Dikes Overtopping Kernel - Technical documentation

Generated by Doxygen 1.8.9.1

Fri Mar 25 2016 10:14:01

Contents

Chapter 1

Modules Index

1.1 Modules List

Here is a list of all modules with brief descriptions:

Chapter 2

Data Type Index

2.1 Class List

Here are the data types with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

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, verbosity, logFile)

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

  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 validateinputc (geometryInput, dikeHeight, modelFactors, success, errorText)

  Subroutine that validates the geometry Wrapper for ValidateInputFold: convert C-like input structures to Fortran input structures.
- subroutine, public validateinputf (geometryF, dikeHeight, modelFactors, errorStruct)

  Subroutine that validates the geometry.
- subroutine, public setlanguage (lang)

  Subroutine that sets the language for error and validation messages.
- subroutine, public getlanguage (lang)

  Subroutine that gets the language for error and validation messages.
- subroutine, public versionnumber (version)

  Subroutine that delivers the version number.
- type(overtoppinggeometrytypef) function geometry_c_f (geometryInput)

  Private subroutine that converts geometry from c-pointer to fortran struct.

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(tpovertoppinginput), intent(inout) *modelFactors,* type (tpovertopping), intent(out) *overtopping,* logical, intent(out) *success,* character(len=∗), intent(out) *errorText,* integer, intent(in) *verbosity,* character(len=∗), intent(in) *logFile*)

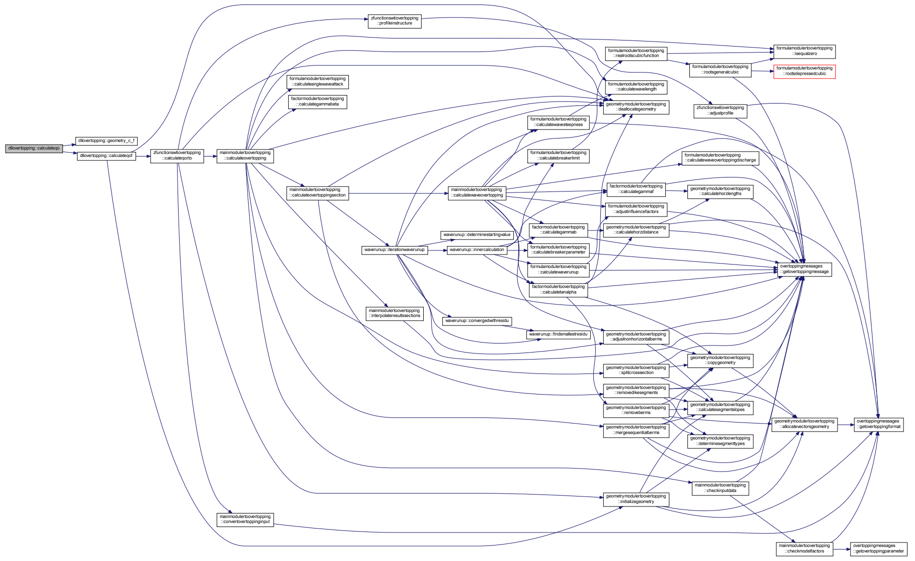
Subroutine that calculates the discharge needed for the Z-function DikesOvertopping Wrapper for calculateQoF: convert C-like input structures to Fortran input structures.

Parameters

in	*geometryinput*	struct with geometry and roughness as c-pointers
in	*load*	struct with waterlevel and wave parameters
in	*dikeheight*	dike height
in,out	*modelfactors*	struct with modelfactors
out	*overtopping*	structure with overtopping results
out	*success*	flag for success
out	*errortext*	error message (only set if not successful)
in	*verbosity*	level of verbosity
in	*logfile*	filename of logfile

Definition at line 43 of file dllOvertopping.f90.

Here is the call graph for this function:


dllovertopping::calculateqo
dllovertopping::geometry_c_f
dllovertopping::calculateqof
zfunctionswtiovertopping::calculateqorto
mainmodulertoovertopping::calculateovertopping
zfunctionswtiovertopping::profileinstructure
formulamodulertoovertopping::calculateanglewaveattack
factormodulertoovertopping::calculategammabeta
mainmodulertoovertopping::calculateovertoppingsection
waverunup::iterationwaverunup
mainmodulertoovertopping::interpolateresultssections
mainmodulertoovertopping::convertovertoppinginput
formulamodulertoovertopping::realrootscubicfunction
formulamodulertoovertopping::isequalzero
formulamodulertoovertopping::rootsgeneralcubic
formulamodulertoovertopping::rootsdepressedcubic
formulamodulertoovertopping::calculatewavelength
geometrymodulertoovertopping::deallocategeometry
zfunctionswtiovertopping::adjustprofile
formulamodulertoovertopping::calculatewavesteepness
formulamodulertoovertopping::calculatebreakerlimit
formulamodulertoovertopping::calculatewaveovertoppingdischarge
mainmodulertoovertopping::calculatewaveovertopping
factormodulertoovertopping::calculategammaf
geometrymodulertoovertopping::calculatehorzlengths
formulamodulertoovertopping::adjustinfluencefactors
waverunup::determinestartingvalue
factormodulertoovertopping::calculategammab
geometrymodulertoovertopping::calculatehorzdistance
waverunup::innercalculation
formulamodulertoovertopping::calculatebreakerparameter
formulamodulertoovertopping::calculatewaverunup
overtoppingmessages::getovertoppingmessage
factormodulertoovertopping::calculatetanalpha
waverunup::convergedwithresidu
waverunup::findsmallestresidu
geometrymodulertoovertopping::adjustnonhorizontalberms
geometrymodulertoovertopping::copygeometry
geometrymodulertoovertopping::splitcrosssection
geometrymodulertoovertopping::removedikesegments
geometrymodulertoovertopping::calculatesegmentslopes
geometrymodulertoovertopping::removeberms
geometrymodulertoovertopping::allocatevectorsgeometry
overtoppingmessages::getovertoppingformat
geometrymodulertoovertopping::mergesequentialberms
geometrymodulertoovertopping::determinesegmenttypes
mainmodulertoovertopping::checkinputdata
geometrymodulertoovertopping::initializegeometry
mainmodulertoovertopping::checkmodelfactors
overtoppingmessages::getovertoppingparameter

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(tpovertoppinginput), intent(inout) *modelFactors,* type (tpovertopping), intent(out) *overtopping,* logical, intent(out) *success,* character(len=∗), intent(out) *errorText,* type(tlogging), intent(in) *logging*)

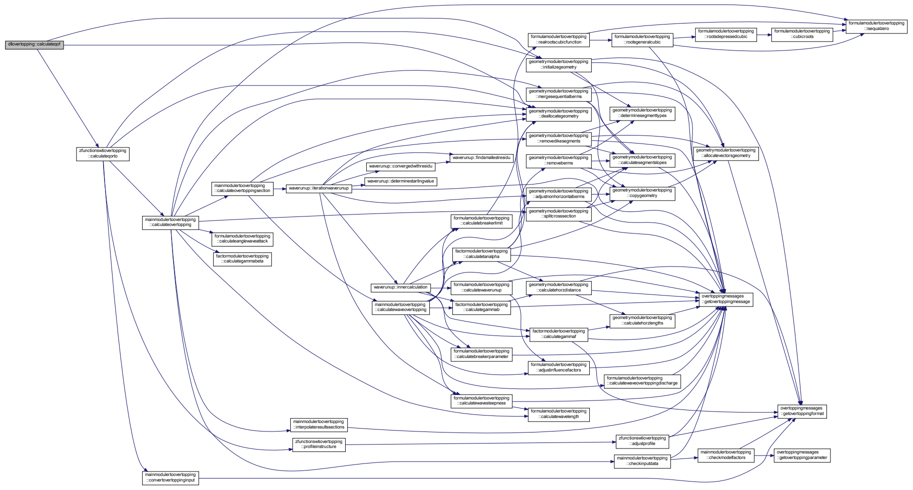
Subroutine that calculates the discharge needed for the Z-function DikesOvertopping.

Parameters

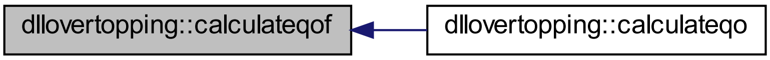
in	*geometryf*	struct with geometry and roughness
in	*load*	struct with waterlevel and wave parameters
in	*dikeheight*	dike height
in,out	*modelfactors*	struct with modelFactors
out	*overtopping*	structure with overtopping results
out	*success*	flag for success
out	*errortext*	error message (only set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

Definition at line 74 of file dllOvertopping.f90.

Here is the call graph for this function:


dllovertopping::calculateqof
zfunctionswtiovertopping ::calculateqorto
mainmodulertoovertopping ::calculateovertopping
mainmodulertoovertopping ::calculateovertoppingsection
waverunup::iterationwaverunup
waverunup::convergedwithresidu
waverunup::determinestartingvalue
waverunup::findsmallestresidu
formulamodulertoovertopping ::calculateanglewaveattack
factormodulertoovertopping ::calculategammabeta
waverunup::innercalculation
mainmodulertoovertopping ::calculatewaveovertopping
formulamodulertoovertopping ::calculatebreakerlimit
factormodulertoovertopping ::calculatetanalpha
formulamodulertoovertopping ::calculatewaverunup
factormodulertoovertopping ::calculategammab
formulamodulertoovertopping ::calculatebreakerparameter
formulamodulertoovertopping ::calculatewavesteepness
formulamodulertoovertopping ::realrootscubicfunction
formulamodulertoovertopping ::rootsgeneralcubic
formulamodulertoovertopping ::rootsdepressedcubic
formulamodulertoovertopping ::cubicroots
formulamodulertoovertopping ::isequalzero
geometrymodulertoovertopping ::initializegeometry
geometrymodulertoovertopping ::mergesequentialberms
geometrymodulertoovertopping ::deallocategeometry
geometrymodulertoovertopping ::determinesegmenttypes
geometrymodulertoovertopping ::removedikesegments
geometrymodulertoovertopping ::removeberms
geometrymodulertoovertopping ::calculatesegmentslopes
geometrymodulertoovertopping ::allocatevectorsgeometry
geometrymodulertoovertopping ::adjustnonhorizontalberms
geometrymodulertoovertopping ::copygeometry
geometrymodulertoovertopping ::splitcrosssection
geometrymodulertoovertopping ::calculatehorzdistance
overtoppingmessages ::getovertoppingmessage
geometrymodulertoovertopping ::calculatehorzlengths
factormodulertoovertopping ::calculategammaf
formulamodulertoovertopping ::adjustinfluencefactors
formulamodulertoovertopping ::calculatewaveovertoppingdischarge
formulamodulertoovertopping ::calculatewavelength
overtoppingmessages ::getovertoppingformat
mainmodulertoovertopping ::interpolateresultssections
zfunctionswtiovertopping ::profileinstructure
zfunctionswtiovertopping ::adjustprofile
mainmodulertoovertopping ::checkinputdata
mainmodulertoovertopping ::checkmodelfactors
overtoppingmessages ::getovertoppingparameter
mainmodulertoovertopping ::convertovertoppinginput

Here is the caller graph for this function:



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

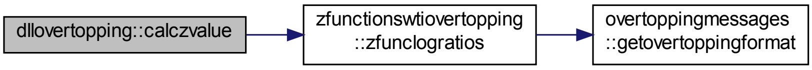
Subroutine that calculates the Z-function DikesOvertopping based on the discharge calculated with calculateQoF.

Parameters

`in`	*criticalovertoppingrate*	critical overtoppingrate
`in,out`	*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 107 of file dllOvertopping.f90.

Here is the call graph for this function:


dllovertopping::calczvalue
zfunctionswtiovertopping ::zfunclogratios
overtoppingmessages ::getovertoppingformat

4.1.2.4 type(overtoppinggeometrytypef) function dllovertopping::geometry_c_f (type(overtoppinggeometrytype), intent(in) *geometryInput*) [private]

Private subroutine that converts geometry from c-pointer to fortran struct.

Parameters

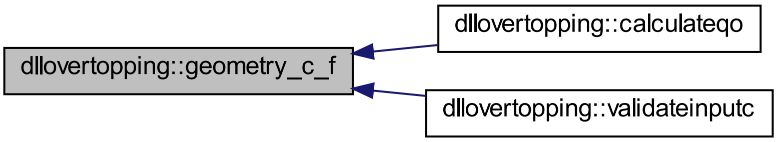
in	*geometryinput*	struct with geometry and roughness as c-pointers

Returns

fortran struct with geometry and roughness

Definition at line 310 of file dllOvertopping.f90.

Here is the caller graph for this function:


dllovertopping::geometry_c_f
dllovertopping::calculateqo
dllovertopping::validateinputc

4.1.2.5 subroutine, public dllovertopping::getlanguage (character(len=∗), intent(out) *lang*)

Subroutine that gets the language for error and validation messages.

Definition at line 276 of file dllOvertopping.f90.

4.1.2.6 subroutine, public dllovertopping::setlanguage (character(len=∗), intent(in) *lang*)

Subroutine that sets the language for error and validation messages.

Definition at line 263 of file dllOvertopping.f90.

4.1.2.7 subroutine, public dllovertopping::validateinputc (type(overtoppinggeometrytype), intent(in) *geometryInput,* real(kind=wp), intent(in) *dikeHeight,* type(tpovertoppinginput), intent(inout) *modelFactors,* logical, intent(out) *success,* character(len=∗), intent(out) *errorText*)

Subroutine that validates the geometry Wrapper for ValidateInputFold: convert C-like input structures to Fortran input structures.

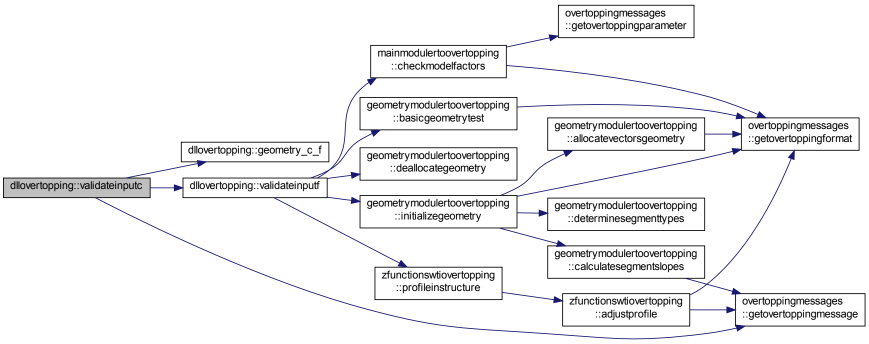
Parameters

in	*geometryinput*	struct with geometry and roughness as c-pointers
in	*dikeheight*	dike height
in,out	*modelfactors*	struct with modelfactors
out	*success*	flag for success

out	*errortext*	error message (only set if not successful)

Definition at line 127 of file dllOvertopping.f90.

Here is the call graph for this function:


overtoppingmessages ::getovertoppingparameter
mainmodulertoovertopping ::checkmodelfactors
geometrymodulertoovertopping ::basicgeometrytest
geometrymodulertoovertopping ::allocatevectorsgeometry
overtoppingmessages ::getovertoppingformat
dllovertopping::geometry_c_f
geometrymodulertoovertopping ::deallocategeometry
dllovertopping::validateinputc
dllovertopping::validateinputf
geometrymodulertoovertopping ::initializegeometry
geometrymodulertoovertopping ::determinesegmenttypes
geometrymodulertoovertopping ::calculatesegmentslopes
zfunctionswtiovertopping ::profileinstructure
zfunctionswtiovertopping ::adjustprofile
overtoppingmessages ::getovertoppingmessage

4.1.2.8 subroutine, public dllovertopping::validateinputf (type(overtoppinggeometrytypef), intent(in) *geometryF,* real(kind=wp), intent(in) *dikeHeight,* type(tpovertoppinginput), intent(inout) *modelFactors,* type(terrormessages), intent(inout) *errorStruct*)

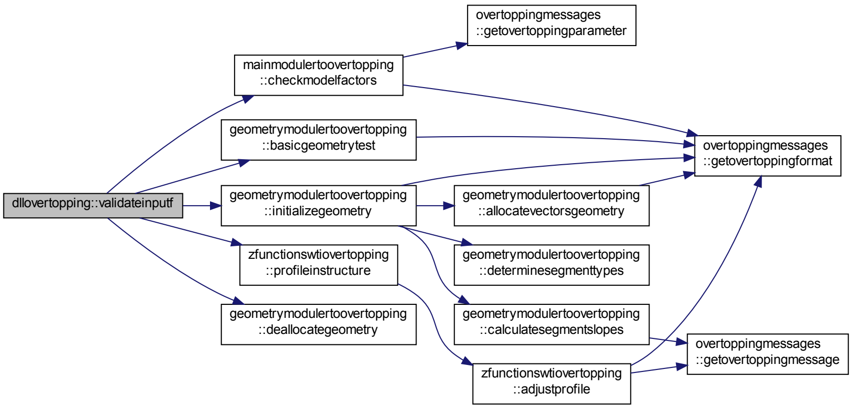
Subroutine that validates the geometry.

Parameters

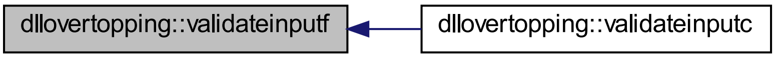
in	*geometryf*	struct with geometry and roughness
in	*dikeheight*	dike height
in,out	*modelfactors*	struct with modelFactors
in,out	*errorstruct*	error message (only set if not successful)

Definition at line 178 of file dllOvertopping.f90.

Here is the call graph for this function:


overtoppingmessages ::getovertoppingparameter
mainmodulertoovertopping ::checkmodelfactors
geometrymodulertoovertopping ::basicgeometrytest
overtoppingmessages ::getovertoppingformat
dllovertopping::validateinputf
geometrymodulertoovertopping ::initializegeometry
geometrymodulertoovertopping ::allocatevectorsgeometry
zfunctionswtiovertopping ::profileinstructure
geometrymodulertoovertopping ::determinesegmenttypes
geometrymodulertoovertopping ::deallocategeometry
geometrymodulertoovertopping ::calculatesegmentslopes
overtoppingmessages ::getovertoppingmessage
zfunctionswtiovertopping ::adjustprofile

Here is the caller graph for this function:


dllovertopping::validateinputf
dllovertopping::validateinputc

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

Subroutine that delivers the version number.

Parameters

out	*version*	version number

Definition at line 288 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, gammaB, succes, errorMessage)

  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,* real(wp), intent(out) *gammaB,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

calculateGammaB influence factor berms

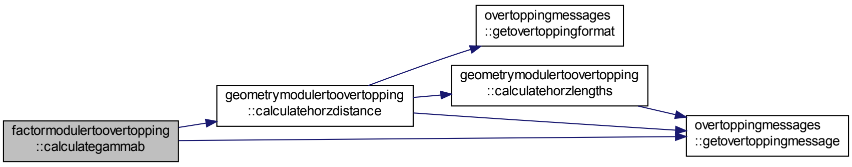
Parameters

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
out	*gammab*	influence factor berms

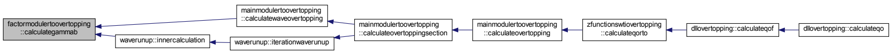
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 286 of file factorModuleRTOovertopping.f90.

Here is the call graph for this function:



Here is the caller graph for this function:


factormodulertoovertopping
::calculategammab
mainmodulertoovertopping
::calculatewaveovertopping
waverunup::innercalculation
waverunup::iterationwaverunup
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*)

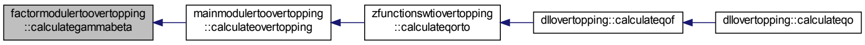
calculateGammaBeta influence factor angle of wave attack

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 115 of file factorModuleRTOovertopping.f90.

Here is the caller graph for this function:



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*)

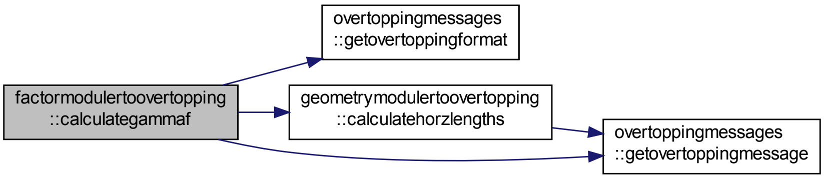
calculateGammaF influence factor roughness

Parameters

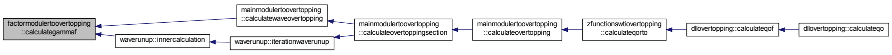
in	*h*	local water level (m+NAP)
in	*ksi0*	breaker parameter
in	*ksi0limit*	limit value breaker parameter
in	*gammab*	influence factor berms
in	*z2*	2% wave run-up (m)
in	*geometry*	structure with geometry data
out	*gammaf*	influence factor roughness
out	*succes*	flag for succes
out	*errormessage*	error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

Definition at line 155 of file factorModuleRTOovertopping.f90.

Here is the call graph for this function:



Here is the caller graph for this function:


factormodulertoovertopping
::calculategammaf
mainmodulertoovertopping
::calculatewaveovertopping
waverunup::innercalculation
waverunup::iterationwaverunup
mainmodulertoovertopping
::calculateovertoppingsection
mainmodulertoovertopping
::calculateovertopping
zfunctionswtiovertopping
::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

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*)

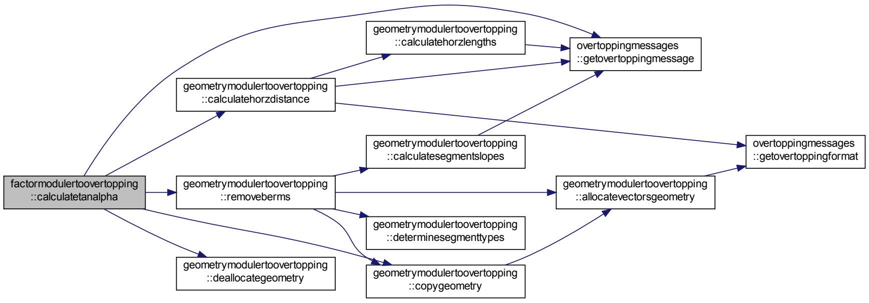
calculateTanAlpha representative slope angle

Parameters

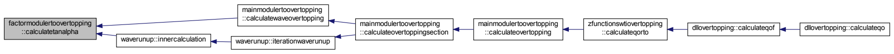
in	*h*	local water level (m+NAP)
in	*hm0*	significant wave height (m)
in	*z2*	2% wave run-up (m)
in	*geometry*	structure with geometry data
out	*tanalpha*	representative slope angle
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 36 of file factorModuleRTOovertopping.f90.

Here is the call graph for this function:


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

Here is the caller graph for this function:


factormodulertoovertopping ::calculatetanalpha
mainmodulertoovertopping ::calculatewaveovertopping
waverunup::innercalculation
waverunup::iterationwaverunup
mainmodulertoovertopping ::calculateovertoppingsection
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.3 formulamodulertoovertopping Module Reference

Functions/Subroutines

- subroutine, public calculatewaverunup (Hm0, ksi0, ksi0Limit, gammaB, gammaF, gammaBeta, modelFactors, 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 realrootscubicfunction (a, b, c, d, N, x, succes, errorMessage)

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

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

  rootsDepressedCubic: calculate the roots of a depressed cubic function
- subroutine 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*)

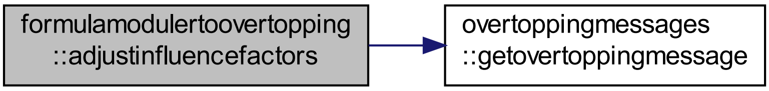
adjustInfluenceFactors: adjust the influence factors

Parameters

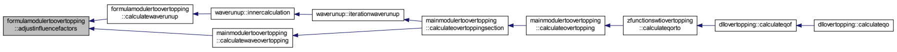
`in,out`	*gammab*	influence factor berms
`in,out`	*gammaf*	influence factor roughness
`in,out`	*gammabeta*	influence factor angle of wave attack
`in`	*gammabetatype*	type influence factor angle of wave attack: 1 = wave run-up, 2 = overtopping
`in`	*ksi0*	breaker parameter
`in`	*ksi0limit*	limit value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

Definition at line 377 of file formulaModuleRTOovertopping.f90.

Here is the call graph for this function:


formulamodulertoovertopping ::adjustinfluencefactors
overtoppingmessages ::getovertoppingmessage

Here is the caller graph for this function:


formulamodulertoovertopping ::adjustinfluencefactors
formulamodulertoovertopping ::calculatewaverunup
waverunup::innercalculation
waverunup::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*)

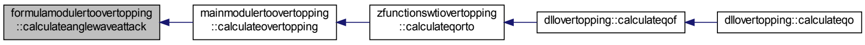
calculateAngleWaveAttack: calculate the angle of wave attack

Parameters

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

Definition at line 285 of file formulaModuleRTOovertopping.f90.

Here is the caller graph for this function:

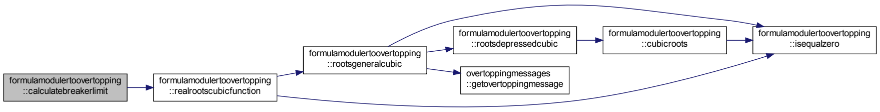

formulamodulertoovertopping
::calculateanglewaveattack
mainmodulertoovertopping
::calculateovertopping
zfunctionswtiovertopping
::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

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

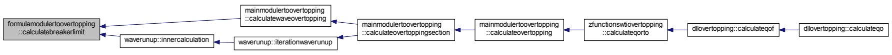
calculateBreakerLimit: calculate the breaker limit

Definition at line 308 of file formulaModuleRTOovertopping.f90.

Here is the call graph for this function:


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

Here is the caller graph for this function:


formulamodulertoovertopping
::calculatebreakerlimit
mainmodulertoovertopping
::calculatewaveovertopping
waverunup::innercalculation
waverunup::iterationwaverunup
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*)

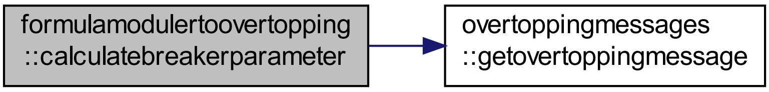
calculateBreakerParameter: calculate the breaker parameter

Parameters

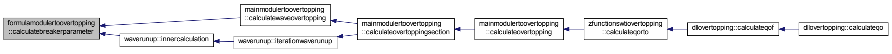
in	*tanalpha*	representative slope angle
in	*s0*	wave steepness
out	*ksi0*	breaker parameter
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 244 of file formulaModuleRTOovertopping.f90.

Here is the call graph for this function:



Here is the caller graph for this function:


formulamodulertoovertopping
::calculatebreakerparameter
mainmodulertoovertopping
::calculatewaveovertopping
waverunup::innercalculation
waverunup::iterationwaverunup
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*)

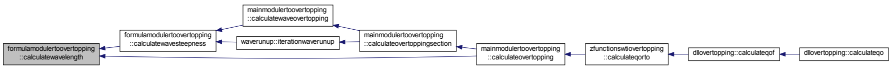
calculateWaveLength: calculate the wave length

Parameters

`in`	*tm_10*	spectral wave period (s)
`out`	*l0*	wave length (m)

Definition at line 181 of file formulaModuleRTOovertopping.f90.

Here is the caller graph for this function:


formulamodulertoovertopping
::calculatewavelength
formulamodulertoovertopping
::calculatewavesteepness
mainmodulertoovertopping
::calculatewaveovertopping
waverunup::iterationwaverunup
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(tpovertoppinginput), intent(in) *modelFactors,* real(wp), intent(out) *Qo,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

calculateWaveOvertoppingDischarge: calculate the wave overtopping discharge

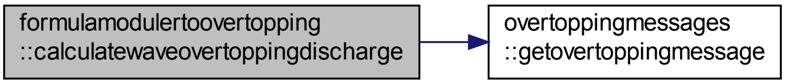
Parameters

`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

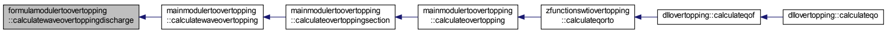
in	*gammabeta*	influence factor angle of wave attack
in	*ksi0*	breaker parameter
in	*hcrest*	crest level (m+NAP)
in	*modelfactors*	structure with model factors
out	*qo*	wave overtopping discharge (l/m per s)
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 84 of file formulaModuleRTOovertopping.f90.

Here is the call graph for this function:


formulamodulertoovertopping
::calculatewaveovertoppingdischarge
overtoppingmessages
::getovertoppingmessage

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) *ksi0,* real(wp), intent(in) *ksi0Limit,* real(wp), intent(inout) *gammaB,* real(wp), intent(inout) *gammaF,* real(wp), intent(inout) *gammaBeta,* type (tpovertoppinginput), intent(in) *modelFactors,* real(wp), intent(out) *z2,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

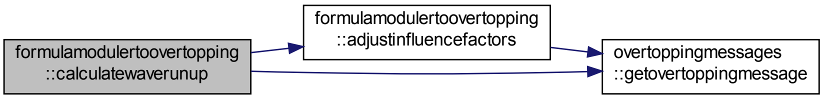
calculateWaveRunup: calculate wave runup

Parameters

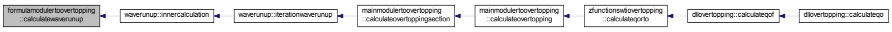
in	*hm0*	significant wave height (m)
in	*ksi0*	breaker parameter
in	*ksi0limit*	limit value breaker parameter
in,out	*gammab*	influence factor berms
in,out	*gammaf*	influence factor roughness
in,out	*gammabeta*	influence factor angle of wave attack
in	*modelfactors*	structure with model factors
out	*z2*	2% wave run-up (m)
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 35 of file formulaModuleRTOovertopping.f90.

Here is the call graph for this function:



Here is the caller graph for this function:


formulamodulertoovertopping ::calculatewaverunup
waverunup::innercalculation
waverunup::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*)

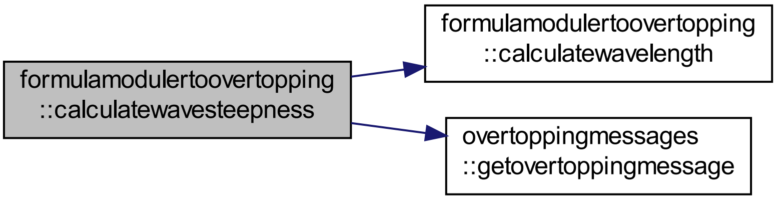
calculateWaveSteepness: calculate the wave steepness

Parameters

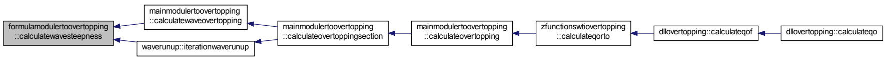
in	*hm0*	significant wave height (m)
in	*tm_10*	spectral wave period (s)
out	*s0*	wave steepness
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 202 of file formulaModuleRTOovertopping.f90.

Here is the call graph for this function:


formulamodulertoovertopping ::calculatewavesteepness
formulamodulertoovertopping ::calculatewavelength
overtoppingmessages ::getovertoppingmessage

Here is the caller graph for this function:


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

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

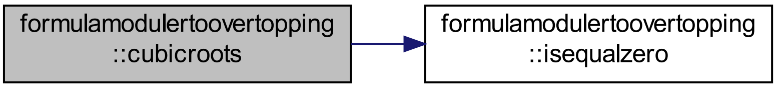
cubicRoots: calculate the roots of a cubic function

Parameters

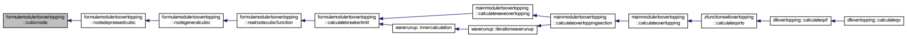
in	*z*	complex number
out	*roots*	cubic roots

Definition at line 625 of file formulaModuleRTOovertopping.f90.

Here is the call graph for this function:


formulamodulertoovertopping ::cubicroots
formulamodulertoovertopping ::isequalzero

Here is the caller graph for this function:


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

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 666 of file formulaModuleRTOovertopping.f90.

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

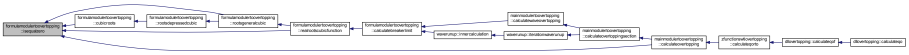
isEqualZero: is a real (almost) zero

Parameters

`in`	*x*	real number

Definition at line 690 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 ::calculatewaveovertopping
waverunup::innercalculation
waverunup::iterationwaverunup
mainmodulertoovertopping ::calculateovertoppingsection
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.3.1.12 subroutine 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*) `[private]`

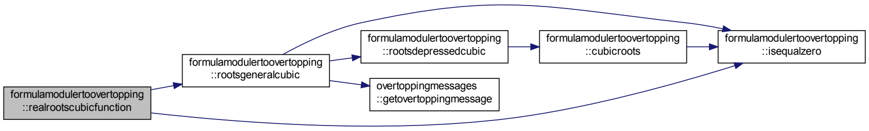
realRootsCubicFunction: calculate the roots of a cubic function

Parameters

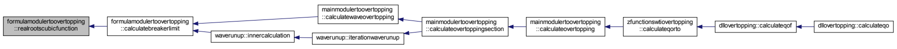
in	*a*	coefficient a cubic function
in	*b*	coefficient b cubic function
in	*c*	coefficient c cubic function
in	*d*	coefficient d cubic function
out	*n*	number of real roots cubic function
out	*x*	real roots cubic function
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 475 of file formulaModuleRTOovertopping.f90.

Here is the call graph for this function:


formulamodulertoovertopping ::realrootscubicfunction
formulamodulertoovertopping ::rootsgeneralcubic
formulamodulertoovertopping ::rootsdepressedcubic
overtoppingmessages ::getovertoppingmessage
formulamodulertoovertopping ::cubicroots
formulamodulertoovertopping ::isequalzero

Here is the caller graph for this function:


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

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

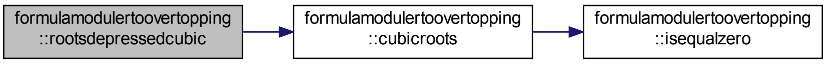
rootsDepressedCubic: calculate the roots of a depressed cubic function

Parameters

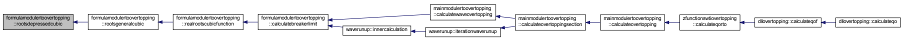
in	*p*	coefficient p depressed cubic
in	*q*	coefficient q depressed cubic
out	*z*	roots depressed cubic

Definition at line 584 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 ::calculatewaveovertopping
waverunup::innercalculation
waverunup::iterationwaverunup
mainmodulertoovertopping ::calculateovertoppingsection
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.3.1.14 subroutine 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*) `[private]`

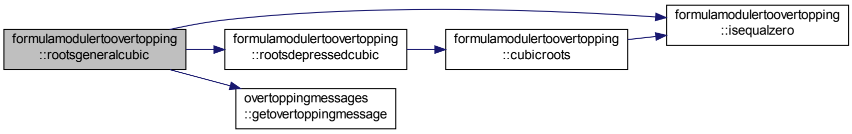
rootsGeneralCubic: calculate the roots of a generic cubic function

Parameters

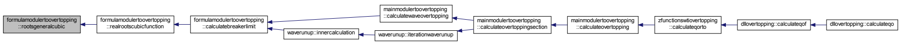
in	*a*	coefficients a cubic function
in	*b*	coefficients b cubic function
in	*c*	coefficients c cubic function
in	*d*	coefficients d cubic function
out	*z*	roots cubic function
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 531 of file formulaModuleRTOovertopping.f90.

Here is the call graph for this function:


formulamodulertoovertopping ::rootsgeneralcubic
formulamodulertoovertopping ::rootsdepressedcubic
formulamodulertoovertopping ::cubicroots
formulamodulertoovertopping ::isequalzero
overtoppingmessages ::getovertoppingmessage

Here is the caller graph for this function:


formulamodulertoovertopping ::rootsgeneralcubic
formulamodulertoovertopping ::realrootscubicfunction
formulamodulertoovertopping ::calculatebreakerlimit
mainmodulertoovertopping ::calculatewaveovertopping
waverunup::innercalculation
waverunup::iterationwaverunup
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, succes, errorMessage)

  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, succes, errorMessage)

  copyGeometry: copy a geometry structure
- 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 basicgeometrytest (geometryF, success, errorStruct)

  basicGeometryTest: test the input geometry (the adjusted geometry is checked elsewhere)

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*)

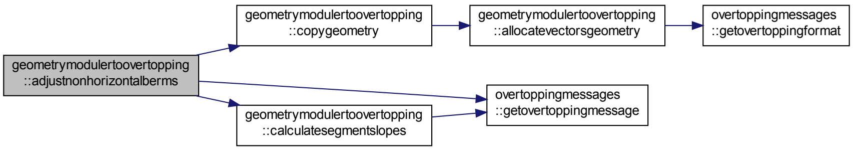
adjustNonHorizontalBerms: adjust non-horizontal berms

Parameters

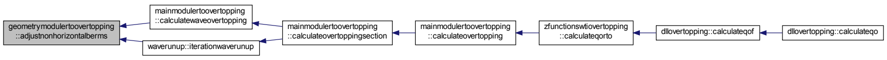
in	*geometry*	structure with geometry data
out	*geometryflatberms*	geometry data with horizontal berms
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 526 of file geometryModuleRTOovertopping.f90.

Here is the call graph for this function:


geometrymodulertoovertopping
::adjustnonhorizontalberms
geometrymodulertoovertopping
::copygeometry
geometrymodulertoovertopping
::allocatevectorsgeometry
overtoppingmessages
::getovertoppingformat
overtoppingmessages
::getovertoppingmessage
geometrymodulertoovertopping
::calculatesegmentslopes

Here is the caller graph for this function:


geometrymodulertoovertopping
::adjustnonhorizontalberms
mainmodulertoovertopping
::calculatewaveovertopping
waverunup::iterationwaverunup
mainmodulertoovertopping
::calculateovertoppingsection
mainmodulertoovertopping
::calculateovertopping
zfunctionswtiovertopping
::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.4.1.2 subroutine, public geometrymodulertoovertopping::allocatevectorsgeometry (integer, intent(in) *nCoordinates,* type (tpgeometry), intent(inout) *geometry,* logical, intent(out) *succes,* character(len=∗), intent(inout) *errorMessage*)

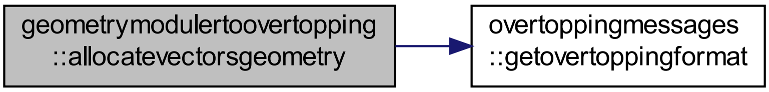
allocateVectorsGeometry: allocate the geometry vectors

Parameters

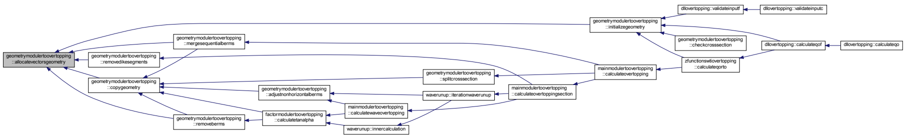
in	*ncoordinates*	number of coordinates
in,out	*geometry*	structure with geometry data
out	*succes*	succes flag
in,out	*errormessage*	error message (only set in case of error)

Definition at line 222 of file geometryModuleRTOovertopping.f90.

Here is the call graph for this function:



Here is the caller graph for this function:


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

4.4.1.3 subroutine, public geometrymodulertoovertopping::basicgeometrytest (type(overtoppinggeometrytypef), intent(in) *geometryF,* logical, intent(out) *success,* type(terrormessages), intent(inout) *errorStruct*)

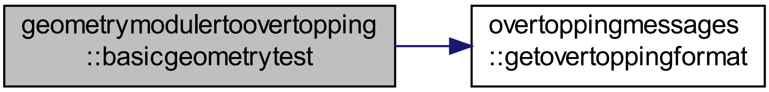
basicGeometryTest: test the input geometry (the adjusted geometry is checked elsewhere)

Parameters

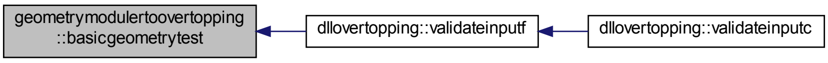
in	*geometryf*	struct with geometry and roughness
in,out	*errorstruct*	error message (only set if not successful)
out	*success*	success flag

Definition at line 1028 of file geometryModuleRTOovertopping.f90.

Here is the call graph for this function:



Here is the caller graph for this function:


geometrymodulertoovertopping
::basicgeometrytest
dllovertopping::validateinputf
dllovertopping::validateinputc

4.4.1.4 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*)

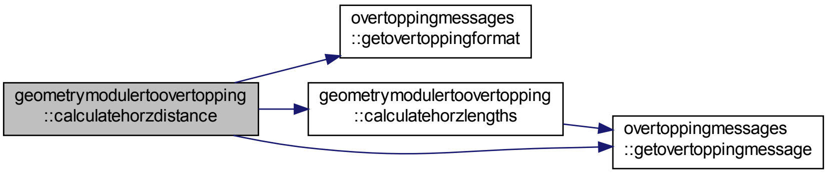
calculateHorzDistance: calculate horizontal distance

Parameters

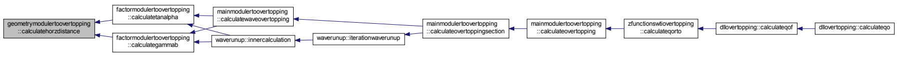
in	*geometry*	structure with geometry data
in	*ylower*	y-coordinate lower bound (m+NAP)
in	*yupper*	y-coordinate upper bound (m+NAP)
out	*dx*	horizontal distance between bounds (m)
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 979 of file geometryModuleRTOovertopping.f90.

Here is the call graph for this function:



Here is the caller graph for this function:


geometrymodulertoovertopping
::calculatehorzdistance
factormodulertoovertopping
::calculatetanalpha
factormodulertoovertopping
::calculategammab
mainmodulertoovertopping
::calculatewaveovertopping
waverunup::innercalculation
waverunup::iterationwaverunup
mainmodulertoovertopping
::calculateovertoppingsection
mainmodulertoovertopping
::calculateovertopping
zfunctionswtiovertopping
::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.4.1.5 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*)

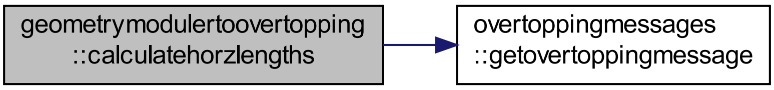
calculateHorzLengths: calculate horizontal lengths

Parameters

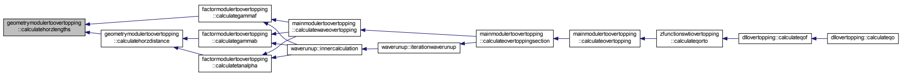
in	*geometry*	structure with geometry data
in	*ylower*	y-coord. lower bound (m+NAP)
in	*yupper*	y-coord. upper bound (m+NAP)
out	*horzlengths*	horizontal lengths segments (m)
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 883 of file geometryModuleRTOovertopping.f90.

Here is the call graph for this function:


geometrymodulertoovertopping ::calculatehorzlengths
overtoppingmessages ::getovertoppingmessage

Here is the caller graph for this function:


geometrymodulertoovertopping ::calculatehorzlengths
factormodulertoovertopping ::calculategammaf
geometrymodulertoovertopping ::calculatehorzdistance
factormodulertoovertopping ::calculategammab
factormodulertoovertopping ::calculatetanalpha
mainmodulertoovertopping ::calculatewaveovertopping
waverunup::innercalculation
waverunup::iterationwaverunup
mainmodulertoovertopping ::calculateovertoppingsection
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

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

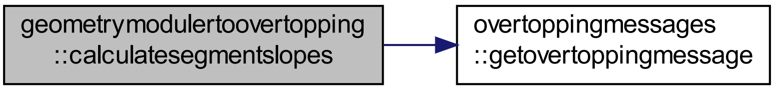
calculateSegmentSlopes: calculate the segment slopes

Parameters

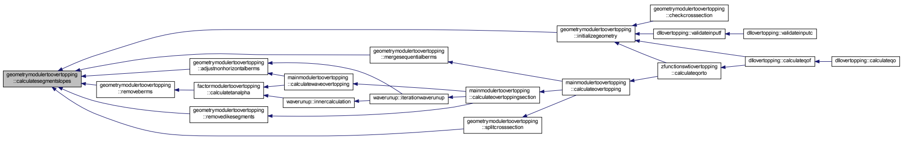
in,out	*geometry*	structure with geometry data
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 284 of file geometryModuleRTOovertopping.f90.

Here is the call graph for this function:


geometrymodulertoovertopping ::calculatesegmentslopes
overtoppingmessages ::getovertoppingmessage

Here is the caller graph for this function:


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

4.4.1.7 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*)

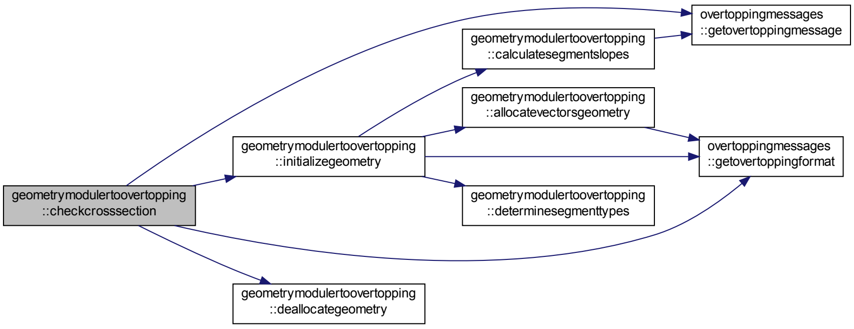
checkCrossSection: check cross section

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 35 of file geometryModuleRTOovertopping.f90.

Here is the call graph for this function:


overtoppingmessages ::getovertoppingmessage
geometrymodulertoovertopping ::calculatesegmentslopes
geometrymodulertoovertopping ::allocatevectorsgeometry
geometrymodulertoovertopping ::initializegeometry
overtoppingmessages ::getovertoppingformat
geometrymodulertoovertopping ::checkcrosssection
geometrymodulertoovertopping ::determinesegmenttypes
geometrymodulertoovertopping ::deallocategeometry

4.4.1.8 subroutine, public geometrymodulertoovertopping::copygeometry (type (tpgeometry), intent(in) *geometry,* type (tpgeometry), intent(inout) *geometryCopy,* logical, intent(out) *succes,* character(len=∗), intent(inout) *errorMessage*)

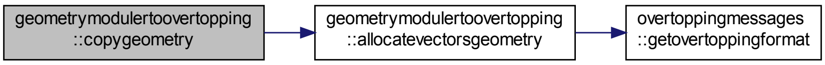
copyGeometry: copy a geometry structure

Parameters

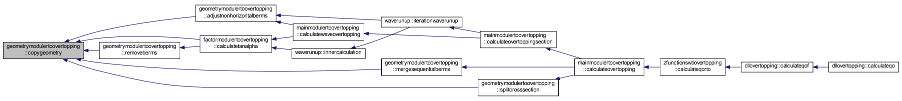
in	*geometry*	structure with geometry data
in,out	*geometrycopy*	structure with geometry data copy
out	*succes*	succes flag
in,out	*errormessage*	error message, only set in case of error

Definition at line 366 of file geometryModuleRTOovertopping.f90.

Here is the call graph for this function:


geometrymodulertoovertopping
::copygeometry
geometrymodulertoovertopping
::allocatevectorsgeometry
overtoppingmessages
::getovertoppingformat

Here is the caller graph for this function:


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

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

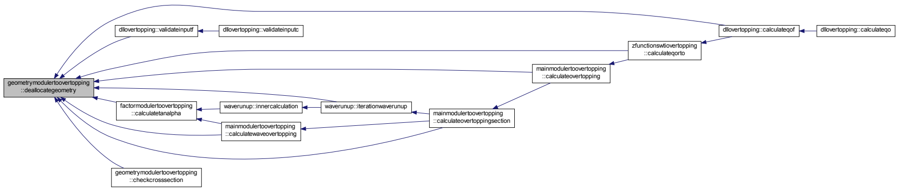
deallocateGeometry: deallocate the geometry vectors

Parameters

in,out	*geometry*	structure with geometry data

Definition at line 261 of file geometryModuleRTOovertopping.f90.

Here is the caller graph for this function:


geometrymodulertoovertopping
::deallocategeometry
dllovertopping::validateinputf
dllovertopping::validateinputc
dllovertopping::calculateqof
dllovertopping::calculateqo
zfunctionswtiovertopping
::calculateqorto
mainmodulertoovertopping
::calculateovertopping
factormodulertoovertopping
::calculatetanalpha
waverunup::innercalculation
waverunup::iterationwaverunup
mainmodulertoovertopping
::calculateovertoppingsection
mainmodulertoovertopping
::calculatewaveovertopping
geometrymodulertoovertopping
::checkcrosssection

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

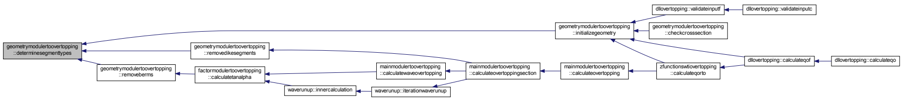
determineSegmentTypes: determine the segment types

Parameters

in,out	*geometry*	structure with geometry data

Definition at line 323 of file geometryModuleRTOovertopping.f90.

Here is the caller graph for this function:


geometrymodulertoovertopping ::determinesegmenttypes
geometrymodulertoovertopping ::removedikesegments
geometrymodulertoovertopping ::removeberms
factormodulertoovertopping ::calculatetanalpha
waverunup::innercalculation
waverunup::iterationwaverunup
mainmodulertoovertopping ::calculatewaveovertopping
mainmodulertoovertopping ::calculateovertoppingsection
mainmodulertoovertopping ::calculateovertopping
geometrymodulertoovertopping ::initializegeometry
dllovertopping::validateinputf
dllovertopping::validateinputc
geometrymodulertoovertopping ::checkcrosssection
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.4.1.11 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*)

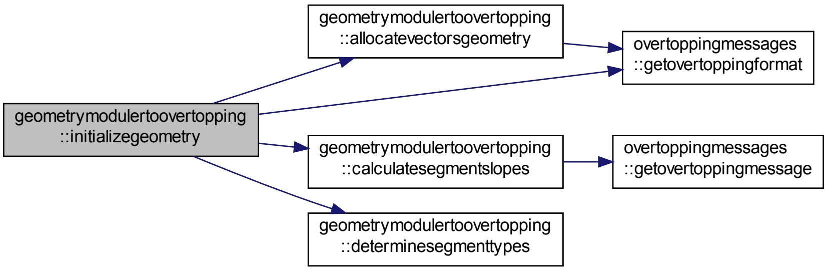
initializeGeometry: initialize the geometry

Parameters

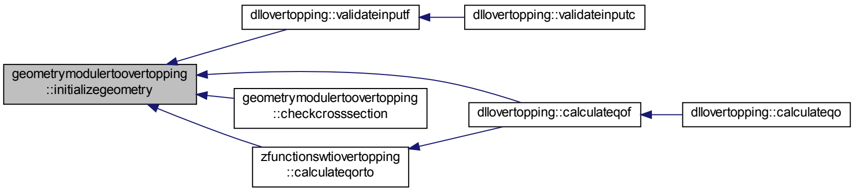
`in`	*psi*	dike normal (degree)
`in`	*ncoordinates*	number of coordinates
`in`	*xcoordinates*	x-coordinates (m)
`in`	*ycoordinates*	y-coordinates (m+NAP)
`in`	*roughnessfactors*	roughness factors
`out`	*geometry*	structure with geometry data
`out`	*succes*	flag for succes
`out`	*errormessage*	error message

Definition at line 146 of file geometryModuleRTOovertopping.f90.

Here is the call graph for this function:


geometrymodulertoovertopping ::allocatevectorsgeometry
overtoppingmessages ::getovertoppingformat
geometrymodulertoovertopping ::initializegeometry
geometrymodulertoovertopping ::calculatesegmentslopes
overtoppingmessages ::getovertoppingmessage
geometrymodulertoovertopping ::determinesegmenttypes

Here is the caller graph for this function:


dllovertopping::validateinputf
dllovertopping::validateinputc
geometrymodulertoovertopping ::initializegeometry
geometrymodulertoovertopping ::checkcrosssection
dllovertopping::calculateqof
dllovertopping::calculateqo
zfunctionswtiovertopping ::calculateqorto

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*)

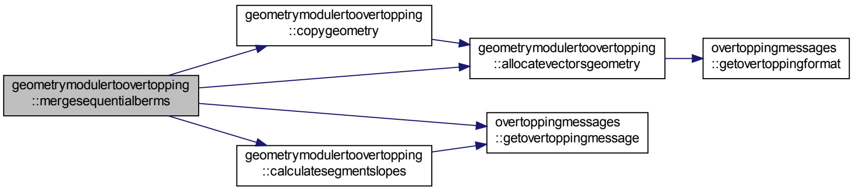
mergeSequentialBerms: merge sequential berms

Parameters

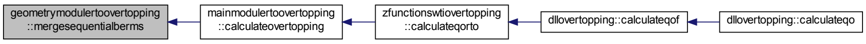
in	*geometry*	structure with geometry data
in,out	*geometrymergedberms*	geometry data with merged sequential berms
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 417 of file geometryModuleRTOovertopping.f90.

Here is the call graph for this function:


geometrymodulertoovertopping ::mergesequentialberms
geometrymodulertoovertopping ::copygeometry
geometrymodulertoovertopping ::allocatevectorsgeometry
overtoppingmessages ::getovertoppingformat
overtoppingmessages ::getovertoppingmessage
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*)

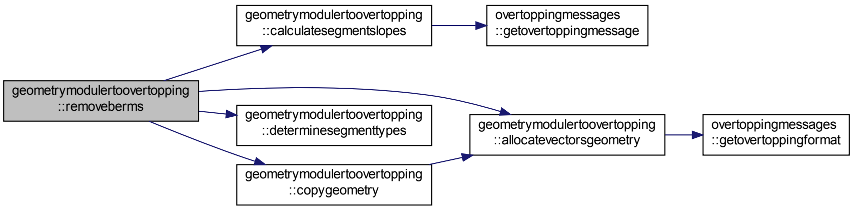
removeBerms: remove berms

Parameters

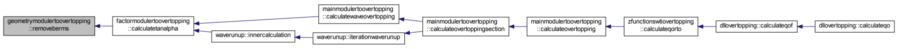
in	*geometry*	structure with geometry data
out	*geometrynoberms*	geometry data withouth berms
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 615 of file geometryModuleRTOovertopping.f90.

Here is the call graph for this function:


geometrymodulertoovertopping ::removeberms
geometrymodulertoovertopping ::calculatesegmentslopes
overtoppingmessages ::getovertoppingmessage
geometrymodulertoovertopping ::determinesegmenttypes
geometrymodulertoovertopping ::allocatevectorsgeometry
overtoppingmessages ::getovertoppingformat
geometrymodulertoovertopping ::copygeometry

Here is the caller graph for this function:


geometrymodulertoovertopping
::removeberms
factormodulertoovertopping
::calculatetanalpha
mainmodulertoovertopping
::calculatewaveovertopping
waverunup::innercalculation
waverunup::iterationwaverunup
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*)

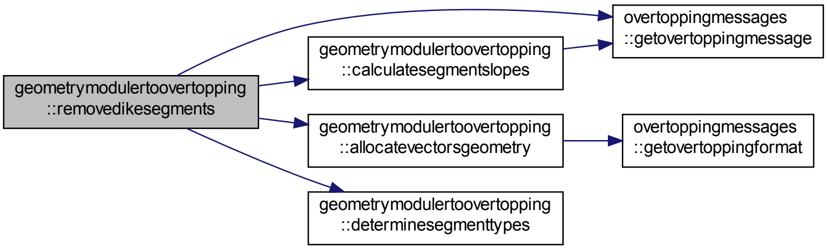
removeDikeSegments: remove dike segments

Parameters

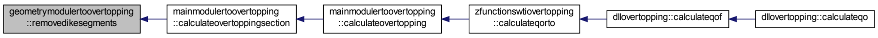
in	*geometry*	structure with geometry data
in	*index*	index starting point new cross section
out	*geometryadjusted*	geometry data with removed dike segments
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 714 of file geometryModuleRTOovertopping.f90.

Here is the call graph for this function:



Here is the caller graph for this function:



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*)

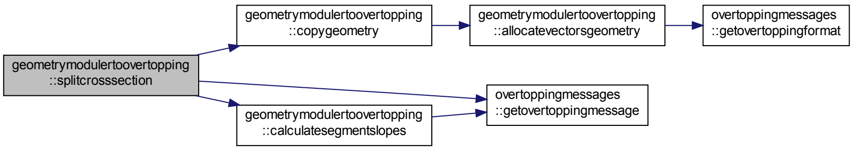
splitCrossSection: split a cross section

Parameters

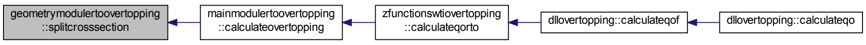
in	*geometry*	structure with geometry data
in	*l0*	wave length (m)
out	*nwideberms*	number of wide berms
out	*geometrysectionb*	geometry data with wide berms to ordinary berms
out	*geometrysectionf*	geometry data with wide berms to foreshores
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 778 of file geometryModuleRTOovertopping.f90.

Here is the call graph for this function:


geometrymodulertoovertopping ::splitcrosssection
geometrymodulertoovertopping ::copygeometry
geometrymodulertoovertopping ::allocatevectorsgeometry
overtoppingmessages ::getovertoppingformat
overtoppingmessages ::getovertoppingmessage
geometrymodulertoovertopping ::calculatesegmentslopes

Here is the caller graph for this function:


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

4.5 mainmodulertoovertopping Module Reference

Functions/Subroutines

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

  checkInputdata: check the input data
- subroutine, public checkmodelfactors (modelFactors, dimErrMessage, errorMessages, ierr)

  checkModelFactors: check the input data
- subroutine, public convertovertoppinginput (modelFactors, success, errorMessage)

  convertOvertoppingInput: convert the model factors from C-like to Fortran

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 (tpovertoppinginput), intent(in) *modelFactors,* type (tpovertopping), intent(out) *overtopping,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

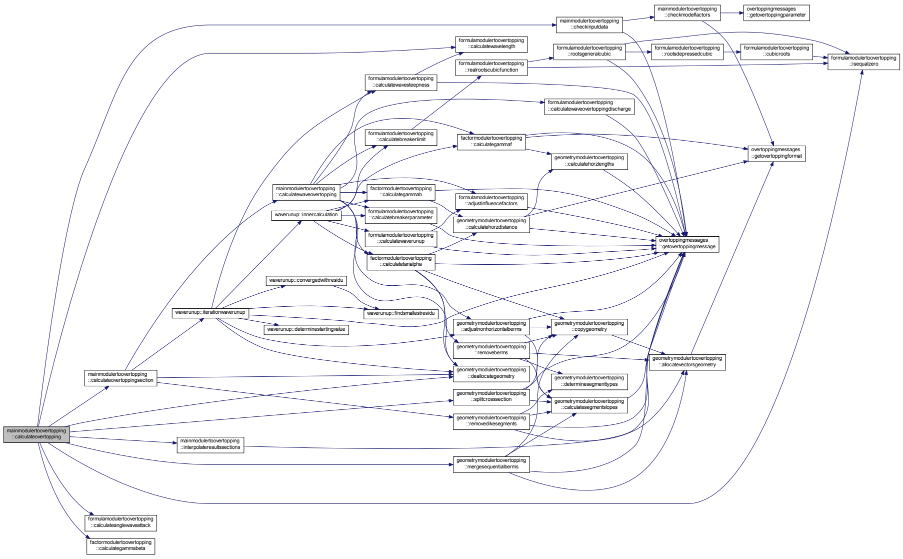
calculateOvertopping: calculate the overtopping

Parameters

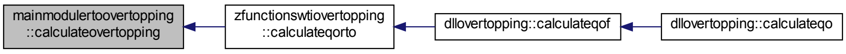
in	*geometry*	structure with geometry data
in	*load*	structure with load parameters
in	*modelfactors*	structure with model factors
out	*overtopping*	structure with overtopping results
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 37 of file mainModuleRTOovertopping.f90.

Here is the call graph for this function:


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

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]

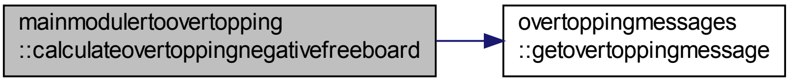
calculateOvertoppingNegativeFreeboard: calculate overtopping in case of negative freeboard

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 479 of file mainModuleRTOovertopping.f90.

Here is the call graph for this function:



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 (tpovertoppinginput), intent(in) *modelFactors*, type (tpovertopping), intent(out) *overtopping*, logical, intent(out) *succes*, character(len=∗), intent(out) *errorMessage*)

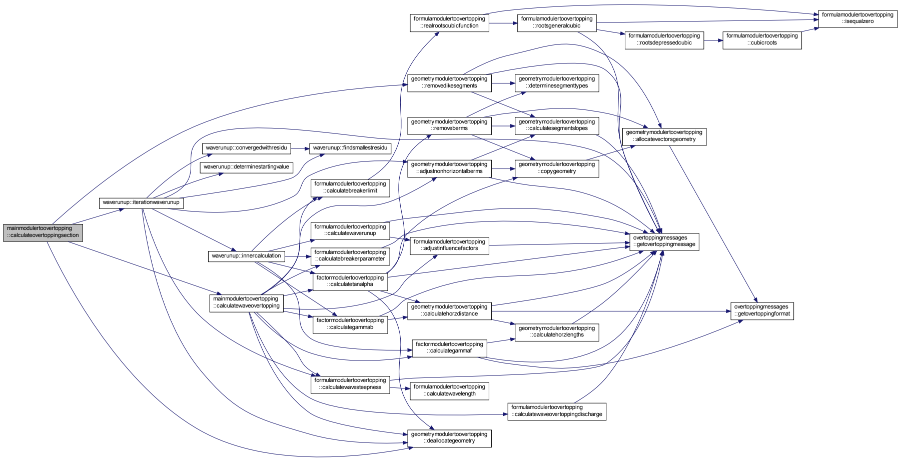
calculateOvertoppingSection: calculate the overtopping for a section

Parameters

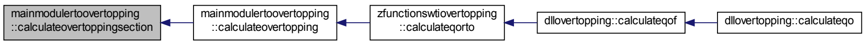
in	*geometry*	structure with geometry data
in	*h*	local water level (m+NAP)
in	*hm0*	significant wave height (m)
in	*tm_10*	spectral wave period (s)
in	*l0*	wave length (m)
in,out	*gammabeta_z*	influence angle wave attack wave run-up
in,out	*gammabeta_o*	influence angle wave attack overtopping
in	*modelfactors*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

Definition at line 166 of file mainModuleRTOovertopping.f90.

Here is the call graph for this function:


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

Here is the caller graph for this function:



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 (tpovertoppinginput), intent(in) *modelFactors,* real(wp), intent(out) *Qo,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

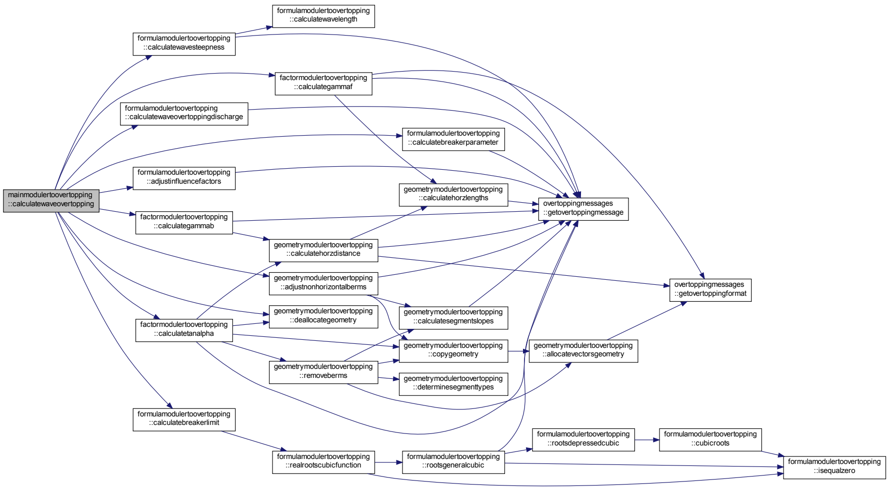
calculateWaveOvertopping: calculate wave overtopping

Parameters

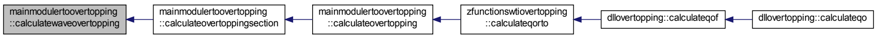
in	*geometry*	structure with geometry data
in	*h*	local water level (m+NAP)
in	*hm0*	significant wave height (m)
in	*tm_10*	spectral wave period (s)
in	*z2*	2% wave run-up (m)
in,out	*gammabeta_o*	influence angle wave attack overtopping
in	*modelfactors*	structure with model factors
out	*qo*	wave overtopping discharge (m3/m per s)
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 392 of file mainModuleRTOovertopping.f90.

Here is the call graph for this function:


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

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 (tpovertoppinginput), intent(in) *modelFactors,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

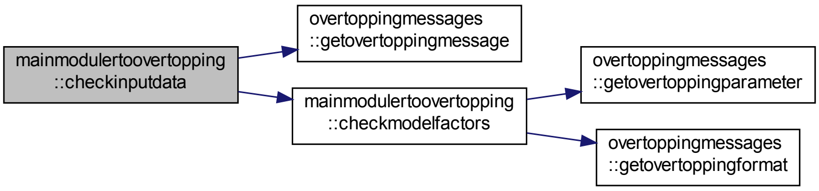
checkInputdata: check the input data

Parameters

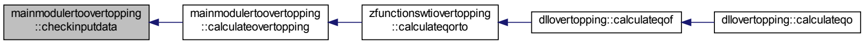
in	*geometry*	structure with geometry data
in	*load*	structure with load parameters
in	*modelfactors*	structure with model factors
out	*succes*	flag for succes
out	*errormessage*	error message

Definition at line 596 of file mainModuleRTOovertopping.f90.

Here is the call graph for this function:


overtoppingmessages ::getovertoppingmessage
mainmodulertoovertopping ::checkinputdata
overtoppingmessages ::getovertoppingparameter
mainmodulertoovertopping ::checkmodelfactors
overtoppingmessages ::getovertoppingformat

Here is the caller graph for this function:



4.5.1.6 subroutine, public mainmodulertoovertopping::checkmodelfactors (type (tpovertoppinginput), intent(in) *modelFactors,* integer, intent(in) *dimErrMessage,* character(len=∗), dimension(dimerrmessage), intent(out) *errorMessages,* integer, intent(out) *ierr*)

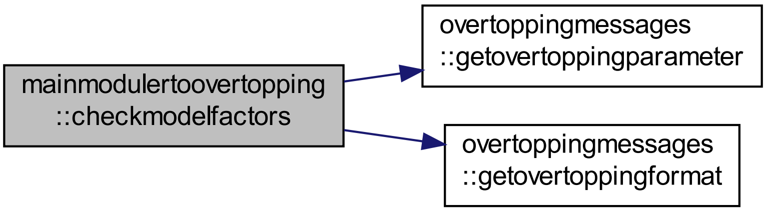
checkModelFactors: check the input data

Parameters

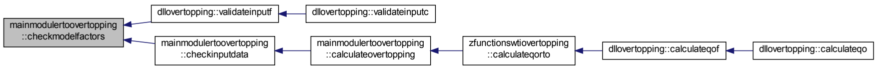
in	*modelfactors*	structure with model factors
in	*dimerrmessage*	max. number of error messages
out	*ierr*	number of errors found
out	*errormessages*	error message

Definition at line 655 of file mainModuleRTOovertopping.f90.

Here is the call graph for this function:



Here is the caller graph for this function:



4.5.1.7 subroutine, public mainmodulertoovertopping::convertovertoppinginput (type (tpovertoppinginput), intent(inout) *modelFactors,* logical, intent(out) *success,* character(len=∗), intent(inout) *errorMessage*)

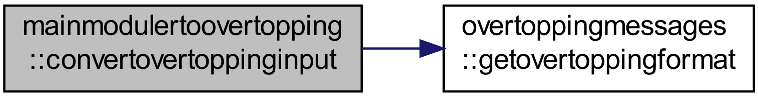
convertOvertoppingInput: convert the model factors from C-like to Fortran

Parameters

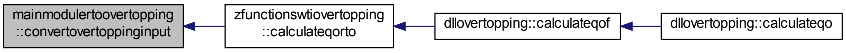
in,out	*modelfactors*	model factors and other input for overtopping
out	*success*	flag for success
in,out	*errormessage*	error message; only set when not successful

Definition at line 751 of file mainModuleRTOovertopping.f90.

Here is the call graph for this function:



Here is the caller graph for this function:


mainmodulertoovertopping
::convertovertoppinginput
zfunctionswtiovertopping
::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

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*)

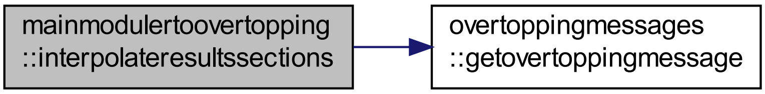
interpolateResultsSections: interpolate results for split cross sections

Parameters

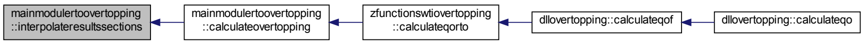
in	*geometry*	structure with geometry data
in	*l0*	wave length (m)
in	*nwideberms*	number of wide berms
in	*overtoppingb*	structure with overtopping results ordinary berms
in	*overtoppingf*	structure with overtopping results foreshores
out	*overtopping*	structure with combined overtopping results
out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

Definition at line 515 of file mainModuleRTOovertopping.f90.

Here is the call graph for this function:



Here is the caller graph for this function:


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

4.6 modulelogging Module Reference

Data Types

- type tlogging

  TLogging: structure for steering the logging.

Variables

- integer, parameter maxfilenamelength = 256

  maximum length of filename
- type(tlogging) currentlogging

  copy of argument logging

4.6.1 Variable Documentation

4.6.1.1 type(tlogging) modulelogging::currentlogging

copy of argument logging

Definition at line 21 of file ModuleLogging.f90.

4.6.1.2 integer, parameter modulelogging::maxfilenamelength = 256

maximum length of filename

Definition at line 13 of file ModuleLogging.f90.

4.7 overtoppinginterface Module Reference

Data Types

- type overtoppinggeometrytype
- type overtoppinggeometrytypef
- type tpprofilecoordinate

Variables

- integer, parameter, public varmodelfactorcriticalovertopping = 8

  Model factor critical overtopping.

4.7.1 Variable Documentation

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

Model factor critical overtopping.

Definition at line 17 of file overtoppingInterface.f90.

4.8 overtoppingmessages Module Reference

Functions/Subroutines

- subroutine setlanguage (lang)

  IDs for the strings in this module:
- subroutine getlanguage (lang)

  Subroutine that gets the language for error and validation messages.
- character(len=maxmsg) function getovertoppingmessage (ID)

  Subroutine that returns a message with the corresponding ID in the current language.
- character(len=maxmsg) function getovertoppingformat (ID)

  Subroutine that returns a Fortran format string with the corresponding ID in the current language.
- character(len=maxpar) function getovertoppingparameter (ID)

  Subroutine that returns the name of an input parameter with the corresponding ID in the current language.

Variables

- integer, parameter, private maxmsg = 128

  Module for the messages in the overtopping dll, in Dutch or English.
- integer, parameter, private maxpar =32
- character(len=2), private language = 'NL'

  default : Dutch

4.8.1 Function/Subroutine Documentation

4.8.1.1 subroutine overtoppingmessages::getlanguage (character(len=∗), intent(out) *lang*)

Subroutine that gets the language for error and validation messages.

Parameters

out	*lang*	filled with current language ID

Definition at line 101 of file OvertoppingMessages.f90.

4.8.1.2 character(len=maxmsg) function overtoppingmessages::getovertoppingformat (integer, intent(in) *ID*)

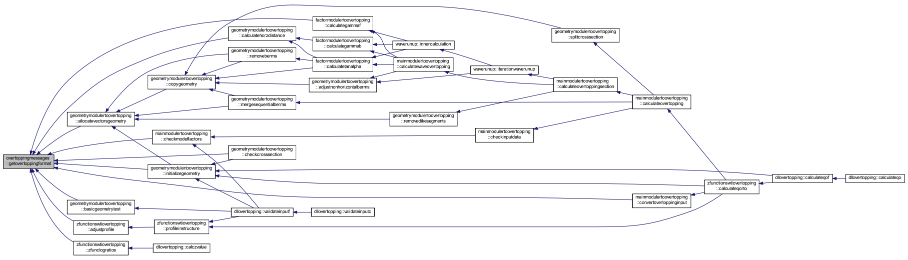
Subroutine that returns a Fortran format string with the corresponding ID in the current language.

Parameters

in	*id*	identification number of string

Definition at line 261 of file OvertoppingMessages.f90.

Here is the caller graph for this function:


overtoppingmessages
::getovertoppingformat
geometrymodulertoovertopping
::allocatevectorsgeometry
geometrymodulertoovertopping
::copygeometry
geometrymodulertoovertopping
::calculatehorzdistance
geometrymodulertoovertopping
::removeberms
factormodulertoovertopping
::calculategammaf
factormodulertoovertopping
::calculategammab
factormodulertoovertopping
::calculatetanalpha
geometrymodulertoovertopping
::adjustnonhorizontalberms
geometrymodulertoovertopping
::mergesequentialberms
waverunup::innercalculation
mainmodulertoovertopping
::calculatewaveovertopping
waverunup::iterationwaverunup
geometrymodulertoovertopping
::splitcrosssection
mainmodulertoovertopping
::calculateovertoppingsection
mainmodulertoovertopping
::calculateovertopping
geometrymodulertoovertopping
::removedikesegments
mainmodulertoovertopping
::checkinputdata
mainmodulertoovertopping
::checkmodelfactors
geometrymodulertoovertopping
::checkcrosssection
geometrymodulertoovertopping
::initializegeometry
dllovertopping::calculateqof
dllovertopping::calculateqo
zfunctionswtiovertopping
::calculateqorto
mainmodulertoovertopping
::convertovertoppinginput
geometrymodulertoovertopping
::basicgeometrytest
dllovertopping::validateinputf
dllovertopping::validateinputc
zfunctionswtiovertopping
::adjustprofile
zfunctionswtiovertopping
::profileinstructure
zfunctionswtiovertopping
::zfunclogratios
dllovertopping::calczvalue

4.8.1.3 character(len=maxmsg) function overtoppingmessages::getovertoppingmessage (integer, intent(in) *ID*)

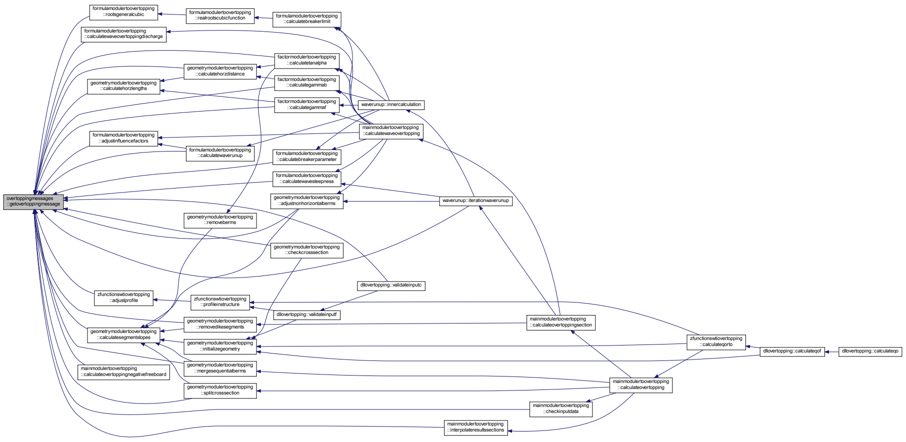
Subroutine that returns a message with the corresponding ID in the current language.

Parameters

in	*id*	identification number of string

Definition at line 111 of file OvertoppingMessages.f90.

Here is the caller graph for this function:


overtoppingmessages
::getovertoppingmessage
formulamodulertoovertopping
::rootsgeneralcubic
formulamodulertoovertopping
::realrootscubicfunction
formulamodulertoovertopping
::calculatebreakerlimit
formulamodulertoovertopping
::calculatewaveovertoppingdischarge
factormodulertoovertopping
::calculatetanalpha
geometrymodulertoovertopping
::calculatehorzdistance
geometrymodulertoovertopping
::calculatehorzlengths
factormodulertoovertopping
::calculategammab
factormodulertoovertopping
::calculategammaf
waverunup::innercalculation
mainmodulertoovertopping
::calculatewaveovertopping
formulamodulertoovertopping
::adjustinfluencefactors
formulamodulertoovertopping
::calculatewaverunup
formulamodulertoovertopping
::calculatebreakerparameter
formulamodulertoovertopping
::calculatewavesteepness
geometrymodulertoovertopping
::adjustnonhorizontalberms
waverunup::iterationwaverunup
geometrymodulertoovertopping
::removeberms
geometrymodulertoovertopping
::checkcrosssection
dllovertopping::validateinputc
zfunctionswtiovertopping
::adjustprofile
zfunctionswtiovertopping
::profileinstructure
dllovertopping::validateinputf
geometrymodulertoovertopping
::removedikesegments
mainmodulertoovertopping
::calculateovertoppingsection
geometrymodulertoovertopping
::calculatesegmentslopes
geometrymodulertoovertopping
::initializegeometry
zfunctionswtiovertopping
::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo
mainmodulertoovertopping
::calculateovertoppingnegativefreeboard
geometrymodulertoovertopping
::mergesequentialberms
geometrymodulertoovertopping
::splitcrosssection
mainmodulertoovertopping
::calculateovertopping
mainmodulertoovertopping
::checkinputdata
mainmodulertoovertopping
::interpolateresultssections

4.8.1.4 character(len=maxpar) function overtoppingmessages::getovertoppingparameter (integer, intent(in) *ID*)

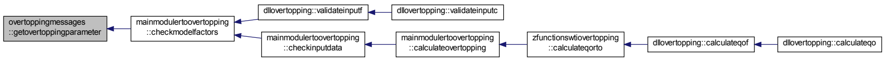
Subroutine that returns the name of an input parameter with the corresponding ID in the current language.

Parameters

in	*id*	identification number of string

Definition at line 323 of file OvertoppingMessages.f90.

Here is the caller graph for this function:



4.8.1.5 subroutine overtoppingmessages::setlanguage (character(len=*), intent(in) *lang*)

IDs for the strings in this module:

Subroutine that sets the language for error and validation messages only strings 'NL' and 'UK' are recoqnized (lower and upper case)

Parameters

in	*lang*	new language ID to be used

Definition at line 83 of file OvertoppingMessages.f90.

4.8.2 Variable Documentation

4.8.2.1 character(len=2), private overtoppingmessages::language = 'NL'

default : Dutch

Definition at line 17 of file OvertoppingMessages.f90.

4.8.2.2 integer, parameter, private overtoppingmessages::maxmsg = 128

Module for the messages in the overtopping dll, in Dutch or English.

Definition at line 15 of file OvertoppingMessages.f90.

4.8.2.3 integer, parameter, private overtoppingmessages::maxpar =32

Definition at line 15 of file OvertoppingMessages.f90.

4.9 typedefinitionsrtoovertopping Module Reference

Data Types

- type tpgeometry

  tpGeometry: structure with geometry data
- type tpload

  tpLoad: structure with load parameters
- type tpovertopping

  tpOvertopping: structure with overtopping results
- type tpovertoppinginput

  OvertoppingModelFactors: C-structure with model factors.

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 mz2_min = 0.0d0

  minimal value model factor of 2% runup height
- real(wp), parameter mz2_max = huge(mz2_max)

  maximal value model factor of 2% runup height
- real(wp), parameter frunup1_min = 0.0d0

  minimal value model factor 1 for wave run-up
- real(wp), parameter frunup1_max = huge(fRunup1_max)

  maximal value model factor 1 for wave run-up
- real(wp), parameter frunup2_min = 0.0d0

  minimal value model factor 2 for wave run-up
- real(wp), parameter frunup2_max = huge(fRunup2_max)

  maximal value model factor 2 for wave run-up
- real(wp), parameter frunup3_min = 0.0d0

  minimal value model factor 3 for wave run-up
- real(wp), parameter frunup3_max = huge(fRunup3_max)

  maximal value model factor 3 for wave run-up
- real(wp), parameter fb_min = 0.0d0

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

  maximal value model factor for breaking waves
- real(wp), parameter fn_min = 0.0d0

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

  maximal value model factor for non-breaking waves
- real(wp), parameter fs_min = 0.0d0

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

  maximal value model factor for shallow waves
- real(wp), parameter foreshore_min = 0.3d0

  minimal value reduction factor foreshore
- real(wp), parameter foreshore_max = 1.0d0

  maximal value reduction factor foreshore
- integer, parameter z2_iter_max1 = 49

  maximal number of iterations for calculation z2 part 1
- integer, parameter z2_iter_max2 = 70

  maximal number of iterations for calculation z2 part 1 & 2
- real(wp), parameter z2_margin = 0.001d0

  margin for convergence criterium calculation z2

4.9.1 Variable Documentation

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

maximal value gradient berm segment

Definition at line 67 of file typeDefinitionsRTOovertopping.f90.

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

minimal value gradient berm segment

Definition at line 66 of file typeDefinitionsRTOovertopping.f90.

4.9.1.3 real(wp), parameter typedefinitionsrtoovertopping::fb_max = huge(fB_max)

maximal value model factor for breaking waves

Definition at line 82 of file typeDefinitionsRTOovertopping.f90.

4.9.1.4 real(wp), parameter typedefinitionsrtoovertopping::fb_min = 0.0d0

minimal value model factor for breaking waves

Definition at line 81 of file typeDefinitionsRTOovertopping.f90.

4.9.1.5 real(wp), parameter typedefinitionsrtoovertopping::fn_max = huge(fN_max)

maximal value model factor for non-breaking waves

Definition at line 84 of file typeDefinitionsRTOovertopping.f90.

4.9.1.6 real(wp), parameter typedefinitionsrtoovertopping::fn_min = 0.0d0

minimal value model factor for non-breaking waves

Definition at line 83 of file typeDefinitionsRTOovertopping.f90.

4.9.1.7 real(wp), parameter typedefinitionsrtoovertopping::foreshore_max = 1.0d0

maximal value reduction factor foreshore

Definition at line 88 of file typeDefinitionsRTOovertopping.f90.

4.9.1.8 real(wp), parameter typedefinitionsrtoovertopping::foreshore_min = 0.3d0

minimal value reduction factor foreshore

Definition at line 87 of file typeDefinitionsRTOovertopping.f90.

4.9.1.9 real(wp), parameter typedefinitionsrtoovertopping::frunup1_max = huge(fRunup1_max)

maximal value model factor 1 for wave run-up

Definition at line 76 of file typeDefinitionsRTOovertopping.f90.

4.9.1.10 real(wp), parameter typedefinitionsrtoovertopping::frunup1_min = 0.0d0

minimal value model factor 1 for wave run-up

Definition at line 75 of file typeDefinitionsRTOovertopping.f90.

4.9.1.11 real(wp), parameter typedefinitionsrtoovertopping::frunup2_max = huge(fRunup2_max)

maximal value model factor 2 for wave run-up

Definition at line 78 of file typeDefinitionsRTOovertopping.f90.

4.9.1.12 real(wp), parameter typedefinitionsrtoovertopping::frunup2_min = 0.0d0

minimal value model factor 2 for wave run-up

Definition at line 77 of file typeDefinitionsRTOovertopping.f90.

4.9.1.13 real(wp), parameter typedefinitionsrtoovertopping::frunup3_max = huge(fRunup3_max)

maximal value model factor 3 for wave run-up

Definition at line 80 of file typeDefinitionsRTOovertopping.f90.

4.9.1.14 real(wp), parameter typedefinitionsrtoovertopping::frunup3_min = 0.0d0

minimal value model factor 3 for wave run-up

Definition at line 79 of file typeDefinitionsRTOovertopping.f90.

4.9.1.15 real(wp), parameter typedefinitionsrtoovertopping::fs_max = huge(fS_max)

maximal value model factor for shallow waves

Definition at line 86 of file typeDefinitionsRTOovertopping.f90.

4.9.1.16 real(wp), parameter typedefinitionsrtoovertopping::fs_min = 0.0d0

minimal value model factor for shallow waves

Definition at line 85 of file typeDefinitionsRTOovertopping.f90.

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

margin for minimal distance (m)

Definition at line 65 of file typeDefinitionsRTOovertopping.f90.

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

margin for minimal and maximal gradients

Definition at line 70 of file typeDefinitionsRTOovertopping.f90.

4.9.1.19 real(wp), parameter typedefinitionsrtoovertopping::mz2_max = huge(mz2_max)

maximal value model factor of 2% runup height

Definition at line 74 of file typeDefinitionsRTOovertopping.f90.

4.9.1.20 real(wp), parameter typedefinitionsrtoovertopping::mz2_min = 0.0d0

minimal value model factor of 2% runup height

Definition at line 73 of file typeDefinitionsRTOovertopping.f90.

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

maximal value roughness factor dike segments

Definition at line 72 of file typeDefinitionsRTOovertopping.f90.

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

minimal value roughness factor dike segments

Definition at line 71 of file typeDefinitionsRTOovertopping.f90.

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

maximal value gradient slope segment

Definition at line 69 of file typeDefinitionsRTOovertopping.f90.

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

minimal value gradient slope segment

Definition at line 68 of file typeDefinitionsRTOovertopping.f90.

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

minimal value distance between x-coordinates (m)

Definition at line 64 of file typeDefinitionsRTOovertopping.f90.

4.9.1.26 integer, parameter typedefinitionsrtoovertopping::z2_iter_max1 = 49

maximal number of iterations for calculation z2 part 1

Definition at line 89 of file typeDefinitionsRTOovertopping.f90.

4.9.1.27 integer, parameter typedefinitionsrtoovertopping::z2_iter_max2 = 70

maximal number of iterations for calculation z2 part 1 & 2

Definition at line 90 of file typeDefinitionsRTOovertopping.f90.

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

margin for convergence criterium calculation z2

Definition at line 91 of file typeDefinitionsRTOovertopping.f90.

4.10 waverunup Module Reference

Functions/Subroutines

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

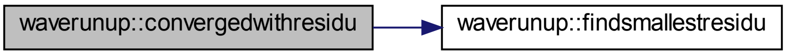
  iterationWaveRunup: iteration for the wave runup
- real(kind=wp) function innercalculation (geometry, h, Hm0, gammaBeta_z, modelFactors, z2, s0, geometry↵FlatBerms, succes, errorMessage)
- real(kind=wp) function determinestartingvalue (i, relaxationFactor, z2_start, z2_end, Hm0)
- integer function findsmallestresidu (z2_start, z2_end, n)
- subroutine convergedwithresidu (z2_start, z2_end)

4.10.1 Function/Subroutine Documentation

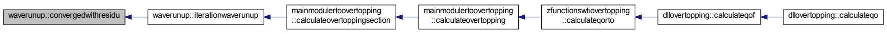
4.10.1.1 subroutine waverunup::convergedwithresidu (real(kind=wp), dimension(:), intent(in) *z2_start,* real(kind=wp), dimension(:), intent(inout) *z2_end*) [private]

Definition at line 317 of file waveRunup.f90.

Here is the call graph for this function:



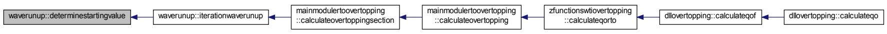
Here is the caller graph for this function:


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

4.10.1.2 real(kind=wp) function waverunup::determinestartingvalue (integer, intent(in) *i,* real(kind=wp), intent(in) *relaxationFactor,* real(kind=wp), dimension(:), intent(in) *z2_start,* real(kind=wp), dimension(:), intent(in) *z2_end,* real(kind=wp), intent(in) *Hm0*) [private]

Definition at line 266 of file waveRunup.f90.

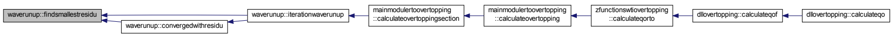
Here is the caller graph for this function:


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

4.10.1.3 integer function waverunup::findsmallestresidu (real(kind=wp), dimension(:), intent(in) *z2_start,* real(kind=wp), dimension(:), intent(in) *z2_end,* integer, intent(in), optional *n*) `[private]`

Definition at line 288 of file waveRunup.f90.

Here is the caller graph for this function:


waverunup::findsmallestresidu
waverunup::convergedwithresidu
waverunup::iterationwaverunup
mainmodulertoovertopping ::calculateovertoppingsection
mainmodulertoovertopping ::calculateovertopping
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.10.1.4 real(kind=wp) function waverunup::innercalculation (type (tpgeometry), intent(in) *geometry,* real(wp), intent(in) *h,* real(wp), intent(in) *Hm0,* real(wp), intent(inout) *gammaBeta_z,* type (tpovertoppinginput), intent(in) *modelFactors,* real(wp), intent(in) *z2,* real(kind=wp), intent(in) *s0,* type (tpgeometry), intent(in) *geometryFlatBerms,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*) `[private]`

Parameters

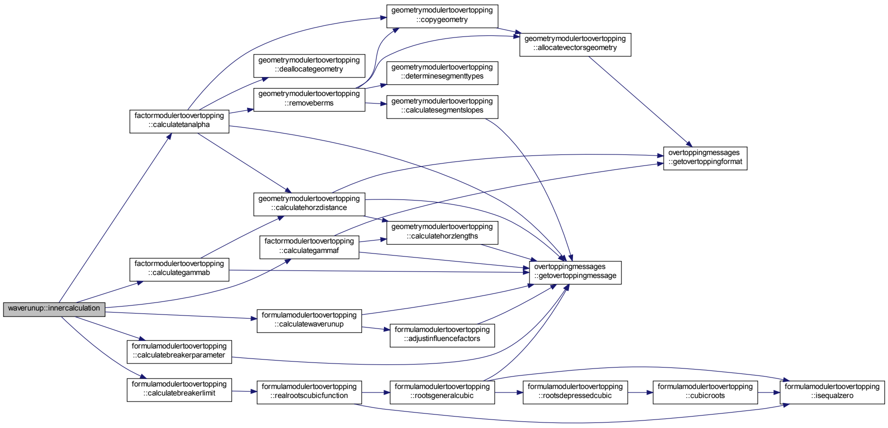
`in`	*geometry*	structure with geometry data
`in`	*h*	local water level (m+NAP)
`in`	*hm0*	significant wave height (m)
`in,out`	*gammabeta_z*	influence factor angle wave attack 2% run-up
`in`	*modelfactors*	structure with model factors
`in`	*z2*	2% wave run-up (m)
`in`	*s0*	wave steepness
`in`	*geometryflatberms*	structure with geometry data with horizontal berms
`out`	*succes*	flag for succes
`out`	*errormessage*	error message

Returns

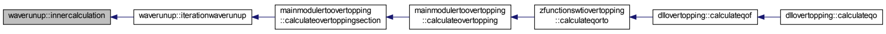
2% wave run-up at end of inner calculation

Definition at line 162 of file waveRunup.f90.

Here is the call graph for this function:


geometrymodulertoovertopping ::copygeometry
geometrymodulertoovertopping ::allocatevectorsgeometry
geometrymodulertoovertopping ::deallocategeometry
geometrymodulertoovertopping ::determinesegmenttypes
geometrymodulertoovertopping ::removeberms
geometrymodulertoovertopping ::calculatesegmentslopes
factormodulertoovertopping ::calculatetanalpha
overtoppingmessages ::getovertoppingformat
geometrymodulertoovertopping ::calculatehorzdistance
geometrymodulertoovertopping ::calculatehorzlengths
factormodulertoovertopping ::calculategammaf
factormodulertoovertopping ::calculategammab
overtoppingmessages ::getovertoppingmessage
waverunup::innercalculation
formulamodulertoovertopping ::calculatewaverunup
formulamodulertoovertopping ::adjustinfluencefactors
formulamodulertoovertopping ::calculatebreakerparameter
formulamodulertoovertopping ::calculatebreakerlimit
formulamodulertoovertopping ::realrootscubicfunction
formulamodulertoovertopping ::rootsgeneralcubic
formulamodulertoovertopping ::rootsdepressedcubic
formulamodulertoovertopping ::cubicroots
formulamodulertoovertopping ::isequalzero

Here is the caller graph for this function:


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

4.10.1.5 subroutine, public waverunup::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(inout) *gammaBeta_z,* type (tpovertoppinginput), intent(in) *modelFactors,* real(wp), intent(out) *z2,* logical, intent(out) *succes,* character(len=∗), intent(out) *errorMessage*)

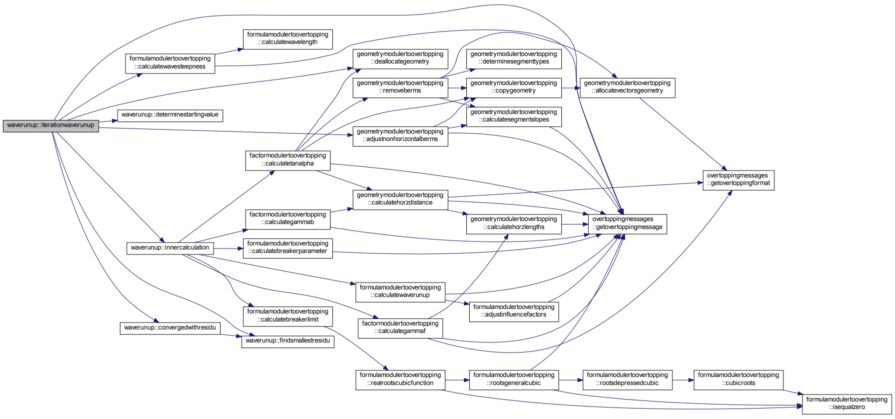
iterationWaveRunup: iteration for the wave runup

Parameters

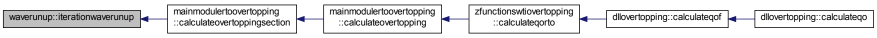
`in`	*geometry*	structure with geometry data
`in`	*h*	local water level (m+NAP)
`in`	*hm0*	significant wave height (m)
`in`	*tm_10*	spectral wave period (s)
`in,out`	*gammabeta_z*	influence factor angle wave attack 2% run-up
`in`	*modelfactors*	structure with model factors
`out`	*z2*	2% wave run-up (m)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

Definition at line 34 of file waveRunup.f90.

Here is the call graph for this function:


waverunup::iterationwaverunup
formulamodulertoovertopping ::calculatewavelength
formulamodulertoovertopping ::calculatewavesteepness
waverunup::determinestartingvalue
geometrymodulertoovertopping ::deallocategeometry
geometrymodulertoovertopping ::removeberms
geometrymodulertoovertopping ::adjustnonhorizontalberms
geometrymodulertoovertopping ::determinesegmenttypes
geometrymodulertoovertopping ::copygeometry
geometrymodulertoovertopping ::calculatesegmentslopes
geometrymodulertoovertopping ::allocatevectorsgeometry
factormodulertoovertopping ::calculatetanalpha
overtoppingmessages ::getovertoppingformat
geometrymodulertoovertopping ::calculatehorzdistance
factormodulertoovertopping ::calculategammab
geometrymodulertoovertopping ::calculatehorzlengths
overtoppingmessages ::getovertoppingmessage
waverunup::innercalculation
formulamodulertoovertopping ::calculatebreakerparameter
formulamodulertoovertopping ::calculatewaverunup
formulamodulertoovertopping ::adjustinfluencefactors
formulamodulertoovertopping ::calculatebreakerlimit
factormodulertoovertopping ::calculategammaf
waverunup::convergedwithresidu
waverunup::findsmallestresidu
formulamodulertoovertopping ::realrootscubicfunction
formulamodulertoovertopping ::rootsgeneralcubic
formulamodulertoovertopping ::rootsdepressedcubic
formulamodulertoovertopping ::cubicroots
formulamodulertoovertopping ::isequalzero

Here is the caller graph for this function:


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

4.11 zfunctionswtiovertopping Module Reference

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

Functions/Subroutines

- subroutine, public calculateqorto (dikeHeight, modelFactors, overtopping, load, geometry, succes, error←↩ Message)

  Subroutine to calculate the overtopping discharge with the RTO-overtopping dll.
- subroutine, public profileinstructure (nrCoordinates, xcoordinates, ycoordinates, dikeHeight, nrCoords←↩ Adjusted, 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, z←↩ CoordsAdjusted, 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)

4.11.1 Detailed Description

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

4.11.2 Function/Subroutine Documentation

4.11.2.1 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]

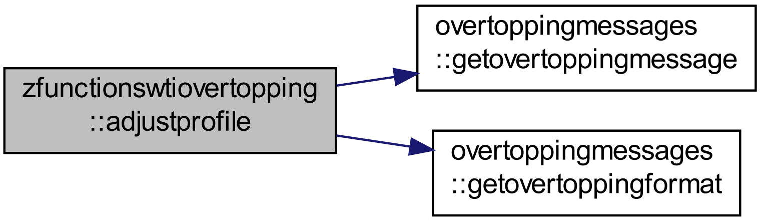
Subroutine adjust the profile due to a desired dike height.

Parameters

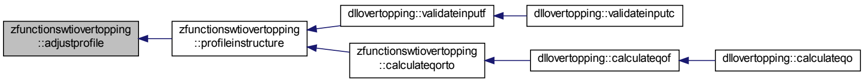
in	*nrcoordinates*	number of coordinates of the profile
in	*coordinates*	structure for the profile
in	*dikeheight*	dike height
out	*nrcoordsadjusted*	number of coordinates in the adjusted profile
	xcoordsadjusted	vector with x-coordinates of the adjusted profile
	zcoordsadjusted	vector with y-coordinates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

Definition at line 110 of file zFunctionsWTIOvertopping.f90.

Here is the call graph for this function:



Here is the caller graph for this function:



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

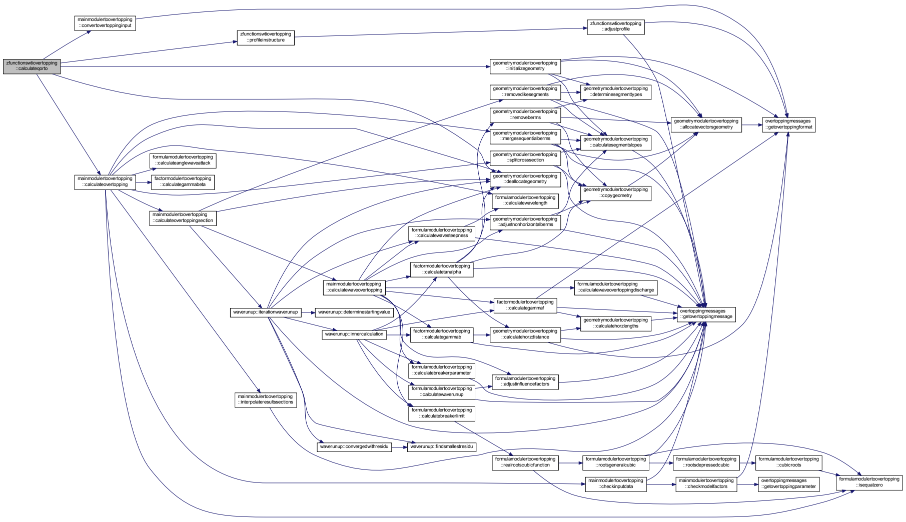
Subroutine to calculate the overtopping discharge with the RTO-overtopping dll.

Parameters

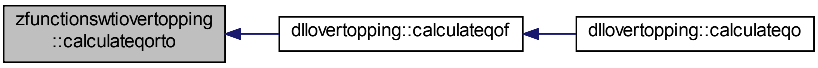
`in`	*dikeheight*	dike height
`in,out`	*modelfactors*	struct with model factors
`out`	*overtopping*	structure with overtopping results
`in`	*geometry*	structure with geometry data
`in`	*load*	structure with load parameters
`out`	*succes*	flag for succes
`out`	*errormessage*	error message

Definition at line 33 of file zFunctionsWTIOvertopping.f90.

Here is the call graph for this function:


zfunctionswtiovertopping
::calculateqorto
mainmodulertoovertopping
::convertovertoppinginput
zfunctionswtiovertopping
::profileinstructure
zfunctionswtiovertopping
::adjustprofile
geometrymodulertoovertopping
::initializegeometry
geometrymodulertoovertopping
::removedikesegments
geometrymodulertoovertopping
::determinesegmenttypes
geometrymodulertoovertopping
::removeberms
geometrymodulertoovertopping
::allocatevectorsgeometry
overtoppingmessages
::getovertoppingformat
geometrymodulertoovertopping
::mergesequentialberms
geometrymodulertoovertopping
::calculatesegmentslopes
geometrymodulertoovertopping
::splitcrosssection
formulamodulertoovertopping
::calculateanglewaveattack
mainmodulertoovertopping
::calculateovertopping
factormodulertoovertopping
::calculategammabeta
geometrymodulertoovertopping
::deallocategeometry
formulamodulertoovertopping
::calculatewavelength
geometrymodulertoovertopping
::copygeometry
mainmodulertoovertopping
::calculateovertoppingsection
geometrymodulertoovertopping
::adjustnonhorizontalberms
formulamodulertoovertopping
::calculatewavesteepness
factormodulertoovertopping
::calculatetanalpha
mainmodulertoovertopping
::calculatewaveovertopping
formulamodulertoovertopping
::calculatewaveovertoppingdischarge
factormodulertoovertopping
::calculategammaf
waverunup::iterationwaverunup
waverunup::determinestartingvalue
overtoppingmessages
::getovertoppingmessage
geometrymodulertoovertopping
::calculatehorzlengths
waverunup::innercalculation
factormodulertoovertopping
::calculategammab
geometrymodulertoovertopping
::calculatehorzdistance
formulamodulertoovertopping
::calculatebreakerparameter
formulamodulertoovertopping
::adjustinfluencefactors
mainmodulertoovertopping
::interpolateresultssections
formulamodulertoovertopping
::calculatewaverunup
formulamodulertoovertopping
::calculatebreakerlimit
waverunup::convergedwithresidu
waverunup::findsmallestresidu
formulamodulertoovertopping
::realrootscubicfunction
formulamodulertoovertopping
::rootsgeneralcubic
formulamodulertoovertopping
::rootsdepressedcubic
formulamodulertoovertopping
::cubicroots
mainmodulertoovertopping
::checkinputdata
mainmodulertoovertopping
::checkmodelfactors
overtoppingmessages
::getovertoppingparameter
formulamodulertoovertopping
::isequalzero

Here is the caller graph for this function:



4.11.2.3 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*)

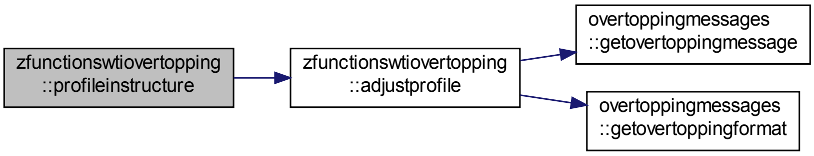
Subroutine to fill the profile in a structure and call the adjustment function of the profile due to a desired dike height.

Parameters

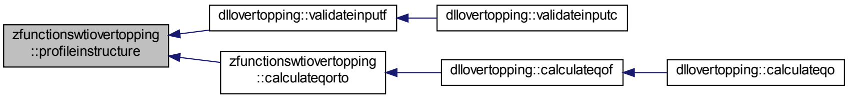
in	*nrcoordinates*	number of coordinates of the profile
in	*xcoordinates*	vector with x-coordinates of the profile
in	*ycoordinates*	vector with y-coordinates of the profile
in	*dikeheight*	dike height
out	*nrcoordsadjusted*	number of coordinates in the adjusted profile
	xcoordsadjusted	vector with x-coordinates 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 85 of file zFunctionsWTIOvertopping.f90.

Here is the call graph for this function:


zfunctionswtiovertopping ::profileinstructure
zfunctionswtiovertopping ::adjustprofile
overtoppingmessages ::getovertoppingmessage
overtoppingmessages ::getovertoppingformat

Here is the caller graph for this function:


zfunctionswtiovertopping ::profileinstructure
dllovertopping::validateinputf
dllovertopping::validateinputc
zfunctionswtiovertopping ::calculateqorto
dllovertopping::calculateqof
dllovertopping::calculateqo

4.11.2.4 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

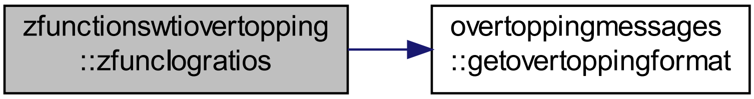
in	*qc*	Critical overtopping discharge
in	*mqo*	Model factor computed overtopping discharge
in	*mqc*	Model factor Critical overtopping discharge
out	*success*	Flag for succes
out	*errormessage*	error message, only set if not successful

Returns

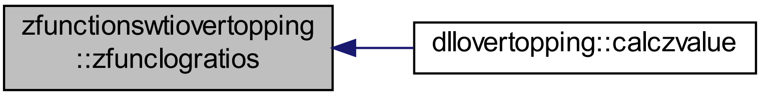
Value z-function

Definition at line 212 of file zFunctionsWTIOvertopping.f90.

Here is the call graph for this function:


zfunctionswtiovertopping ::zfunclogratios
overtoppingmessages ::getovertoppingformat

Here is the caller graph for this function:

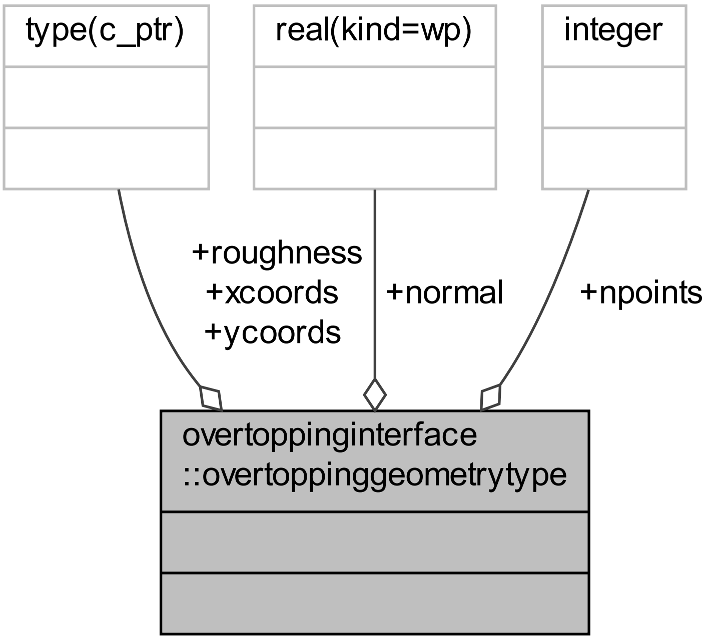

zfunctionswtiovertopping ::zfunclogratios
dllovertopping::calczvalue

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

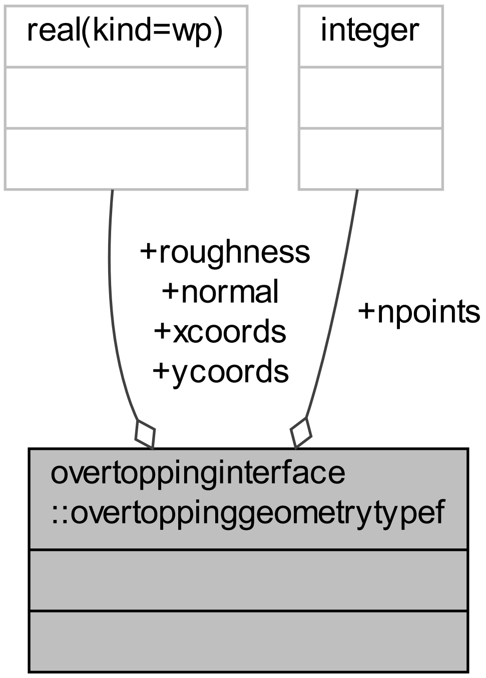
Definition at line 28 of file overtoppingInterface.f90.

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.

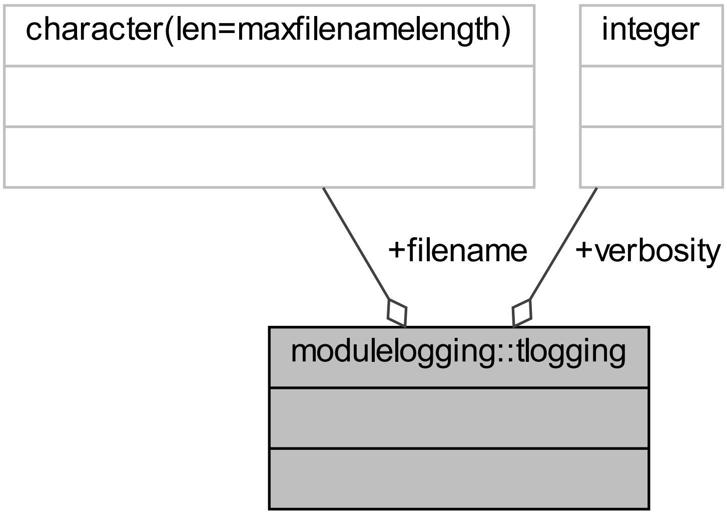
5.2.2.5 real(kind=wp), dimension(:), pointer overtoppinginterface::overtoppinggeometrytypef::ycoords

Definition at line 37 of file overtoppingInterface.f90.

5.3 modulelogging::tlogging Type Reference

TLogging: structure for steering the logging.

Collaboration diagram for modulelogging::tlogging:



Public Attributes

- integer verbosity = verboseNone

  level of verbosity: one of verboseNone, verboseBasic, verboseDetailed, verboseDebugging
- character(len=maxfilenamelength) filename = ' '

  filename of logging

5.3.1 Detailed Description

TLogging: structure for steering the logging.

Definition at line 16 of file ModuleLogging.f90.

5.3.2 Member Data Documentation

5.3.2.1 character(len=maxfilenamelength) modulelogging::tlogging::filename = ' '

filename of logging

Definition at line 18 of file ModuleLogging.f90.

5.3.2.2 integer modulelogging::tlogging::verbosity = verboseNone

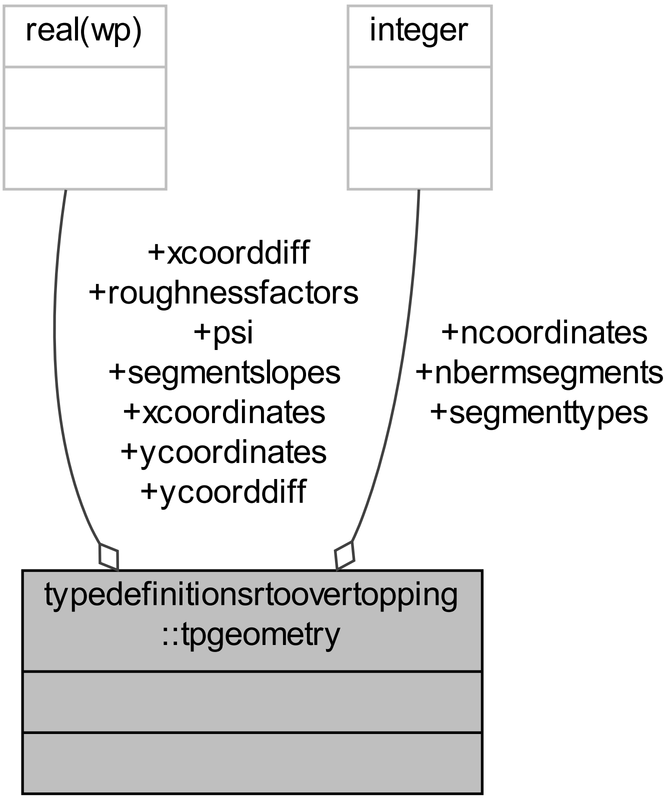
level of verbosity: one of verboseNone, verboseBasic, verboseDetailed, verboseDebugging

Definition at line 17 of file ModuleLogging.f90.

5.4 typedefinitionsrtoovertopping::tpgeometry Type Reference

tpGeometry: structure with geometry data

Collaboration diagram for typedefinitionsrtoovertopping::tpgeometry:



Public Attributes

- real(wp) psi

dike normal (degrees)

- 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 (degrees)

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

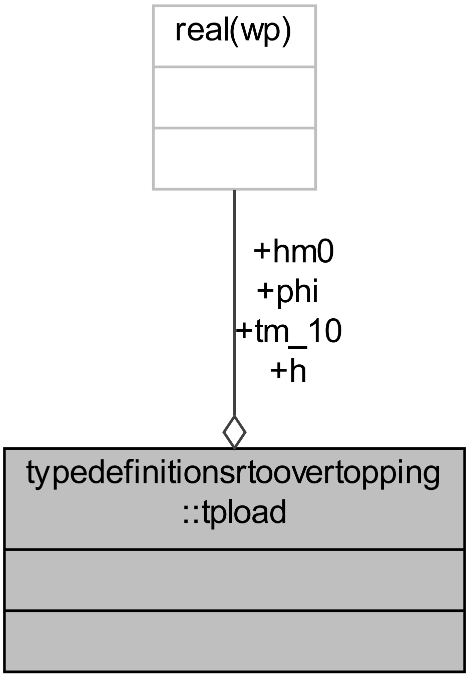
vector with y-coordinates cross section (m+NAP)

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 (degrees)

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 (degrees)

Definition at line 36 of file typeDefinitionsRTOovertopping.f90.

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

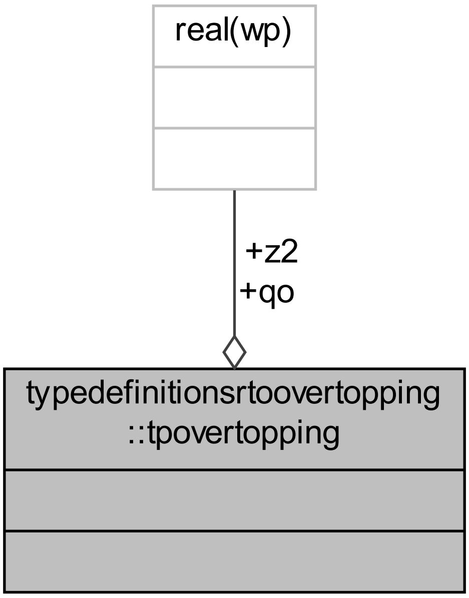
spectral wave period (s)

Definition at line 35 of file typeDefinitionsRTOovertopping.f90.

5.6 typedefinitionsrtoovertopping::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)

5.6.1 Detailed Description

tpOvertopping: structure with overtopping results

Definition at line 56 of file typeDefinitionsRTOovertopping.f90.

5.6.2 Member Data Documentation

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

wave overtopping discharge (m3/m per s)

Definition at line 58 of file typeDefinitionsRTOovertopping.f90.

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

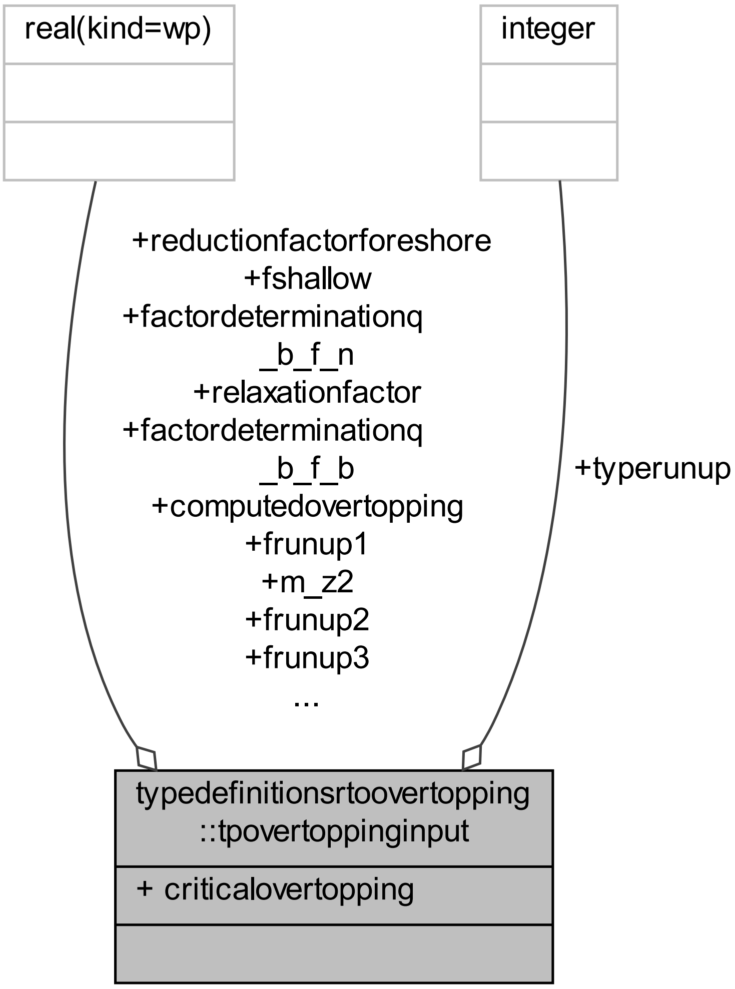
2% wave run-up (m)

Definition at line 57 of file typeDefinitionsRTOovertopping.f90.

5.7 typedefinitionsrtoovertopping::tpovertoppinginput Type Reference

OvertoppingModelFactors: C-structure with model factors.

Collaboration diagram for typedefinitionsrtoovertopping::tpovertoppinginput:



Public Attributes

- real(kind=wp) factordeterminationq_b_f_n

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

  model factor for breaking waves
- real(kind=wp) m_z2

  model factor describing the uncertainty of 2% runup height
- real(kind=wp) frunup1

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

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

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

model factor for shallow waves

- real(kind=wp) computedovertopping

  model factor computed overtopping
- real(kind=wp) criticalovertopping

  model factor critical overtopping
- integer typerunup

  0: fRunup1, 2 and 3 are given; 1: m_z2 is given
- real(kind=wp) relaxationfactor

  relaxation factor iteration procedure wave runup
- real(kind=wp) reductionfactorforeshore = 0.5_wp

  reduction factor foreshore

5.7.1 Detailed Description

OvertoppingModelFactors: C-structure with model factors.

Definition at line 40 of file typeDefinitionsRTOovertopping.f90.

5.7.2 Member Data Documentation

5.7.2.1 real(kind=wp) typedefinitionsrtoovertopping::tpovertoppinginput::computedovertopping

model factor computed overtopping

Definition at line 48 of file typeDefinitionsRTOovertopping.f90.

5.7.2.2 real(kind=wp) typedefinitionsrtoovertopping::tpovertoppinginput::criticalovertopping

model factor critical overtopping

Definition at line 49 of file typeDefinitionsRTOovertopping.f90.

5.7.2.3 real(kind=wp) typedefinitionsrtoovertopping::tpovertoppinginput::factordeterminationq_b_f_b

model factor for breaking waves

Definition at line 42 of file typeDefinitionsRTOovertopping.f90.

5.7.2.4 real(kind=wp) typedefinitionsrtoovertopping::tpovertoppinginput::factordeterminationq_b_f_n

model factor for non-breaking waves

Definition at line 41 of file typeDefinitionsRTOovertopping.f90.

5.7.2.5 real(kind=wp) typedefinitionsrtoovertopping::tpovertoppinginput::frunup1

model factor 1 for wave run-up (for backwards compatability)

Definition at line 44 of file typeDefinitionsRTOovertopping.f90.

5.7.2.6 real(kind=wp) typedefinitionsrtoovertopping::tpovertoppinginput::frunup2

model factor 2 for wave run-up (idem)

Definition at line 45 of file typeDefinitionsRTOovertopping.f90.

5.7.2.7 real(kind=wp) typedefinitionsrtoovertopping::tpovertoppinginput::frunup3

model factor 3 for wave run-up (idem)

Definition at line 46 of file typeDefinitionsRTOovertopping.f90.

5.7.2.8 real(kind=wp) typedefinitionsrtoovertopping::tpovertoppinginput::fshallow

model factor for shallow waves

Definition at line 47 of file typeDefinitionsRTOovertopping.f90.

5.7.2.9 real(kind=wp) typedefinitionsrtoovertopping::tpovertoppinginput::m_z2

model factor describing the uncertainty of 2% runup height

Definition at line 43 of file typeDefinitionsRTOovertopping.f90.

5.7.2.10 real(kind=wp) typedefinitionsrtoovertopping::tpovertoppinginput::reductionfactorforeshore = 0.5_wp

reduction factor foreshore

Definition at line 52 of file typeDefinitionsRTOovertopping.f90.

5.7.2.11 real(kind=wp) typedefinitionsrtoovertopping::tpovertoppinginput::relaxationfactor

relaxation factor iteration procedure wave runup

Definition at line 51 of file typeDefinitionsRTOovertopping.f90.

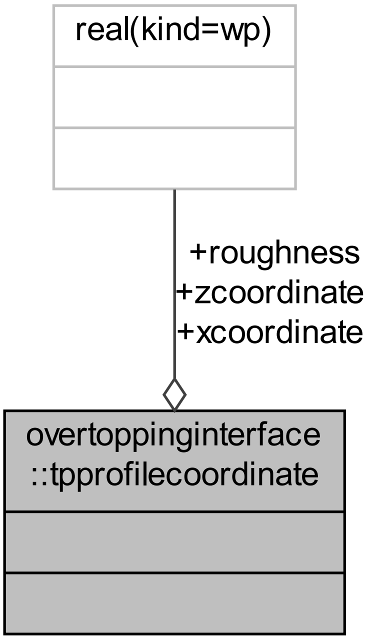
5.7.2.12 integer typedefinitionsrtoovertopping::tpovertoppinginput::typerunup

0: fRunup1, 2 and 3 are given; 1: m_z2 is given

Definition at line 50 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, verbosity, logFile)

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

  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::validateinputc (geometryInput, dikeHeight, modelFactors, success, error↩ Text)

  Subroutine that validates the geometry Wrapper for ValidateInputFold: convert C-like input structures to Fortran input structures.
- subroutine, public dllovertopping::validateinputf (geometryF, dikeHeight, modelFactors, errorStruct)

  Subroutine that validates the geometry.
- subroutine, public dllovertopping::setlanguage (lang)

  Subroutine that sets the language for error and validation messages.
- subroutine, public dllovertopping::getlanguage (lang)

  Subroutine that gets the language for error and validation messages.
- subroutine, public dllovertopping::versionnumber (version)

  Subroutine that delivers the version number.
- type(overtoppinggeometrytypef) function dllovertopping::geometry_c_f (geometryInput)

  Private subroutine that converts geometry from c-pointer to fortran struct.

6.1.1 Detailed Description

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

- calcZValue
- calculateQo
- calculateQoF
- ValidateInputC
- ValidateInputF
- SetLanguage
- GetLanguage
- 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, 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, 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 formulamodulertoovertopping::realrootscubicfunction (a, b, c, d, N, x, succes, errorMessage)

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

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

  rootsDepressedCubic: calculate the roots of a depressed cubic function
- subroutine 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, succes, errorMessage)

  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, succes, error←Message)

  copyGeometry: copy a geometry structure
- 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::basicgeometrytest (geometryF, success, errorStruct)

  basicGeometryTest: test the input geometry (the adjusted geometry is checked elsewhere)

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, 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::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, succes, errorMessage)

  checkInputdata: check the input data
- subroutine, public mainmodulertoovertopping::checkmodelfactors (modelFactors, dimErrMessage, error↩ Messages, ierr)

  checkModelFactors: check the input data
- subroutine, public mainmodulertoovertopping::convertovertoppinginput (modelFactors, success, error↩ Message)

  convertOvertoppingInput: convert the model factors from C-like to Fortran

6.5.1 Detailed Description

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

6.6 ModuleLogging.f90 File Reference

Module for steering the extra logging.

Data Types

- type modulelogging::tlogging

  TLogging: structure for steering the logging.

Modules

- module modulelogging

Variables

- integer, parameter modulelogging::maxfilenamelength = 256

  maximum length of filename
- type(tlogging) modulelogging::currentlogging

  copy of argument logging

6.6.1 Detailed Description

Module for steering the extra logging.

6.7 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.

6.7.1 Detailed Description

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

6.8 OvertoppingMessages.f90 File Reference

This file contains the messages in the overtopping dll, in Dutch or English.

Modules

- module overtoppingmessages

Functions/Subroutines

- subroutine overtoppingmessages::setlanguage (lang)

  IDs for the strings in this module:
- subroutine overtoppingmessages::getlanguage (lang)

  Subroutine that gets the language for error and validation messages.
- character(len=maxmsg) function overtoppingmessages::getovertoppingmessage (ID)

  Subroutine that returns a message with the corresponding ID in the current language.
- character(len=maxmsg) function overtoppingmessages::getovertoppingformat (ID)

  Subroutine that returns a Fortran format string with the corresponding ID in the current language.
- character(len=maxpar) function overtoppingmessages::getovertoppingparameter (ID)

  Subroutine that returns the name of an input parameter with the corresponding ID in the current language.

Variables

- integer, parameter, private overtoppingmessages::maxmsg = 128

  Module for the messages in the overtopping dll, in Dutch or English.
- integer, parameter, private overtoppingmessages::maxpar =32
- character(len=2), private overtoppingmessages::language = 'NL'

  default : Dutch

6.8.1 Detailed Description

This file contains the messages in the overtopping dll, in Dutch or English.

6.9 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::tpovertoppinginput

  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::mz2_min = 0.0d0

  minimal value model factor of 2% runup height
- real(wp), parameter typedefinitionsrtoovertopping::mz2_max = huge(mz2_max)

  maximal value model factor of 2% runup height
- real(wp), parameter typedefinitionsrtoovertopping::frunup1_min = 0.0d0

  minimal value model factor 1 for wave run-up
- real(wp), parameter typedefinitionsrtoovertopping::frunup1_max = huge(fRunup1_max)

  maximal value model factor 1 for wave run-up
- real(wp), parameter typedefinitionsrtoovertopping::frunup2_min = 0.0d0

  minimal value model factor 2 for wave run-up
- real(wp), parameter typedefinitionsrtoovertopping::frunup2_max = huge(fRunup2_max)

  maximal value model factor 2 for wave run-up
- real(wp), parameter typedefinitionsrtoovertopping::frunup3_min = 0.0d0

  minimal value model factor 3 for wave run-up
- real(wp), parameter typedefinitionsrtoovertopping::frunup3_max = huge(fRunup3_max)

  maximal value model factor 3 for wave run-up
- real(wp), parameter typedefinitionsrtoovertopping::fb_min = 0.0d0

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

  maximal value model factor for breaking waves
- real(wp), parameter typedefinitionsrtoovertopping::fn_min = 0.0d0

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

  maximal value model factor for non-breaking waves
- real(wp), parameter typedefinitionsrtoovertopping::fs_min = 0.0d0

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

  maximal value model factor for shallow waves
- real(wp), parameter typedefinitionsrtoovertopping::foreshore_min = 0.3d0

  minimal value reduction factor foreshore
- real(wp), parameter typedefinitionsrtoovertopping::foreshore_max = 1.0d0

maximal value reduction factor foreshore
- integer, parameter typedefinitionsrtoovertopping::z2_iter_max1 = 49

  maximal number of iterations for calculation z2 part 1
- integer, parameter typedefinitionsrtoovertopping::z2_iter_max2 = 70

  maximal number of iterations for calculation z2 part 1 & 2
- real(wp), parameter typedefinitionsrtoovertopping::z2_margin = 0.001d0

  margin for convergence criterium calculation z2

6.9.1 Detailed Description

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

6.10 waveRunup.f90 File Reference

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

Modules

- module waverunup

Functions/Subroutines

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

  iterationWaveRunup: iteration for the wave runup
- real(kind=wp) function waverunup::innercalculation (geometry, h, Hm0, gammaBeta_z, modelFactors, z2, s0, geometryFlatBerms, succes, errorMessage)
- real(kind=wp) function waverunup::determinestartingvalue (i, relaxationFactor, z2_start, z2_end, Hm0)
- integer function waverunup::findsmallestresidu (z2_start, z2_end, n)
- subroutine waverunup::convergedwithresidu (z2_start, z2_end)

6.10.1 Detailed Description

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

6.11 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, modelFactors, 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)

6.11.1 Detailed Description

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

Index