parameters

in	*geometry*	structure with geometry data
in	*load*	structure with load parameters
in	*modelfactors*	structure with model factors
in	*probcontext*	flag for using the overtopping module in a probabilistic context
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 810 of file mainModuleRTOovertopping.f90.

Here is the caller graph for this function:


mainmodulertoovertopping ::checkinputdata
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.5.1.6 subroutine, public mainmodulertoovertopping::convertmodelfactorsrto (type (tpmodelfactors), intent(out) *modelFactorsF*, type (overtoppingmodelfactors), intent(in) *modelFactorsC*)

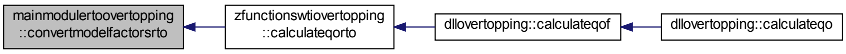
convertModelfactorsRTO: convert the model factors

Parameters

in	*modelfactorsc*	model factors in C-like structure
out	*modelfactorsf*	model factors in Fortran like structure

Definition at line 927 of file mainModuleRTOovertopping.f90.

Here is the caller graph for this function:


mainmodulertoovertopping ::convertmodelfactorsrto
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.5.1.7 subroutine, public mainmodulertoovertopping::fillmodelfactorsrto (type (tpmodelfactors), intent(out) *modelFactors,* real (kind=wp), intent(in) *fn,* real (kind=wp), intent(in) *fb,* real (kind=wp), intent(in) *frunup1,* real (kind=wp), intent(in) *frunup2,* real (kind=wp), intent(in) *frunup3,* real (kind=wp), intent(in) *fshallow*)

fillModelfactorsRTO: fill the model factors

Parameters

in	*fb*	Model factor for breaking waves
in	*fn*	Model factor for non-breaking waves
in	*frunup1*	Model factor for wave run-up 1
in	*frunup2*	Model factor for wave run-up 2
in	*frunup3*	Model factor for wave run-up 3
in	*fshallow*	Model factor for shallow waves
out	*modelfactors*	struct with model factors

Definition at line 947 of file mainModuleRTOovertopping.f90.

4.5.1.8 subroutine, public mainmodulertoovertopping::interpolateresultssections (type (tpgeometry), intent(in) *geometry,* real(wp), intent(in) *L0,* integer, intent(in) *NwideBerms,* type (tpovertopping), intent(in) *overtoppingB,* type (tpovertopping), intent(in) *overtoppingF,* type (tpovertopping), intent(out) *overtopping,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

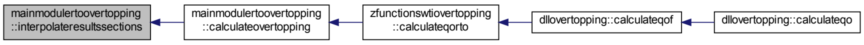
interpolateResultsSections: interpolate results for split cross sections

Parameters

in	*geometry*	structure with geometry data
in	*l0*	wave length (m)
in	*nwideberms*	number of wide berms
in	*overtoppingb*	structure with overtopping results ordinary berms
in	*overtoppingf*	structure with overtopping results foreshores
out	*overtopping*	structure with combined overtopping results
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 729 of file mainModuleRTOovertopping.f90.

Here is the caller graph for this function:


mainmodulertoovertopping ::interpolateresultssections
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.5.1.9 subroutine, public mainmodulertoovertopping::iterationwaverunup (type (tpgeometry), intent(in) *geometry,* real(wp), intent(in) *h,* real(wp), intent(in) *Hm0,* real(wp), intent(in) *Tm_10,* real(wp), intent(in) *L0,* real(wp), intent(inout) *gammaBeta_z,* type (tpmodelfactors), intent(in) *modelFactors,* real(wp), intent(out) *z2,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

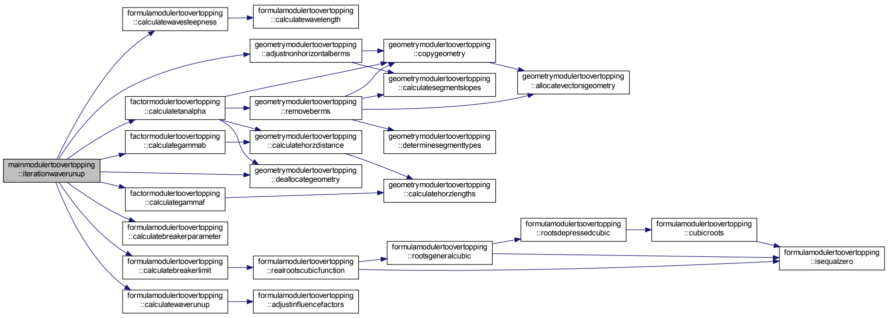
iterationWaveRunup: iteration for the wave runup

Parameters

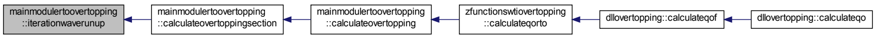
in	*geometry*	structure with geometry data
in	*h*	local water level (m+NAP)
in	*hm0*	significant wave height (m)
in	*tm_10*	spectral wave period (s)
in	*l0*	wave length (m)
in,out	*gammabeta_z*	influence factor angle wave attack 2% run-up
in	*modelfactors*	structure with model factors
out	*z2*	2% wave run-up (m)
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 391 of file mainModuleRTOovertopping.f90.

Here is the call graph for this function:


formulamodulertoovertopping ::calculatewavesteepness
formulamodulertoovertopping ::calculatewavelength
geometrymodulertoovertopping ::adjustnonhorizontalberms
geometrymodulertoovertopping ::copygeometry
geometrymodulertoovertopping ::calculatesegmentslopes
geometrymodulertoovertopping ::allocatevectorsgeometry
factormodulertoovertopping ::calculatetanalpha
geometrymodulertoovertopping ::removeberms
factormodulertoovertopping ::calculategammab
geometrymodulertoovertopping ::calculatehorzdistance
geometrymodulertoovertopping ::determinesegmenttypes
mainmodulertoovertopping ::iterationwaverunup
geometrymodulertoovertopping ::deallocategeometry
factormodulertoovertopping ::calculategammaf
geometrymodulertoovertopping ::calculatehorzlengths
formulamodulertoovertopping ::calculatebreakerparameter
formulamodulertoovertopping ::rootsdepressedcubic
formulamodulertoovertopping ::cubicroots
formulamodulertoovertopping ::rootsgeneralcubic
formulamodulertoovertopping ::isequalzero
formulamodulertoovertopping ::calculatebreakerlimit
formulamodulertoovertopping ::realrootscubicfunction
formulamodulertoovertopping ::calculatewaverunup
formulamodulertoovertopping ::adjustinfluencefactors

Here is the caller graph for this function:


mainmodulertoovertopping ::iterationwaverunup
mainmodulertoovertopping ::calculateovertoppingsection
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.6 overtoppinginterface Module Reference

Data Types

- type overtoppinggeometrytype
- type overtoppinggeometrytypef
- type tpprofilecoordinate

Variables

- integer, parameter, public varmodelfactorcriticalovertopping = 8

  Model factor critical overtopping.

4.6.1 Variable Documentation

4.6.1.1 integer, parameter, public overtoppinginterface::varmodelfactorcriticalovertopping = 8

Model factor critical overtopping.

Definition at line 17 of file overtoppingInterface.f90.

4.7 typedefinitionsrtoovertopping Module Reference

Data Types

- type overtoppingmodelfactors

  OvertoppingModelFactors: C-structure with model factors.
- type tpgeometry

  tpGeometry: structure with geometry data
- type tpload

  tpLoad: structure with load parameters
- type tpmodelfactors

  tpModelFactors: structure with model factors
- type tpovertopping

  tpOvertopping: structure with overtopping results

Variables

- real(wp), parameter xdiff_min = 2.0d-2

  minimal value distance between x-coordinates (m)
- real(wp), parameter margindiff = 1.0d-14

  margin for minimal distance (m)
- real(wp), parameter berm_min = 0.0d0

  minimal value gradient berm segment
- real(wp), parameter berm_max = 1.0d0/15

  maximal value gradient berm segment
- real(wp), parameter slope_min = 1.0d0/8

  minimal value gradient slope segment
- real(wp), parameter slope_max = 1.0d0

  maximal value gradient slope segment
- real(wp), parameter margingrad = 0.0025d0

  margin for minimal and maximal gradients
- real(wp), parameter rfactor_min = 0.5d0

  minimal value roughness factor dike segments
- real(wp), parameter rfactor_max = 1.0d0

  maximal value roughness factor dike segments
- real(wp), parameter frunup1_min = 1.24d0

  minimal value model factor 1 for wave run-up
- real(wp), parameter frunup1_max = 2.06d0

  maximal value model factor 1 for wave run-up
- real(wp), parameter frunup2_min = 3.00d0

  minimal value model factor 2 for wave run-up
- real(wp), parameter frunup2_max = 5.00d0

  maximal value model factor 2 for wave run-up

- real(wp), parameter frunup3_min = 1.13d0

  minimal value model factor 3 for wave run-up
- real(wp), parameter frunup3_max = 1.87d0

  maximal value model factor 3 for wave run-up
- real(wp), parameter fb_min = 3.0d0

  minimal value model factor for breaking waves
- real(wp), parameter fb_max = 6.5d0

  maximal value model factor for breaking waves
- real(wp), parameter fn_min = 1.37d0

  minimal value model factor for non-breaking waves
- real(wp), parameter fn_max = 3.83d0

  maximal value model factor for non-breaking waves
- real(wp), parameter fs_min = 0.017d0

  minimal value model factor for shallow waves
- real(wp), parameter fs_max = 0.924d0

  maximal value model factor for shallow waves
- integer, parameter z2_iter_max = 100

  maximal number of iterations for calculation z2
- real(wp), parameter z2_margin = 0.001d0

  margin for convergence criterium calculation z2

4.7.1 Variable Documentation

4.7.1.1 real(wp), parameter typedefinitionsrtoovertopping::berm_max = 1.0d0/15

maximal value gradient berm segment

Definition at line 73 of file typeDefinitionsRTOovertopping.f90.

4.7.1.2 real(wp), parameter typedefinitionsrtoovertopping::berm_min = 0.0d0

minimal value gradient berm segment

Definition at line 72 of file typeDefinitionsRTOovertopping.f90.

4.7.1.3 real(wp), parameter typedefinitionsrtoovertopping::fb_max = 6.5d0

maximal value model factor for breaking waves

Definition at line 86 of file typeDefinitionsRTOovertopping.f90.

4.7.1.4 real(wp), parameter typedefinitionsrtoovertopping::fb_min = 3.0d0

minimal value model factor for breaking waves

Definition at line 85 of file typeDefinitionsRTOovertopping.f90.

4.7.1.5 real(wp), parameter typedefinitionsrtoovertopping::fn_max = 3.83d0

maximal value model factor for non-breaking waves

Definition at line 88 of file typeDefinitionsRTOovertopping.f90.

4.7.1.6 real(wp), parameter typedefinitionsrtoovertopping::fn_min = 1.37d0

minimal value model factor for non-breaking waves

Definition at line 87 of file typeDefinitionsRTOovertopping.f90.

4.7.1.7 real(wp), parameter typedefinitionsrtoovertopping::frunup1_max = 2.06d0

maximal value model factor 1 for wave run-up

Definition at line 80 of file typeDefinitionsRTOovertopping.f90.

4.7.1.8 real(wp), parameter typedefinitionsrtoovertopping::frunup1_min = 1.24d0

minimal value model factor 1 for wave run-up

Definition at line 79 of file typeDefinitionsRTOovertopping.f90.

4.7.1.9 real(wp), parameter typedefinitionsrtoovertopping::frunup2_max = 5.00d0

maximal value model factor 2 for wave run-up

Definition at line 82 of file typeDefinitionsRTOovertopping.f90.

4.7.1.10 real(wp), parameter typedefinitionsrtoovertopping::frunup2_min = 3.00d0

minimal value model factor 2 for wave run-up

Definition at line 81 of file typeDefinitionsRTOovertopping.f90.

4.7.1.11 real(wp), parameter typedefinitionsrtoovertopping::frunup3_max = 1.87d0

maximal value model factor 3 for wave run-up

Definition at line 84 of file typeDefinitionsRTOovertopping.f90.

4.7.1.12 real(wp), parameter typedefinitionsrtoovertopping::frunup3_min = 1.13d0

minimal value model factor 3 for wave run-up

Definition at line 83 of file typeDefinitionsRTOovertopping.f90.

4.7.1.13 real(wp), parameter typedefinitionsrtoovertopping::fs_max = 0.924d0

maximal value model factor for shallow waves

Definition at line 90 of file typeDefinitionsRTOovertopping.f90.

4.7.1.14 real(wp), parameter typedefinitionsrtoovertopping::fs_min = 0.017d0

minimal value model factor for shallow waves

Definition at line 89 of file typeDefinitionsRTOovertopping.f90.

4.7.1.15 real(wp), parameter typedefinitionsrtoovertopping::margindiff = 1.0d-14

margin for minimal distance (m)

Definition at line 71 of file typeDefinitionsRTOovertopping.f90.

4.7.1.16 real(wp), parameter typedefinitionsrtoovertopping::margingrad = 0.0025d0

margin for minimal and maximal gradients

Definition at line 76 of file typeDefinitionsRTOovertopping.f90.

4.7.1.17 real(wp), parameter typedefinitionsrtoovertopping::rfactor_max = 1.0d0

maximal value roughness factor dike segments

Definition at line 78 of file typeDefinitionsRTOovertopping.f90.

4.7.1.18 real(wp), parameter typedefinitionsrtoovertopping::rfactor_min = 0.5d0

minimal value roughness factor dike segments

Definition at line 77 of file typeDefinitionsRTOovertopping.f90.

4.7.1.19 real(wp), parameter typedefinitionsrtoovertopping::slope_max = 1.0d0

maximal value gradient slope segment

Definition at line 75 of file typeDefinitionsRTOovertopping.f90.

4.7.1.20 real(wp), parameter typedefinitionsrtoovertopping::slope_min = 1.0d0/8

minimal value gradient slope segment

Definition at line 74 of file typeDefinitionsRTOovertopping.f90.

4.7.1.21 real(wp), parameter typedefinitionsrtoovertopping::xdiff_min = 2.0d-2

minimal value distance between x-coordinates (m)

Definition at line 70 of file typeDefinitionsRTOovertopping.f90.

4.7.1.22 integer, parameter typedefinitionsrtoovertopping::z2_iter_max = 100

maximal number of iterations for calculation z2

Definition at line 91 of file typeDefinitionsRTOovertopping.f90.

4.7.1.23 real(wp), parameter typedefinitionsrtoovertopping::z2_margin = 0.001d0

margin for convergence criterium calculation z2

Definition at line 92 of file typeDefinitionsRTOovertopping.f90.

4.8 zfunctionswtiovertopping Module Reference

Module for the Limit State Functions (Z-functions) for wave overtopping.

Functions/Subroutines

- subroutine, public calculateqorto (dikeHeight, modelFactorsC, overtopping, load, geometry, succes, errorMessage)

  Subroutine to calculate the overtopping discharge with the RTO-overtopping dll.
- subroutine, public profileinstructure (nrCoordinates, xcoordinates, ycoordinates, dikeHeight, nrCoordsAdjusted, xCoordsAdjusted, zCoordsAdjusted, succes, errorMessage)

  Subroutine to fill the profile in a structure and call the adjustment function of the profile due to a desired dike height.
- subroutine adjustprofile (nrCoordinates, coordinates, dikeHeight, nrCoordsAdjusted, xCoordsAdjusted, zCoordsAdjusted, succes, errorMessage)

  Subroutine adjust the profile due to a desired dike height.
- real(kind=wp) function, public zfunclogratios (qo, qc, mqo, mqc, success, errorMessage)

  Routine to compute the limit state value by using the logs of the overtopping discharges (computed and desired)

4.8.1 Detailed Description

Module for the Limit State Functions (Z-functions) for wave overtopping.

4.8.2 Function/Subroutine Documentation

4.8.2.1 subroutine zfunctionswtiovertopping::adjustprofile (integer, intent(in) *nrCoordinates,* type(tpprofilecoordinate), dimension(nrcoordinates), intent(in) *coordinates,* real(kind=wp), intent(in) *dikeHeight,* integer, intent(out) *nrCoordsAdjusted,* real(kind=wp), dimension(:), pointer *xCoordsAdjusted,* real(kind=wp), dimension(:), pointer *zCoordsAdjusted,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*) [private]

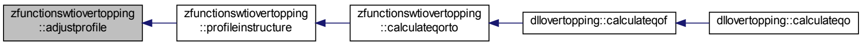
Subroutine adjust the profile due to a desired dike height.

Parameters

in	*nrcoordinates*	number of coordinates of the profile
in	*coordinates*	structure for the profile
in	*dikeheight*	dike height
out	*nrcoordsadjusted*	number of coordinates in the adjusted profile
	xcoordsadjusted	vector with x-coordinates of the adjusted profile
	zcoordsadjusted	vector with y-coordinates of the adjusted profile
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 108 of file zFunctionsWTIOvertopping.f90.

Here is the caller graph for this function:


zfunctionswtiovertopping ::adjustprofile
zfunctionswtiovertopping ::profileinstructure
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.8.2.2 subroutine, public zfunctionswtiovertopping::calculateqorto (real(kind=wp), intent(in) *dikeHeight,* type(overtoppingmodelfactors), intent(in) *modelFactorsC,* type (tpovertopping), intent(out) *overtopping,* type (tpload), intent(in) *load,* type (tpgeometry), intent(in) *geometry,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

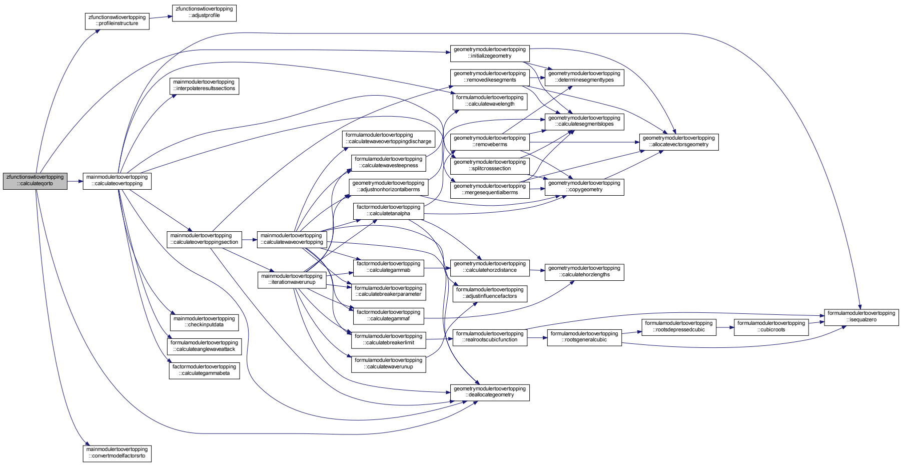
Subroutine to calculate the overtopping discharge with the RTO-overtopping dll.

Parameters

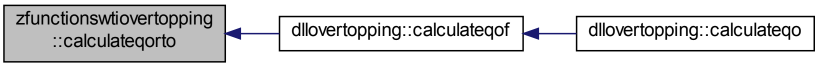
`in`	*dikeheight*	dike height
`in`	*modelfactorsc*	struct with model factors
`out`	*overtopping*	structure with overtopping results
`in`	*geometry*	structure with geometry data
`in`	*load*	structure with load parameters
`out`	*succes*	flag for succes
`out`	*errormessage*	error message

Definition at line 32 of file zFunctionsWTIOvertopping.f90.

Here is the call graph for this function:


zfunctionswtiovertopping ::calculateqorto
zfunctionswtiovertopping ::profileinstructure
zfunctionswtiovertopping ::adjustprofile
mainmodulertoovertopping ::calculateovertopping
mainmodulertoovertopping ::interpolateresultssections
mainmodulertoovertopping ::calculateovertoppingsection
mainmodulertoovertopping ::calculatewaveovertopping
mainmodulertoovertopping ::iterationwaverunup
mainmodulertoovertopping ::checkinputdata
formulamodulertoovertopping ::calculateanglewaveattack
factormodulertoovertopping ::calculategammabeta
mainmodulertoovertopping ::convertmodelfactorsrto
formulamodulertoovertopping ::calculatewaveovertoppingdischarge
formulamodulertoovertopping ::calculatewavesteepness
geometrymodulertoovertopping ::adjustnonhorizontalberms
factormodulertoovertopping ::calculatetanalpha
factormodulertoovertopping ::calculategammab
formulamodulertoovertopping ::calculatebreakerparameter
factormodulertoovertopping ::calculategammaf
formulamodulertoovertopping ::calculatebreakerlimit
formulamodulertoovertopping ::calculatewaverunup
geometrymodulertoovertopping ::initializegeometry
geometrymodulertoovertopping ::removedikesegments
formulamodulertoovertopping ::calculatewavelength
geometrymodulertoovertopping ::removeberms
geometrymodulertoovertopping ::splitcrosssection
geometrymodulertoovertopping ::mergesequentialberms
geometrymodulertoovertopping ::determinesegmenttypes
geometrymodulertoovertopping ::calculatesegmentslopes
geometrymodulertoovertopping ::copygeometry
geometrymodulertoovertopping ::allocatevectorsgeometry
geometrymodulertoovertopping ::calculatehorzdistance
geometrymodulertoovertopping ::calculatehorzlengths
formulamodulertoovertopping ::adjustinfluencefactors
formulamodulertoovertopping ::realrootscubicfunction
formulamodulertoovertopping ::rootsgeneralcubic
formulamodulertoovertopping ::rootsdepressedcubic
formulamodulertoovertopping ::cubicroots
formulamodulertoovertopping ::isequalzero
geometrymodulertoovertopping ::deallocategeometry

Here is the caller graph for this function:


zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.8.2.3 subroutine, public zfunctionswtiovertopping::profileinstructure (integer, intent(in) *nrCoordinates,* real(kind=wp), dimension(nrcoordinates), intent(in) *xcoordinates,* real(kind=wp), dimension(nrcoordinates), intent(in) *ycoordinates,* real(kind=wp), intent(in) *dikeHeight,* integer, intent(out) *nrCoordsAdjusted,* real(kind=wp), dimension(:), pointer *xCoordsAdjusted,* real(kind=wp), dimension(:), pointer *zCoordsAdjusted,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

Subroutine to fill the profile in a structure and call the adjustment function of the profile due to a desired dike height.

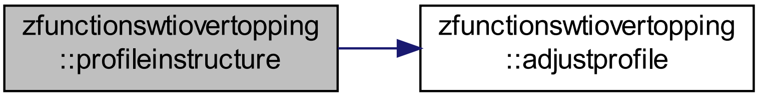
Parameters

`in`	*nrcoordinates*	number of coordinates of the profile
`in`	*xcoordinates*	vector with x-coordinates of the profile
`in`	*ycoordinates*	vector with y-coordinates of the profile

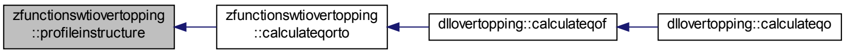
in	*dikeheight*	dike height
out	*nrcoordsadjusted*	number of coordinates in the adjusted profile
	xcoordsadjusted	vector with x-coordinates of the adjusted profile
	zcoordsadjusted	vector with y-coordinates of the adjusted profile
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 83 of file zFunctionsWTIOvertopping.f90.

Here is the call graph for this function:



Here is the caller graph for this function:



4.8.2.4 real (kind=wp) function, public zfunctionswtiovertopping::zfunclogratios (real (kind=wp), intent(in) *qo*, real (kind=wp), intent(in) *qc*, real (kind=wp), intent(in) *mqo*, real (kind=wp), intent(in) *mqc*, logical, intent(out) *success*, character(len=∗), intent(out) *errorMessage*)

Routine to compute the limit state value by using the logs of the overtopping discharges (computed and desired)

Parameters

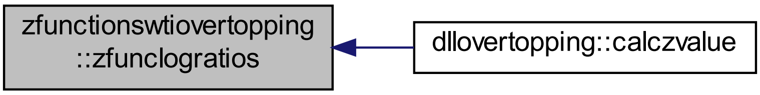
in	*qo*	computed overtopping discharge
in	*qc*	Critical overtopping discharge
in	*mqo*	Model factor computed overtopping discharge
in	*mqc*	Model factor Critical overtopping discharge
out	*success*	Flag for succes
out	*errormessage*	error message, only set if not successful

Returns

Value z-function

Definition at line 193 of file zFunctionsWTIOvertopping.f90.

Here is the caller graph for this function:

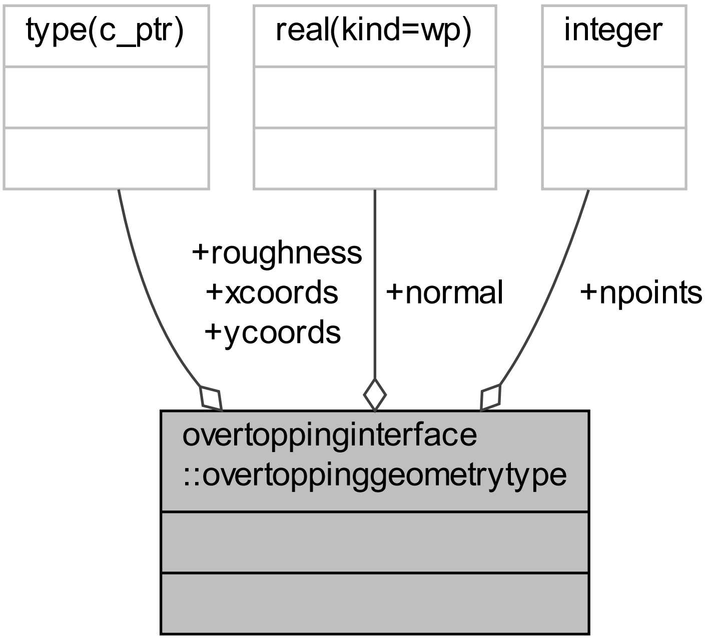


Chapter 5

Data Type Documentation

5.1 overtoppinginterface::overtoppinggeometrytype Type Reference

Collaboration diagram for overtoppinginterface::overtoppinggeometrytype:



Public Attributes

- real(kind=wp) normal
- integer npoints
- type(c_ptr) xcoords
- type(c_ptr) ycoords
- type(c_ptr) roughness

5.1.1 Detailed Description

Definition at line 25 of file overtoppingInterface.f90.

5.1.2 Member Data Documentation

5.1.2.1 real(kind=wp) overtoppinginterface::overtoppinggeometrytype::normal

Definition at line 26 of file overtoppingInterface.f90.

5.1.2.2 integer overtoppinginterface::overtoppinggeometrytype::npoints

Definition at line 27 of file overtoppingInterface.f90.

5.1.2.3 type(c_ptr) overtoppinginterface::overtoppinggeometrytype::roughness

Definition at line 30 of file overtoppingInterface.f90.

5.1.2.4 type(c_ptr) overtoppinginterface::overtoppinggeometrytype::xcoords

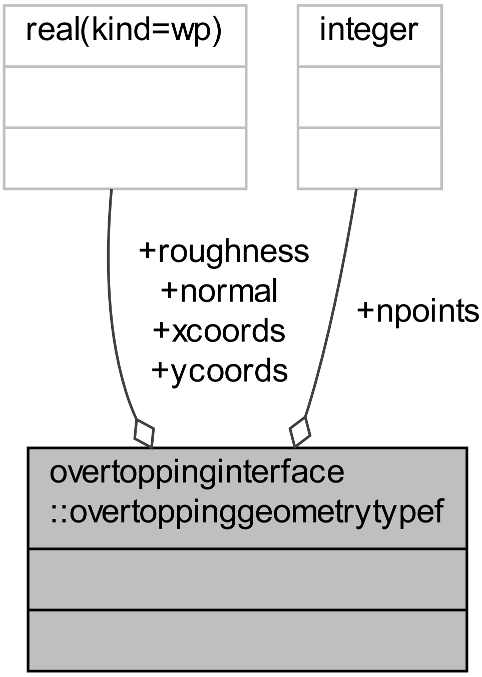
Definition at line 28 of file overtoppingInterface.f90.

5.1.2.5 type(c_ptr) overtoppinginterface::overtoppinggeometrytype::ycoords

Definition at line 29 of file overtoppingInterface.f90.

5.2 overtoppinginterface::overtoppinggeometrytypef Type Reference

Collaboration diagram for overtoppinginterface::overtoppinggeometrytypef:



Public Attributes

- real(kind=wp) normal
- integer npoints
- real(kind=wp), dimension(:), pointer xcoords
- real(kind=wp), dimension(:), pointer ycoords
- real(kind=wp), dimension(:), pointer roughness

5.2.1 Detailed Description

Definition at line 33 of file overtoppingInterface.f90.

5.2.2 Member Data Documentation

5.2.2.1 real(kind=wp) overtoppinginterface::overtoppinggeometrytypef::normal

Definition at line 34 of file overtoppingInterface.f90.

5.2.2.2 integer overtoppinginterface::overtoppinggeometrytypef::npoints

Definition at line 35 of file overtoppingInterface.f90.

5.2.2.3 real(kind=wp), dimension(:), pointer overtoppinginterface::overtoppinggeometrytypef::roughness

Definition at line 38 of file overtoppingInterface.f90.

5.2.2.4 real(kind=wp), dimension(:), pointer overtoppinginterface::overtoppinggeometrytypef::xcoords

Definition at line 36 of file overtoppingInterface.f90.

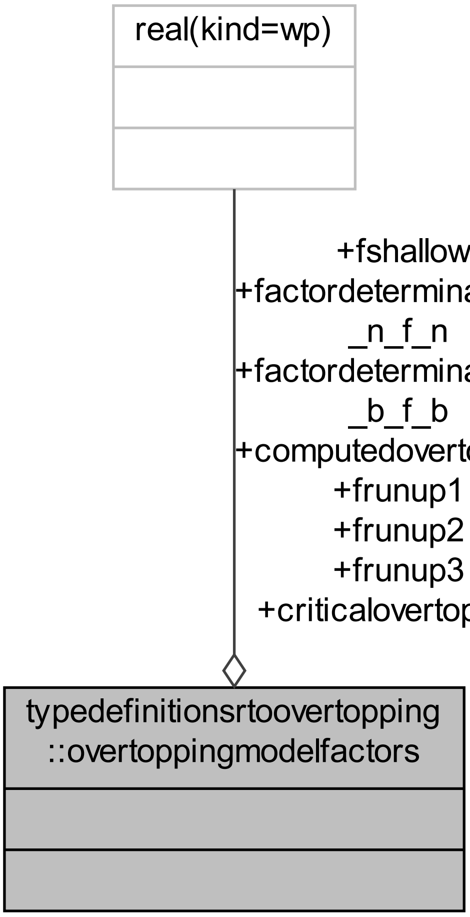
5.2.2.5 real(kind=wp), dimension(:), pointer overtoppinginterface::overtoppinggeometrytypef::ycoords

Definition at line 37 of file overtoppingInterface.f90.

5.3 typedefinitionsrtoovertopping::overtoppingmodelfactors Type Reference

OvertoppingModelFactors: C-structure with model factors.

Collaboration diagram for typedefinitionsrtoovertopping::overtoppingmodelfactors:



Public Attributes

- real(kind=wp) factordeterminationq_n_f_n

  model factor for non-breaking waves
- real(kind=wp) factordeterminationq_b_f_b

  model factor for breaking waves
- real(kind=wp) frunup1

  model factor 1 for wave run-up
- real(kind=wp) frunup2

  model factor 2 for wave run-up
- real(kind=wp) frunup3

  model factor 3 for wave run-up
- real(kind=wp) fshallow

  model factor for shallow waves
- real(kind=wp) computedovertopping

  computed overtopping
- real(kind=wp) criticalovertopping

  critical overtopping

5.3.1 Detailed Description

OvertoppingModelFactors: C-structure with model factors.

Definition at line 50 of file typeDefinitionsRTOovertopping.f90.

5.3.2 Member Data Documentation

5.3.2.1 real(kind=wp) typedefinitionsrtoovertopping::overtoppingmodelfactors::computedovertopping

computed overtopping

Definition at line 57 of file typeDefinitionsRTOovertopping.f90.

5.3.2.2 real(kind=wp) typedefinitionsrtoovertopping::overtoppingmodelfactors::criticalovertopping

critical overtopping

Definition at line 58 of file typeDefinitionsRTOovertopping.f90.

5.3.2.3 real(kind=wp) typedefinitionsrtoovertopping::overtoppingmodelfactors::factordeterminationq_b_f_b

model factor for breaking waves

Definition at line 52 of file typeDefinitionsRTOovertopping.f90.

5.3.2.4 real(kind=wp) typedefinitionsrtoovertopping::overtoppingmodelfactors::factordeterminationq_n_f_n

model factor for non-breaking waves

Definition at line 51 of file typeDefinitionsRTOovertopping.f90.

5.3.2.5 real(kind=wp) typedefinitionsrtoovertopping::overtoppingmodelfactors::frunup1

model factor 1 for wave run-up

Definition at line 53 of file typeDefinitionsRTOovertopping.f90.

5.3.2.6 real(kind=wp) typedefinitionsrtoovertopping::overtoppingmodelfactors::frunup2

model factor 2 for wave run-up

Definition at line 54 of file typeDefinitionsRTOovertopping.f90.

5.3.2.7 real(kind=wp) typedefinitionsrtoovertopping::overtoppingmodelfactors::frunup3

model factor 3 for wave run-up

Definition at line 55 of file typeDefinitionsRTOovertopping.f90.

5.3.2.8 real(kind=wp) typedefinitionsrtoovertopping::overtoppingmodelfactors::fshallow

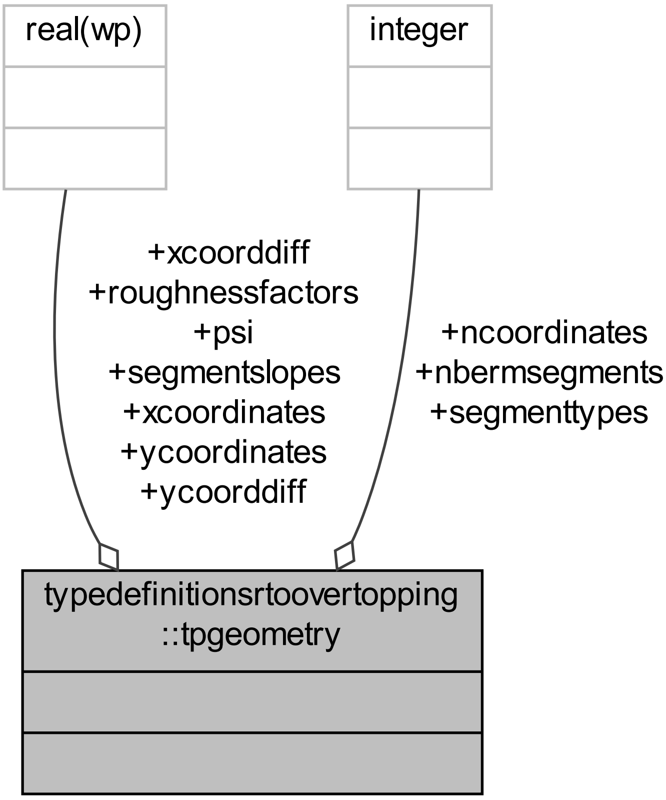
model factor for shallow waves

Definition at line 56 of file typeDefinitionsRTOovertopping.f90.

5.4 typedefinitionsrtoovertopping::tpgeometry Type Reference

tpGeometry: structure with geometry data

Collaboration diagram for typedefinitionsrtoovertopping::tpgeometry:



Public Attributes

- real(wp) psi

  dike normal (degree)
- integer ncoordinates

  number of coordinates cross section
- real(wp), dimension(:), pointer xcoordinates

  vector with x-coordinates cross section (m)
- real(wp), dimension(:), pointer ycoordinates

  vector with y-coordinates cross section (m+NAP)
- real(wp), dimension(:), pointer roughnessfactors

  vector with roughness factors cross section
- real(wp), dimension(:), pointer xcoorddiff

  vector with differences in x-coordinates (m)
- real(wp), dimension(:), pointer ycoorddiff

  vector with differences in y-coordinates (m)
- real(wp), dimension(:), pointer segmentslopes

  vector with slopes dike segments
- integer, dimension(:), pointer segmenttypes

vector with segment types (1=slope,2=berm,3=other)

- integer nbermsegments

  number of berm segments

5.4.1 Detailed Description

tpGeometry: structure with geometry data

Definition at line 18 of file typeDefinitionsRTOovertopping.f90.

5.4.2 Member Data Documentation

5.4.2.1 integer typedefinitionsrtoovertopping::tpgeometry::nbermsegments

number of berm segments

Definition at line 28 of file typeDefinitionsRTOovertopping.f90.

5.4.2.2 integer typedefinitionsrtoovertopping::tpgeometry::ncoordinates

number of coordinates cross section

Definition at line 20 of file typeDefinitionsRTOovertopping.f90.

5.4.2.3 real(wp) typedefinitionsrtoovertopping::tpgeometry::psi

dike normal (degree)

Definition at line 19 of file typeDefinitionsRTOovertopping.f90.

5.4.2.4 real(wp), dimension(:), pointer typedefinitionsrtoovertopping::tpgeometry::roughnessfactors

vector with roughness factors cross section

Definition at line 23 of file typeDefinitionsRTOovertopping.f90.

5.4.2.5 real(wp), dimension(:), pointer typedefinitionsrtoovertopping::tpgeometry::segmentslopes

vector with slopes dike segments

Definition at line 26 of file typeDefinitionsRTOovertopping.f90.

5.4.2.6 integer, dimension(:), pointer typedefinitionsrtoovertopping::tpgeometry::segmenttypes

vector with segment types (1=slope,2=berm,3=other)

Definition at line 27 of file typeDefinitionsRTOovertopping.f90.

5.4.2.7 real(wp), dimension(:), pointer typedefinitionsrtoovertopping::tpgeometry::xcoorddiff

vector with differences in x-coordinates (m)

Definition at line 24 of file typeDefinitionsRTOovertopping.f90.

5.4.2.8 real(wp), dimension(:), pointer typedefinitionsrtoovertopping::tpgeometry::xcoordinates

vector with x-coordinates cross section (m)

Definition at line 21 of file typeDefinitionsRTOovertopping.f90.

5.4.2.9 real(wp), dimension(:), pointer typedefinitionsrtoovertopping::tpgeometry::ycoorddiff

vector with differences in y-coordinates (m)

Definition at line 25 of file typeDefinitionsRTOovertopping.f90.

5.4.2.10 real(wp), dimension(:), pointer typedefinitionsrtoovertopping::tpgeometry::ycoordinates

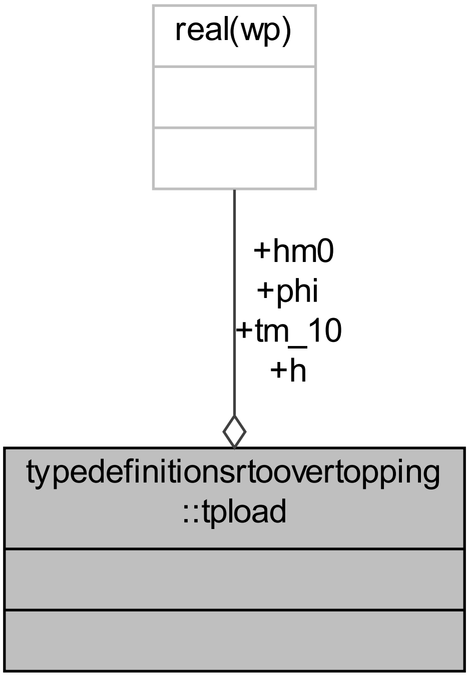
vector with y-coordinates cross section (m+NAP)

Definition at line 22 of file typeDefinitionsRTOovertopping.f90.

5.5 typedefinitionsrtoovertopping::tpload Type Reference

tpLoad: structure with load parameters

Collaboration diagram for typedefinitionsrtoovertopping::tpload:



Public Attributes

- real(wp) h

local water level (m+NAP)

- real(wp) hm0

  significant wave height (m)

- real(wp) tm_10

  spectral wave period (s)

- real(wp) phi

  wave direction (degree)

5.5.1 Detailed Description

tpLoad: structure with load parameters

Definition at line 32 of file typeDefinitionsRTOovertopping.f90.

5.5.2 Member Data Documentation

5.5.2.1 real(wp) typedefinitionsrtoovertopping::tpload::h

local water level (m+NAP)

Definition at line 33 of file typeDefinitionsRTOovertopping.f90.

5.5.2.2 real(wp) typedefinitionsrtoovertopping::tpload::hm0

significant wave height (m)

Definition at line 34 of file typeDefinitionsRTOovertopping.f90.

5.5.2.3 real(wp) typedefinitionsrtoovertopping::tpload::phi

wave direction (degree)

Definition at line 36 of file typeDefinitionsRTOovertopping.f90.

5.5.2.4 real(wp) typedefinitionsrtoovertopping::tpload::tm_10

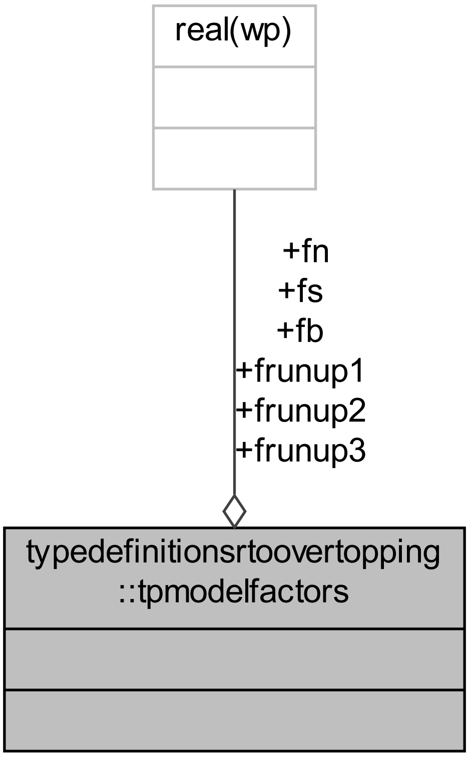
spectral wave period (s)

Definition at line 35 of file typeDefinitionsRTOovertopping.f90.

5.6 typedefinitionsrtoovertopping::tpmodelfactors Type Reference

tpModelFactors: structure with model factors

Collaboration diagram for typedefinitionsrtoovertopping::tpmodelfactors:



Public Attributes

- real(wp) frunup1

  model factor 1 for wave run-up
- real(wp) frunup2

  model factor 2 for wave run-up
- real(wp) frunup3

  model factor 3 for wave run-up
- real(wp) fb

  model factor for breaking waves
- real(wp) fn

  model factor for non-breaking waves
- real(wp) fs

  model factor for shallow waves

5.6.1 Detailed Description

tpModelFactors: structure with model factors

Definition at line 40 of file typeDefinitionsRTOovertopping.f90.

5.6.2 Member Data Documentation

5.6.2.1 real(wp) typedefinitionsrtoovertopping::tpmodelfactors::fb

model factor for breaking waves

Definition at line 44 of file typeDefinitionsRTOovertopping.f90.

5.6.2.2 real(wp) typedefinitionsrtoovertopping::tpmodelfactors::fn

model factor for non-breaking waves

Definition at line 45 of file typeDefinitionsRTOovertopping.f90.

5.6.2.3 real(wp) typedefinitionsrtoovertopping::tpmodelfactors::frunup1

model factor 1 for wave run-up

Definition at line 41 of file typeDefinitionsRTOovertopping.f90.

5.6.2.4 real(wp) typedefinitionsrtoovertopping::tpmodelfactors::frunup2

model factor 2 for wave run-up

Definition at line 42 of file typeDefinitionsRTOovertopping.f90.

5.6.2.5 real(wp) typedefinitionsrtoovertopping::tpmodelfactors::frunup3

model factor 3 for wave run-up

Definition at line 43 of file typeDefinitionsRTOovertopping.f90.

5.6.2.6 real(wp) typedefinitionsrtoovertopping::tpmodelfactors::fs

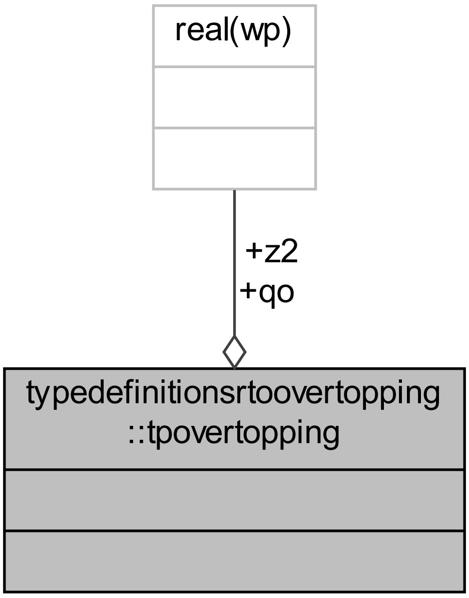
model factor for shallow waves

Definition at line 46 of file typeDefinitionsRTOovertopping.f90.

5.7 typedefinitionsrtoovertopping::tpovertopping Type Reference

tpOvertopping: structure with overtopping results

Collaboration diagram for typedefinitionsrtoovertopping::tpovertopping:



Public Attributes

- real(wp) z2

  2% wave run-up (m)
- real(wp) qo

  wave overtopping discharge (m3/m per s)

5.7.1 Detailed Description

tpOvertopping: structure with overtopping results

Definition at line 62 of file typeDefinitionsRTOovertopping.f90.

5.7.2 Member Data Documentation

5.7.2.1 real(wp) typedefinitionsrtoovertopping::tpovertopping::qo

wave overtopping discharge (m3/m per s)

Definition at line 64 of file typeDefinitionsRTOovertopping.f90.

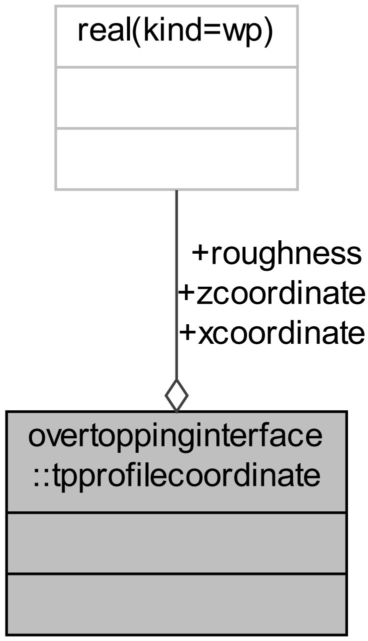
5.7.2.2 real(wp) typedefinitionsrtoovertopping::tpovertopping::z2

2% wave run-up (m)

Definition at line 63 of file typeDefinitionsRTOovertopping.f90.

5.8 overtoppinginterface::tpprofilecoordinate Type Reference

Collaboration diagram for overtoppinginterface::tpprofilecoordinate:



Public Attributes

- real(kind=wp) xcoordinate

  X-coordinate foreland profile.
- real(kind=wp) zcoordinate

  Z-coordinate foreland profile.
- real(kind=wp) roughness

  Roughness of the area between two points.

5.8.1 Detailed Description

Definition at line 19 of file overtoppingInterface.f90.

5.8.2 Member Data Documentation

5.8.2.1 real(kind=wp) overtoppinginterface::tpprofilecoordinate::roughness

Roughness of the area between two points.

Definition at line 22 of file overtoppingInterface.f90.

5.8.2.2 real(kind=wp) overtoppinginterface::tpprofilecoordinate::xcoordinate

X-coordinate foreland profile.

Definition at line 20 of file overtoppingInterface.f90.

5.8.2.3 real(kind=wp) overtoppinginterface::tpprofilecoordinate::zcoordinate

Z-coordinate foreland profile.

Definition at line 21 of file overtoppingInterface.f90.

Chapter 6

File Documentation

6.1 dllOvertopping.f90 File Reference

Main entry for the dll DikesOvertopping FUNCTIONS/SUBROUTINES exported from dllOvertopping.dll:

Modules

- module dllovertopping

  Calculate one type of overtopping.

Functions/Subroutines

- subroutine, public dllovertopping::calculateqo (load, geometryInput, dikeHeight, modelFactors, overtopping, success, errorText)

  Subroutine that calculates the discharge needed for the Z-function DikesOvertopping Wrapper for calculateQoF↩ : covert C-like input structures to Fortran input structures.
- subroutine, public dllovertopping::calculateqof (load, geometryF, dikeHeight, modelFactors, overtopping, success, errorText)

  Subroutine that calculates the discharge needed for the Z-function DikesOvertopping.
- subroutine, public dllovertopping::calczvalue (criticalOvertoppingRate, modelFactors, Qo, z, success, error↩ Message)

  Subroutine that calculates the Z-function DikesOvertopping based on the discharge calculated with calculateQoF.
- subroutine, public dllovertopping::versionnumber (version)

6.1.1 Detailed Description

Main entry for the dll DikesOvertopping FUNCTIONS/SUBROUTINES exported from dllOvertopping.dll:

- zFuncOvertopping
- calculateQo
- calculateQoF
- versionNumber

6.2 factorModuleRTOovertopping.f90 File Reference

This file contains a module with functions for the slope angle and influence factors.

Modules

- module factormodulertoovertopping

Functions/Subroutines

- subroutine, public factormodulertoovertopping::calculatetanalpha (h, Hm0, z2, geometry, tanAlpha, succes, errorMessage)

  calculateTanAlpha representative slope angle
- subroutine, public factormodulertoovertopping::calculategammabeta (Hm0, Tm_10, beta, gammaBeta_↩ z, gammaBeta_o)

  calculateGammaBeta influence factor angle of wave attack
- subroutine, public factormodulertoovertopping::calculategammaf (h, ksi0, ksi0Limit, gammaB, z2, geometry, gammaF, succes, errorMessage)

  calculateGammaF influence factor roughness
- subroutine, public factormodulertoovertopping::calculategammab (h, Hm0, z2, geometry, NbermSegments, gammaB, succes, errorMessage)

  calculateGammaB influence factor berms

6.2.1 Detailed Description

This file contains a module with functions for the slope angle and influence factors.

6.3 formulaModuleRTOovertopping.f90 File Reference

This file contains a module with the core computations for Dikes Overtopping.

Modules

- module formulamodulertoovertopping

Functions/Subroutines

- subroutine, public formulamodulertoovertopping::calculatewaverunup (Hm0, s0, ksi0, ksi0Limit, gammaB, gammaF, gammaBeta, modelFactors, z2, succes, errorMessage)

  calculateWaveRunup: calculate wave runup
- subroutine, public formulamodulertoovertopping::calculatewaveovertoppingdischarge (h, Hm0, tanAlpha, gammaB, gammaF, gammaBeta, ksi0, hCrest, modelFactors, Qo, succes, errorMessage)

  calculateWaveOvertoppingDischarge: calculate the wave overtopping discharge
- subroutine, public formulamodulertoovertopping::calculatewavelength (Tm_10, L0)

  calculateWaveLength: calculate the wave length
- subroutine, public formulamodulertoovertopping::calculatewavesteepness (Hm0, Tm_10, s0, succes, error↩ Message)

  calculateWaveSteepness: calculate the wave steepness
- subroutine, public formulamodulertoovertopping::calculatebreakerparameter (tanAlpha, s0, ksi0, succes, errorMessage)

  calculateBreakerParameter: calculate the breaker parameter
- subroutine, public formulamodulertoovertopping::calculateanglewaveattack (phi, psi, beta)

  calculateAngleWaveAttack: calculate the angle of wave attack

- subroutine, public formulamodulertoovertopping::calculatebreakerlimit (modelFactors, gammaB, ksi0Limit, succes, errorMessage)

  calculateBreakerLimit: calculate the breaker limit
- subroutine, public formulamodulertoovertopping::adjustinfluencefactors (gammaB, gammaF, gammaBeta, gammaBetaType, ksi0, ksi0Limit, succes, errorMessage)

  adjustInfluenceFactors: adjust the influence factors
- subroutine, public formulamodulertoovertopping::realrootscubicfunction (a, b, c, d, N, x, succes, error←Message)

  realRootsCubicFunction: calculate the roots of a cubic function
- subroutine, public formulamodulertoovertopping::rootsgeneralcubic (a, b, c, d, z, succes, errorMessage)

  rootsGeneralCubic: calculate the roots of a generic cubic function
- subroutine, public formulamodulertoovertopping::rootsdepressedcubic (p, q, z)

  rootsDepressedCubic: calculate the roots of a depressed cubic function
- subroutine, public formulamodulertoovertopping::cubicroots (z, roots)

  cubicRoots: calculate the roots of a cubic function
- logical function, public formulamodulertoovertopping::isequalreal (x1, x2)

  isEqualReal: are two reals (almost) equal
- logical function, public formulamodulertoovertopping::isequalzero (x)

  isEqualZero: is a real (almost) zero

6.3.1 Detailed Description

This file contains a module with the core computations for Dikes Overtopping.

6.4 geometryModuleRTOovertopping.f90 File Reference

This file contains a module with the core computations for Dikes Overtopping related to the geometry.

Modules

- module geometrymodulertoovertopping

Functions/Subroutines

- subroutine, public geometrymodulertoovertopping::checkcrosssection (psi, nCoordinates, xCoordinates, y←Coordinates, roughnessFactors, succes, errorMessage)

  checkCrossSection: check cross section
- subroutine, public geometrymodulertoovertopping::initializegeometry (psi, nCoordinates, xCoordinates, y←Coordinates, roughnessFactors, geometry, succes, errorMessage)

  initializeGeometry: initialize the geometry
- subroutine, public geometrymodulertoovertopping::allocatevectorsgeometry (nCoordinates, geometry)

  allocateVectorsGeometry: allocate the geometry vectors
- subroutine, public geometrymodulertoovertopping::deallocategeometry (geometry)

  deallocateGeometry: deallocate the geometry vectors
- subroutine, public geometrymodulertoovertopping::calculatesegmentslopes (geometry, succes, error←Message)

  calculateSegmentSlopes: calculate the segment slopes
- subroutine, public geometrymodulertoovertopping::determinesegmenttypes (geometry)

  determineSegmentTypes: determine the segment types
- subroutine, public geometrymodulertoovertopping::copygeometry (geometry, geometryCopy)

copyGeometry: copy a geometry structure

- subroutine, public geometrymodulertoovertopping::isequalgeometry (geometry1, geometry2, succes, error↩Message)

  isEqualGeometry: are two geometries equal
- subroutine, public geometrymodulertoovertopping::mergesequentialberms (geometry, geometryMerged↩Berms, succes, errorMessage)

  mergeSequentialBerms: merge sequential berms
- subroutine, public geometrymodulertoovertopping::adjustnonhorizontalberms (geometry, geometryFlat↩Berms, succes, errorMessage)

  adjustNonHorizontalBerms: adjust non-horizontal berms
- subroutine, public geometrymodulertoovertopping::removeberms (geometry, geometryNoBerms, succes, errorMessage)

  removeBerms: remove berms
- subroutine, public geometrymodulertoovertopping::removedikesegments (geometry, index, geometry↩Adjusted, succes, errorMessage)

  removeDikeSegments: remove dike segments
- subroutine, public geometrymodulertoovertopping::splitcrosssection (geometry, L0, NwideBerms, geometrysectionB, geometrysectionF, succes, errorMessage)

  splitCrossSection: split a cross section
- subroutine, public geometrymodulertoovertopping::calculatehorzlengths (geometry, yLower, yUpper, horz↩Lengths, succes, errorMessage)

  calculateHorzLengths: calculate horizontal lengths
- subroutine, public geometrymodulertoovertopping::calculatehorzdistance (geometry, yLower, yUpper, dx, succes, errorMessage)

  calculateHorzDistance: calculate horizontal distance
- subroutine, public geometrymodulertoovertopping::writecrosssection (geometry, geometryName)

  writeCrossSection: write a cross section

6.4.1 Detailed Description

This file contains a module with the core computations for Dikes Overtopping related to the geometry.

6.5 mainModuleRTOovertopping.f90 File Reference

This file contains a module with the core computations for Dikes Overtopping.

Modules

- module mainmodulertoovertopping

Functions/Subroutines

- subroutine, public mainmodulertoovertopping::calculateovertopping (geometry, load, modelFactors, prob↩Context, overtopping, succes, errorMessage)

  calculateOvertopping: calculate the overtopping
- subroutine, public mainmodulertoovertopping::calculateovertoppingsection (geometry, h, Hm0, Tm_10, L0, gammaBeta_z, gammaBeta_o, modelFactors, overtopping, succes, errorMessage)

  calculateOvertoppingSection: calculate the overtopping for a section
- subroutine, public mainmodulertoovertopping::iterationwaverunup (geometry, h, Hm0, Tm_10, L0, gamma↩Beta_z, modelFactors, z2, succes, errorMessage)

iterationWaveRunup: iteration for the wave runup

- subroutine, public mainmodulertoovertopping::calculatewaveovertopping (geometry, h, Hm0, Tm_10, z2, gammaBeta_o, modelFactors, Qo, succes, errorMessage)

  calculateWaveOvertopping: calculate wave overtopping
- subroutine mainmodulertoovertopping::calculateovertoppingnegativefreeboard (load, geometry, overtopping, succes, errorMessage)

  calculateOvertoppingNegativeFreeboard: calculate overtopping in case of negative freeboard
- subroutine, public mainmodulertoovertopping::interpolateresultssections (geometry, L0, NwideBerms, overtoppingB, overtoppingF, overtopping, succes, errorMessage)

  interpolateResultsSections: interpolate results for split cross sections
- subroutine, public mainmodulertoovertopping::checkinputdata (geometry, load, modelFactors, probContext, succes, errorMessage)

  checkInputdata: check the input data
- subroutine, public mainmodulertoovertopping::convertmodelfactorsrto (modelFactorsF, modelFactorsC)

  convertModelfactorsRTO: convert the model factors
- subroutine, public mainmodulertoovertopping::fillmodelfactorsrto (modelFactors, fn, fb, frunup1, frunup2, frunup3, fshallow)

  fillModelfactorsRTO: fill the model factors

6.5.1 Detailed Description

This file contains a module with the core computations for Dikes Overtopping.

6.6 overtoppingInterface.f90 File Reference

This file contains the parameters and types (structs) as part of the interface to and from dllOvertopping.

Data Types

- type overtoppinginterface::tpprofilecoordinate
- type overtoppinginterface::overtoppinggeometrytype
- type overtoppinginterface::overtoppinggeometrytypef

Modules

- module overtoppinginterface

Variables

- integer, parameter, public overtoppinginterface::varmodelfactorcriticalovertopping = 8

  Model factor critical overtopping.

6.6.1 Detailed Description

This file contains the parameters and types (structs) as part of the interface to and from dllOvertopping.

6.7 typeDefinitionsRTOovertopping.f90 File Reference

This file contains a module with the type definitions for Dikes Overtopping.

Data Types

- type typedefinitionsrtoovertopping::tpgeometry

  tpGeometry: structure with geometry data
- type typedefinitionsrtoovertopping::tpload

  tpLoad: structure with load parameters
- type typedefinitionsrtoovertopping::tpmodelfactors

  tpModelFactors: structure with model factors
- type typedefinitionsrtoovertopping::overtoppingmodelfactors

  OvertoppingModelFactors: C-structure with model factors.
- type typedefinitionsrtoovertopping::tpovertopping

  tpOvertopping: structure with overtopping results

Modules

- module typedefinitionsrtoovertopping

Variables

- real(wp), parameter typedefinitionsrtoovertopping::xdiff_min = 2.0d-2

  minimal value distance between x-coordinates (m)
- real(wp), parameter typedefinitionsrtoovertopping::margindiff = 1.0d-14

  margin for minimal distance (m)
- real(wp), parameter typedefinitionsrtoovertopping::berm_min = 0.0d0

  minimal value gradient berm segment
- real(wp), parameter typedefinitionsrtoovertopping::berm_max = 1.0d0/15

  maximal value gradient berm segment
- real(wp), parameter typedefinitionsrtoovertopping::slope_min = 1.0d0/8

  minimal value gradient slope segment
- real(wp), parameter typedefinitionsrtoovertopping::slope_max = 1.0d0

  maximal value gradient slope segment
- real(wp), parameter typedefinitionsrtoovertopping::margingrad = 0.0025d0

  margin for minimal and maximal gradients
- real(wp), parameter typedefinitionsrtoovertopping::rfactor_min = 0.5d0

  minimal value roughness factor dike segments
- real(wp), parameter typedefinitionsrtoovertopping::rfactor_max = 1.0d0

  maximal value roughness factor dike segments
- real(wp), parameter typedefinitionsrtoovertopping::frunup1_min = 1.24d0

  minimal value model factor 1 for wave run-up
- real(wp), parameter typedefinitionsrtoovertopping::frunup1_max = 2.06d0

  maximal value model factor 1 for wave run-up
- real(wp), parameter typedefinitionsrtoovertopping::frunup2_min = 3.00d0

  minimal value model factor 2 for wave run-up
- real(wp), parameter typedefinitionsrtoovertopping::frunup2_max = 5.00d0

  maximal value model factor 2 for wave run-up
- real(wp), parameter typedefinitionsrtoovertopping::frunup3_min = 1.13d0

  minimal value model factor 3 for wave run-up
- real(wp), parameter typedefinitionsrtoovertopping::frunup3_max = 1.87d0

  maximal value model factor 3 for wave run-up
- real(wp), parameter typedefinitionsrtoovertopping::fb_min = 3.0d0

  minimal value model factor for breaking waves
- real(wp), parameter typedefinitionsrtoovertopping::fb_max = 6.5d0

  maximal value model factor for breaking waves
- real(wp), parameter typedefinitionsrtoovertopping::fn_min = 1.37d0

  minimal value model factor for non-breaking waves
- real(wp), parameter typedefinitionsrtoovertopping::fn_max = 3.83d0

  maximal value model factor for non-breaking waves
- real(wp), parameter typedefinitionsrtoovertopping::fs_min = 0.017d0

  minimal value model factor for shallow waves
- real(wp), parameter typedefinitionsrtoovertopping::fs_max = 0.924d0

  maximal value model factor for shallow waves
- integer, parameter typedefinitionsrtoovertopping::z2_iter_max = 100

  maximal number of iterations for calculation z2
- real(wp), parameter typedefinitionsrtoovertopping::z2_margin = 0.001d0

  margin for convergence criterium calculation z2

6.7.1 Detailed Description

This file contains a module with the type definitions for Dikes Overtopping.

6.8 zFunctionsWTIOvertopping.f90 File Reference

This file contains the limit state functions for wave overtopping within WTI.

Modules

- module zfunctionswtiovertopping

  Module for the Limit State Functions (Z-functions) for wave overtopping.

Functions/Subroutines

- subroutine, public zfunctionswtiovertopping::calculateqorto (dikeHeight, modelFactorsC, overtopping, load, geometry, succes, errorMessage)

  Subroutine to calculate the overtopping discharge with the RTO-overtopping dll.
- subroutine, public zfunctionswtiovertopping::profileinstructure (nrCoordinates, xcoordinates, ycoordinates, dikeHeight, nrCoordsAdjusted, xCoordsAdjusted, zCoordsAdjusted, succes, errorMessage)

  Subroutine to fill the profile in a structure and call the adjustment function of the profile due to a desired dike height.
- subroutine zfunctionswtiovertopping::adjustprofile (nrCoordinates, coordinates, dikeHeight, nrCoords↩ Adjusted, xCoordsAdjusted, zCoordsAdjusted, succes, errorMessage)

  Subroutine adjust the profile due to a desired dike height.
- real(kind=wp) function, public zfunctionswtiovertopping::zfunclogratios (qo, qc, mqo, mqc, success, error↩ Message)

  Routine to compute the limit state value by using the logs of the overtopping discharges (computed and desired)

6.8.1 Detailed Description

This file contains the limit state functions for wave overtopping within WTI.

Index

adjustinfluencefactors
    formulamodulertoovertopping, 14
adjustnonhorizontalberms
    geometrymodulertoovertopping, 21
adjustprofile
    zfunctionswtiovertopping, 43
allocatevectorsgeometry
    geometrymodulertoovertopping, 22

berm_max
    typedefinitionsrtoovertopping, 40
berm_min
    typedefinitionsrtoovertopping, 40

calculateanglewaveattack
    formulamodulertoovertopping, 14
calculatebreakerlimit
    formulamodulertoovertopping, 14
calculatebreakerparameter
    formulamodulertoovertopping, 15
calculategammab
    factormodulertoovertopping, 10
calculategammabeta
    factormodulertoovertopping, 11
calculategammaf
    factormodulertoovertopping, 11
calculatehorzdistance
    geometrymodulertoovertopping, 22
calculatehorzlengths
    geometrymodulertoovertopping, 23
calculateovertopping
    mainmodulertoovertopping, 32
calculateovertoppingnegativefreeboard
    mainmodulertoovertopping, 33
calculateovertoppingsection
    mainmodulertoovertopping, 33
calculateqo
    dllovertopping, 7
calculateqof
    dllovertopping, 8
calculateqorto
    zfunctionswtiovertopping, 43
calculatesegmentslopes
    geometrymodulertoovertopping, 23
calculatetanalpha
    factormodulertoovertopping, 12
calculatewavelength
    formulamodulertoovertopping, 15
calculatewaveovertopping
    mainmodulertoovertopping, 34
calculatewaveovertoppingdischarge
    formulamodulertoovertopping, 16
calculatewaverunup
    formulamodulertoovertopping, 16
calculatewavesteepness
    formulamodulertoovertopping, 17
calczvalue
    dllovertopping, 9
checkcrosssection
    geometrymodulertoovertopping, 24
checkinputdata
    mainmodulertoovertopping, 35
computedovertopping
    typedefinitionsrtoovertopping::overtoppingmodelfactors, 53
convertmodelfactorsrto
    mainmodulertoovertopping, 35
copygeometry
    geometrymodulertoovertopping, 25
criticalovertopping
    typedefinitionsrtoovertopping::overtoppingmodelfactors, 53
cubicroots
    formulamodulertoovertopping, 18

deallocategeometry
    geometrymodulertoovertopping, 25
determinesegmenttypes
    geometrymodulertoovertopping, 26
dllOvertopping.f90, 65
dllovertopping, 7
    calculateqo, 7
    calculateqof, 8
    calczvalue, 9
    versionnumber, 10

factorModuleRTOovertopping.f90, 65
factordeterminationq_b_f_b
    typedefinitionsrtoovertopping::overtoppingmodelfactors, 53
factordeterminationq_n_f_n
    typedefinitionsrtoovertopping::overtoppingmodelfactors, 53
factormodulertoovertopping, 10
    calculategammab, 10
    calculategammabeta, 11
    calculategammaf, 11
    calculatetanalpha, 12
fb
    typedefinitionsrtoovertopping::tpmodelfactors, 60

fb_max
typedefinitionsrtoovertopping, 40
fb_min
typedefinitionsrtoovertopping, 40
fillmodelfactorsrto
mainmodulertoovertopping, 37
fn
typedefinitionsrtoovertopping::tpmodelfactors, 60
fn_max
typedefinitionsrtoovertopping, 40
fn_min
typedefinitionsrtoovertopping, 40
formulaModuleRTOovertopping.f90, 66
formulamodulertoovertopping, 13
adjustinfluencefactors, 14
calculateanglewaveattack, 14
calculatebreakerlimit, 14
calculatebreakerparameter, 15
calculatewavelength, 15
calculatewaveovertoppingdischarge, 16
calculatewaverunup, 16
calculatewavesteepness, 17
cubicroots, 18
isequalreal, 18
isequalzero, 19
realrootscubicfunction, 19
rootsdepressedcubic, 19
rootsgeneralcubic, 20
frunup1
typedefinitionsrtoovertopping::overtoppingmodelfactors, 53
typedefinitionsrtoovertopping::tpmodelfactors, 60
frunup1_max
typedefinitionsrtoovertopping, 41
frunup1_min
typedefinitionsrtoovertopping, 41
frunup2
typedefinitionsrtoovertopping::overtoppingmodelfactors, 54
typedefinitionsrtoovertopping::tpmodelfactors, 60
frunup2_max
typedefinitionsrtoovertopping, 41
frunup2_min
typedefinitionsrtoovertopping, 41
frunup3
typedefinitionsrtoovertopping::overtoppingmodelfactors, 54
typedefinitionsrtoovertopping::tpmodelfactors, 60
frunup3_max
typedefinitionsrtoovertopping, 41
frunup3_min
typedefinitionsrtoovertopping, 41
fs
typedefinitionsrtoovertopping::tpmodelfactors, 60
fs_max
typedefinitionsrtoovertopping, 41
fs_min
typedefinitionsrtoovertopping, 41
fshallow
typedefinitionsrtoovertopping::overtoppingmodelfactors, 54

geometryModuleRTOovertopping.f90, 67
geometrymodulertoovertopping, 20
adjustnonhorizontalberms, 21
allocatevectorsgeometry, 22
calculatehorzdistance, 22
calculatehorzlengths, 23
calculatesegmentslopes, 23
checkcrosssection, 24
copygeometry, 25
deallocategeometry, 25
determinesegmenttypes, 26
initializegeometry, 26
isequalgeometry, 27
mergesequentialberms, 27
removeberms, 29
removedikesegments, 30
splitcrosssection, 30
writecrosssection, 31

h
typedefinitionsrtoovertopping::tpload, 58
hm0
typedefinitionsrtoovertopping::tpload, 58

initializegeometry
geometrymodulertoovertopping, 26
interpolateresultssections
mainmodulertoovertopping, 37
isequalgeometry
geometrymodulertoovertopping, 27
isequalreal
formulamodulertoovertopping, 18
isequalzero
formulamodulertoovertopping, 19
iterationwaverunup
mainmodulertoovertopping, 37

mainModuleRTOovertopping.f90, 68
mainmodulertoovertopping, 31
calculateovertopping, 32
calculateovertoppingnegativefreeboard, 33
calculateovertoppingsection, 33
calculatewaveovertopping, 34
checkinputdata, 35
convertmodelfactorsrto, 35
fillmodelfactorsrto, 37
interpolateresultssections, 37
iterationwaverunup, 37
margindiff
typedefinitionsrtoovertopping, 41
margingrad
typedefinitionsrtoovertopping, 42
mergesequentialberms
geometrymodulertoovertopping, 27

nbermsegments

typedefinitionsrtoovertopping::tpgeometry, 56
ncoordinates
typedefinitionsrtoovertopping::tpgeometry, 56
normal
overtoppinginterface::overtoppinggeometrytype, 50
overtoppinginterface::overtoppinggeometrytypef, 51
npoints
overtoppinginterface::overtoppinggeometrytype, 50
overtoppinginterface::overtoppinggeometrytypef, 51

overtoppingInterface.f90, 69
overtoppinginterface, 38
varmodelfactorcriticalovertopping, 39
overtoppinginterface::overtoppinggeometrytype, 49
normal, 50
npoints, 50
roughness, 50
xcoords, 50
ycoords, 50
overtoppinginterface::overtoppinggeometrytypef, 51
normal, 51
npoints, 51
roughness, 51
xcoords, 52
ycoords, 52
overtoppinginterface::tpprofilecoordinate, 62
roughness, 62
xcoordinate, 62
zcoordinate, 63

phi
typedefinitionsrtoovertopping::tpload, 58
profileinstructure
zfunctionswtiovertopping, 45
psi
typedefinitionsrtoovertopping::tpgeometry, 56

qo
typedefinitionsrtoovertopping::tpovertopping, 61

realrootscubicfunction
formulamodulertoovertopping, 19
removeberms
geometrymodulertoovertopping, 29
removedikesegments
geometrymodulertoovertopping, 30
rfactor_max
typedefinitionsrtoovertopping, 42
rfactor_min
typedefinitionsrtoovertopping, 42
rootsdepressedcubic
formulamodulertoovertopping, 19
rootsgeneralcubic
formulamodulertoovertopping, 20
roughness
overtoppinginterface::overtoppinggeometrytype, 50
overtoppinginterface::overtoppinggeometrytypef, 51
overtoppinginterface::tpprofilecoordinate, 62
roughnessfactors
typedefinitionsrtoovertopping::tpgeometry, 56

segmentslopes
typedefinitionsrtoovertopping::tpgeometry, 56
segmenttypes
typedefinitionsrtoovertopping::tpgeometry, 56
slope_max
typedefinitionsrtoovertopping, 42
slope_min
typedefinitionsrtoovertopping, 42
splitcrosssection
geometrymodulertoovertopping, 30

tm_10
typedefinitionsrtoovertopping::tpload, 58
typeDefinitionsRTOovertopping.f90, 69
typedefinitionsrtoovertopping, 39
berm_max, 40
berm_min, 40
fb_max, 40
fb_min, 40
fn_max, 40
fn_min, 40
frunup1_max, 41
frunup1_min, 41
frunup2_max, 41
frunup2_min, 41
frunup3_max, 41
frunup3_min, 41
fs_max, 41
fs_min, 41
margindiff, 41
margingrad, 42
rfactor_max, 42
rfactor_min, 42
slope_max, 42
slope_min, 42
xdiff_min, 42
z2_iter_max, 42
z2_margin, 42
typedefinitionsrtoovertopping::overtoppingmodelfactors, 52
computedovertopping, 53
criticalovertopping, 53
factordeterminationq_b_f_b, 53
factordeterminationq_n_f_n, 53
frunup1, 53
frunup2, 54
frunup3, 54
fshallow, 54
typedefinitionsrtoovertopping::tpgeometry, 54
nbermsegments, 56
ncoordinates, 56

psi, 56
roughnessfactors, 56
segmentslopes, 56
segmenttypes, 56
xcoorddiff, 56
xcoordinates, 56
ycoorddiff, 57
ycoordinates, 57
typedefinitionsrtoovertopping::tpload, 57
h, 58
hm0, 58
phi, 58
tm_10, 58
typedefinitionsrtoovertopping::tpmodelfactors, 58
fb, 60
fn, 60
frunup1, 60
frunup2, 60
frunup3, 60
fs, 60
typedefinitionsrtoovertopping::tpovertopping, 60
qo, 61
z2, 61

varmodelfactorcriticalovertopping
overtoppinginterface, 39
versionnumber
dllovertopping, 10

writecrosssection
geometrymodulertoovertopping, 31

xcoorddiff
typedefinitionsrtoovertopping::tpgeometry, 56
xcoordinate
overtoppinginterface::tpprofilecoordinate, 62
xcoordinates
typedefinitionsrtoovertopping::tpgeometry, 56
xcoords
overtoppinginterface::overtoppinggeometrytype, 50
overtoppinginterface::overtoppinggeometrytypef, 52
xdiff_min
typedefinitionsrtoovertopping, 42

ycoorddiff
typedefinitionsrtoovertopping::tpgeometry, 57
ycoordinates
typedefinitionsrtoovertopping::tpgeometry, 57
ycoords
overtoppinginterface::overtoppinggeometrytype, 50
overtoppinginterface::overtoppinggeometrytypef, 52

z2
typedefinitionsrtoovertopping::tpovertopping, 61
z2_iter_max
typedefinitionsrtoovertopping, 42
z2_margin
typedefinitionsrtoovertopping, 42
zFunctionsWTIOvertopping.f90, 71
zcoordinate
overtoppinginterface::tpprofilecoordinate, 63
zfunclogratios
zfunctionswtiovertopping, 46
zfunctionswtiovertopping, 43
adjustprofile, 43
calculateqorto, 43
profileinstructure, 45
zfunclogratios, 46