



3D/2D modelling suite for integral water solutions

DELFT3D

Deltires systems

Hydro-morphodynamics

User Manual

Deltires
Enabling Delta Life 

Delft3D-FLOW

**Simulation of multi-dimensional hydrodynamic flows
and transport phenomena, including sediments**

User Manual

Hydro-Morphodynamics

Version: 3.15
SVN Revision: 52614

October 19, 2017

Delft3D-FLOW, User Manual

Published and printed by:

Deltares
Boussinesqweg 1
2629 HV Delft
P.O. 177
2600 MH Delft
The Netherlands

telephone: +31 88 335 82 73
fax: +31 88 335 85 82
e-mail: info@deltares.nl
www: <https://www.deltares.nl>

For sales contact:

telephone: +31 88 335 81 88
fax: +31 88 335 81 11
e-mail: sales@deltaressystems.nl
www: <http://www.deltaressystems.nl>

For support contact:

telephone: +31 88 335 81 00
fax: +31 88 335 81 11
e-mail: support@deltaressystems.nl
www: <http://www.deltaressystems.nl>

Copyright © 2017 Deltares

All rights reserved. No part of this document may be reproduced in any form by print, photo print, photo copy, microfilm or any other means, without written permission from the publisher: Deltares.

Contents

List of Figures	xiii
List of Tables	xix
1 A guide to this manual	1
1.1 Introduction	1
1.2 Manual version and revisions	2
1.3 Typographical conventions	2
1.4 Changes with respect to previous versions	3
2 Introduction to Delft3D-FLOW	7
2.1 Areas of application	7
2.2 Standard features	7
2.3 Special features	8
2.4 Coupling to other modules	8
2.5 Utilities	9
2.6 Installation and computer configuration	9
3 Getting started	11
3.1 Overview of Delft3D	11
3.2 Starting Delft3D	11
3.3 Getting into Delft3D-FLOW	12
3.4 Exploring some menu options	15
3.5 Exiting the FLOW-GUI	16
4 Graphical User Interface	19
4.1 Introduction	19
4.2 MDF-file and attribute files	19
4.3 Filenames and conventions	20
4.4 Working with the FLOW-GUI	22
4.4.1 Starting the FLOW-GUI	22
4.4.2 Visualisation Area window	23
4.5 Input parameters of MDF-file	25
4.5.1 Description	26
4.5.2 Domain	26
4.5.2.1 Grid parameters	26
4.5.2.2 Bathymetry	32
4.5.2.3 Dry points	34
4.5.2.4 Thin dams	36
4.5.3 Time frame	38
4.5.4 Processes	41
4.5.5 Initial conditions	43
4.5.6 Boundaries	46
4.5.6.1 Flow boundary conditions	53
4.5.6.2 Transport boundary conditions	61
4.5.7 Physical parameters	64
4.5.7.1 Constants	64
4.5.7.2 Viscosity	69
4.5.7.3 Heat flux model	73
4.5.7.4 Sediment	76
4.5.7.5 Morphology	80
4.5.7.6 Wind	83
4.5.7.7 Tidal forces	86

4.5.8	Numerical parameters	86
4.5.9	Operations	90
4.5.9.1	Discharge	90
4.5.9.2	Dredging and dumping	95
4.5.10	Monitoring	95
4.5.10.1	Observations	95
4.5.10.2	Drogues	96
4.5.10.3	Cross-sections	98
4.5.11	Additional parameters	99
4.5.12	Output	100
4.5.12.1	Storage	102
4.5.12.2	Print	105
4.5.12.3	Details	106
4.6	Save the MDF and attribute files and exit	108
4.7	Importing, removing and exporting of data	109
5	Tutorial	111
5.1	Introduction – MDF-file and attribute files	111
5.2	Filenames and conventions	112
5.3	FLOW Graphical User Interface	114
5.3.1	Introduction	114
5.3.2	Saving the input data	115
5.4	Description	115
5.5	Domain	115
5.5.1	Grid parameters	116
5.5.2	Bathymetry	119
5.5.3	Dry points	119
5.5.4	Thin dams	122
5.6	Time frame	124
5.7	Processes	125
5.8	Initial conditions	126
5.9	Boundaries	127
5.10	Physical parameters	133
5.10.1	Constants	133
5.10.2	Roughness	134
5.10.3	Viscosity	135
5.10.4	Wind	135
5.11	Numerical parameters	137
5.12	Operations	138
5.13	Monitoring	140
5.13.1	Observation points	140
5.13.2	Drogues	142
5.13.3	Cross-sections	143
5.14	Additional parameters	144
5.15	Output	144
5.16	Save MDF-file	147
5.17	Additional exercises	148
5.18	Execute the scenario	148
5.19	Inspect the results	149
6	Execute a scenario	155
6.1	Running a scenario	155
6.1.1	Parallel calculations	155
6.1.1.1	DomainDecomposition	155

6.1.1.2	MPI-based parallel	155
6.1.1.3	Fluid mud	156
6.1.1.4	Mormerge	156
6.1.2	Running a scenario using Delft3D-MENU	156
6.1.3	Running a scenario using a batch script	158
6.2	Run time	158
6.3	Files and file sizes	160
6.3.1	History file	160
6.3.2	Map file	161
6.3.3	Print file	161
6.3.4	Communication file	162
6.4	Command-line arguments	162
6.5	Frequently asked questions	163
7	Visualise results	165
7.1	Introduction	165
7.2	Working with GPP	165
7.2.1	Overview	165
7.2.2	Launching GPP	167
7.3	Working with Delft3D-QUICKPLOT	169
7.4	GISVIEW interface	169
8	Manage projects and files	171
8.1	Introduction	171
8.1.1	Managing projects	172
8.1.2	Managing files	172
9	Conceptual description	173
9.1	Introduction	173
9.2	General background	173
9.2.1	Range of applications of Delft3D-FLOW	173
9.2.2	Physical processes	174
9.2.3	Assumptions underlying Delft3D-FLOW	175
9.3	Governing equations	176
9.3.1	Hydrodynamic equations	182
9.3.2	Transport equation (for sigma-grid)	190
9.3.3	Coupling between intake and outfall	193
9.3.4	Equation of state	193
9.4	Boundary conditions	195
9.4.1	Flow boundary conditions	196
9.4.1.1	Vertical boundary conditions	196
9.4.1.2	Open boundary conditions	200
9.4.1.3	Shear-stresses at closed boundaries	207
9.4.2	Transport boundary conditions	207
9.4.2.1	Open boundary conditions for the transport equation	207
9.4.2.2	Thatcher-Harleman boundary conditions	208
9.4.2.3	Vertical boundary conditions transport equation	210
9.5	Turbulence	210
9.5.1	Algebraic turbulence model (AEM)	214
9.5.1.1	Algebraic closure model (ALG)	214
9.5.1.2	Prandtl's Mixing Length model (PML)	215
9.5.2	k-L turbulence model	216
9.5.3	k-eps turbulence model	218
9.5.4	Low Reynolds effect	220

9.6	Secondary flow; sigma-model only	220
9.7	Wave-current interaction	225
9.7.1	Forcing by radiation stress gradients	227
9.7.2	Stokes drift and mass flux	228
9.7.3	Streaming	229
9.7.4	Wave induced turbulence	230
9.7.5	Enhancement of the bed shear-stress by waves	231
9.8	Heat flux models	236
9.8.1	Heat balance	237
9.8.2	Solar radiation	238
9.8.3	Atmospheric radiation (long wave radiation)	241
9.8.4	Back radiation (long wave radiation)	242
9.8.5	Effective back radiation	242
9.8.6	Evaporative heat flux	243
9.8.7	Convective heat flux	245
9.8.8	Overview of heat flux models	246
9.9	Tide generating forces	249
9.9.1	Tidal potential of Equilibrium tide	249
9.9.2	Tidal potential of Earth tide	252
9.10	Hydraulic structures	252
9.10.1	3D gate	253
9.10.2	Quadratic friction	253
9.10.3	Linear friction	254
9.11	Flow resistance: bedforms and vegetation	254
9.11.1	Bedform heights	254
9.11.1.1	Dune height predictor	254
9.11.1.2	Van Rijn (2007) bedform roughness height predictor	256
9.11.2	Trachytopes	258
9.11.2.1	Trachytype classes	258
9.11.2.2	Averaging and accumulation of trachytopes	265
9.11.3	(Rigid) 3D Vegetation model	267
10	Numerical aspects of Delft3D-FLOW	269
10.1	Staggered grid	269
10.2	sigma-grid and Z-grid	271
10.3	Definition of model boundaries	271
10.4	Time integration of the 3D shallow water equations	272
10.4.1	ADI time integration method	274
10.4.2	Accuracy of wave propagation	276
10.4.3	Iterative procedure continuity equation	277
10.4.4	Horizontal viscosity terms	277
10.4.5	Overview time step limitations	278
10.5	Spatial discretizations of 3D shallow water equations	278
10.5.1	Horizontal advection terms	278
10.5.2	Vertical advection term	283
10.5.3	Viscosity terms	283
10.6	Solution method for the transport equation	283
10.6.1	Cyclic method	285
10.6.2	Van Leer-2 scheme	286
10.6.3	Vertical advection	286
10.6.4	Forester filter	287
10.7	Numerical implementation of the turbulence models	289
10.8	Drying and flooding	289
10.8.1	Bottom depth at water level points	290

10.8.2	Total water depth at velocity points	292
10.8.3	Drying and flooding criteria	295
10.9	Hydraulic structures	297
10.9.1	3D Gate	298
10.9.2	Quadratic friction	299
10.9.2.1	Barrier	300
10.9.2.2	Bridge	301
10.9.2.3	Current Deflection Wall	301
10.9.2.4	Weir	301
10.9.2.5	Porous plate	304
10.9.2.6	Culvert	304
10.9.3	Linear friction	307
10.9.4	Floating structure	307
10.10	Artificial vertical mixing due to sigma co-ordinates	307
10.11	Smoothing parameter boundary conditions	311
10.12	Assumptions and restrictions	312
11	Sediment transport and morphology	315
11.1	General formulations	315
11.1.1	Introduction	315
11.1.2	Suspended transport	315
11.1.3	Effect of sediment on fluid density	316
11.1.4	Sediment settling velocity	317
11.1.5	Dispersive transport	317
11.1.6	Three-dimensional wave effects	318
11.1.7	Initial and boundary conditions	318
11.1.7.1	Initial condition	319
11.1.7.2	Boundary conditions	319
11.2	Cohesive sediment	320
11.2.1	Cohesive sediment settling velocity	320
11.2.2	Cohesive sediment dispersion	321
11.2.3	Cohesive sediment erosion and deposition	321
11.2.4	Interaction of sediment fractions	322
11.2.5	Influence of waves on cohesive sediment transport	322
11.2.6	Inclusion of a fixed layer	322
11.2.7	Inflow boundary conditions cohesive sediment	323
11.3	Non-cohesive sediment	323
11.3.1	Non-cohesive sediment settling velocity	323
11.3.2	Non-cohesive sediment dispersion	324
11.3.2.1	Using the algebraic or $k-L$ turbulence model	324
11.3.2.2	Using the $k-\varepsilon$ turbulence model	326
11.3.3	Reference concentration	326
11.3.4	Non-cohesive sediment erosion and deposition	328
11.3.5	Inclusion of a fixed layer	331
11.3.6	Inflow boundary conditions non-cohesive sediment	331
11.4	Bedload sediment transport of non-cohesive sediment	331
11.4.1	Basic formulation	331
11.4.2	Suspended sediment correction vector	332
11.4.3	Interaction of sediment fractions	333
11.4.4	Inclusion of a fixed layer	333
11.4.5	Calculation of bedload transport at open boundaries	334
11.4.6	Bedload transport at U and V velocity points	334
11.4.7	Adjustment of bedload transport for bed-slope effects	335
11.5	Transport formulations for non-cohesive sediment	338

11.5.1	Van Rijn (1993)	338
11.5.2	Engelund-Hansen (1967)	343
11.5.3	Meyer-Peter-Muller (1948)	343
11.5.4	General formula	344
11.5.5	Bijker (1971)	344
11.5.5.1	Basic formulation	345
11.5.5.2	Transport in wave propagation direction (Bailard-approach)	346
11.5.6	Van Rijn (1984)	348
11.5.7	Soulsby/Van Rijn	350
11.5.8	Soulsby	351
11.5.9	Ashida–Michiue (1974)	354
11.5.10	Wilcock–Crowe (2003)	355
11.5.11	Gaeuman et al. (2009) laboratory calibration	355
11.5.12	Gaeuman et al. (2009) Trinity River calibration	356
11.6	Morphological updating	356
11.6.1	Bathymetry updating including bedload transport	359
11.6.2	Erosion of (temporarily) dry points	359
11.6.3	Dredging and dumping	360
11.6.4	Bed composition models and sediment availability	361
11.7	Specific implementation aspects	362
11.8	Validation	363
12	Fixed layers in Z-model	365
12.1	Background	365
12.2	Time integration of the 3D shallow water equations	368
12.2.1	ADI time integration method	368
12.2.2	Linearisation of the continuity equation	370
12.3	Bed stress term	370
12.4	Horizontal viscosity terms	370
12.4.1	Overview time step limitations	371
12.5	Spatial discretisations of 3D shallow water equations	371
12.5.1	Horizontal advection terms	371
12.5.2	Vertical advection term	373
12.5.3	Viscosity terms	373
12.6	Solution method for the transport equation	373
12.6.1	Horizontal advection	376
12.6.1.1	Van Leer-2 scheme	376
12.6.1.2	Implicit upwind scheme	376
12.6.2	Vertical advection	377
12.6.3	Forester filter	377
12.7	Baroclinic pressure term	378
12.8	Numerical implementation of the turbulence models	379
12.9	Drying and flooding	379
12.9.1	Bottom depth at water level points	379
12.9.2	Bottom depth at velocity points	380
12.9.3	Upwinding of the water level in defining the total water depth	381
12.9.4	Drying and flooding criteria	381
12.10	Cut-cell and 45 degrees closed boundaries	383
12.10.1	Introduction	383
12.10.2	Cut Cells	383
12.10.3	45 degrees closed boundary	384
12.11	Hydraulic structures	385
12.11.1	3D Gate	385
12.11.2	Quadratic friction	386

12.11.3 Linear friction	386
12.11.4 Floating structure	386
12.12 Assumptions and restrictions	386
References	389
Glossary of terms	399
A Files of Delft3D-FLOW	403
A.1 MDF-file	403
A.1.1 Introduction	403
A.1.2 Example	403
A.1.3 Physical parameters	406
A.1.3.1 Tide Generating Forces	407
A.1.3.2 Thatcher-Harleman Conditions	407
A.1.4 Output options	407
A.1.4.1 Momentum terms output	413
A.2 Attribute files	414
A.2.1 Introduction	414
A.2.2 Orthogonal curvilinear grid	415
A.2.3 Computational grid enclosure	416
A.2.4 Bathymetry	417
A.2.5 Thin dams	418
A.2.6 Dry points	419
A.2.7 Time-series uniform wind	420
A.2.8 Space varying wind and pressure	421
A.2.8.1 Defined on the computational grid	422
A.2.8.2 Defined on an equidistant grid	425
A.2.8.3 Defined on a curvilinear grid	425
A.2.8.4 Defined on a Spiderweb grid	426
A.2.9 Initial conditions	426
A.2.10 Open boundaries	428
A.2.11 Astronomic flow boundary conditions	429
A.2.12 Astronomic correction factors	430
A.2.13 Harmonic flow boundary conditions	432
A.2.14 QH-relation flow boundary conditions	433
A.2.15 Time-series flow boundary conditions	434
A.2.16 Time-series correction of flow boundary conditions	438
A.2.17 Time-series transport boundary conditions	440
A.2.18 Time-series for the heat model parameters	445
A.2.19 Bottom roughness coefficients	448
A.2.20 Horizontal eddy viscosity and diffusivity	449
A.2.21 Discharge locations	451
A.2.22 Flow rate and concentrations at discharges	452
A.2.23 Dredge and dump characteristics	454
A.2.24 Dredge and nourishment time-series	462
A.2.25 Polygon file	463
A.2.26 Observation points	464
A.2.27 Moving observation points	465
A.2.28 Drogues	467
A.2.29 Cross-sections	468
A.2.30 Fourier analysis	469
A.2.31 (Rigid) 3D vegetation model	470
A.2.32 Space varying subsidence uplift definition	474

B Special features of Delft3D-FLOW	477
B.1 Introduction	477
B.2 Decay rate constituents	479
B.3 Hydraulic structures	479
B.3.1 3D gates	479
B.3.2 Quadratic friction	480
B.3.2.1 Barrier	480
B.3.2.2 Bridge	482
B.3.2.3 Current deflection wall (CDW)	482
B.3.2.4 Weirs (2D model)	483
B.3.2.5 Local weir	485
B.3.3 Porous plate	487
B.3.4 Culvert	489
B.3.4.1 Definition of culvert in the discharge input file (<*.src>)	489
B.3.4.2 Properties for culverts defined in INI file (<name.cul>)	490
B.3.4.3 Additional key-value pairs for culvert of type 'c'	491
B.3.4.4 Additional key-value pairs for culvert of type 'd' or 'e'	491
B.3.4.5 Additional key-value pairs for culvert of type 'f'	492
B.3.4.6 Additional key-value pairs for culvert of type 'u'	492
B.3.4.7 More culverts	493
B.3.5 Linear friction	493
B.3.5.1 Rigid sheet	494
B.3.6 Floating structure	495
B.3.7 Upwind at Discharges	497
B.3.8 User defined discharge through a structure	497
B.4 Space varying Coriolis coefficients	507
B.5 Temperature modelling	508
B.5.1 Direct specification of net solar radiation	508
B.5.2 Specification of the coefficient of free convection	509
B.5.3 Output of computed heat fluxes	509
B.6 Evaporation and precipitation	510
B.7 Space varying wind and pressure	511
B.7.1 Space varying wind and pressure on an equidistant grid	512
B.7.2 Space varying wind and pressure on a separate curvilinear grid	517
B.7.3 Space varying wind and pressure on a Spiderweb grid	520
B.8 Horizontal large eddy simulation	525
B.9 Sediment transport and morphology	529
B.9.1 Sediment input file	530
B.9.2 Morphology input file	536
B.9.3 Sediment transport input file	547
B.9.4 User defined transport routine for sand or bedload fractions	551
B.9.5 User defined transport routine for mud fractions	555
B.9.6 User defined routine for the settling velocity	558
B.9.7 Sediment transport and morphology boundary condition file	561
B.9.8 Morphological factor file	564
B.9.9 Initial bed composition file	566
B.10 Fluid mud (2-layer approach)	567
B.10.1 Two layer system	569
B.10.1.1 Suspension layer	569
B.10.1.2 Fluid mud layer	570
B.10.1.3 Mathematical modelling of fluid mud layer	570
B.10.2 Applying fluid mud	572
B.10.2.1 DelftIO library	573
B.10.2.2 Running a simulation in foreground	574

B.10.2.3	Running a simulation in background	574
B.10.2.4	Pitt falls	574
B.10.3	Post-processing	574
B.11	Z-model	575
B.11.1	Grid definition and construction	575
B.11.2	Defining the keywords in the FLOW-GUI	576
B.11.3	Restrictions and limitations	576
B.11.3.1	Defining Cut-cells and 45 degrees closed boundaries	578
B.11.4	45 degrees staircase closed boundary points (Z -model only)	579
B.11.5	Cut-cell closed boundary points (Z -model only)	580
B.12	Non-hydrostatic solver	581
B.12.1	The use of hydrostatic and non-hydrostatic models	581
B.12.2	Governing equations	581
B.12.3	A pressure correction technique for computing the non-hydrostatic pressure	582
B.12.4	Boundary conditions	585
B.12.5	Conjugate gradient method (CG)	586
B.12.6	Practical aspects of using the non-hydrostatic flow module	587
B.12.6.1	Switches in MDF-file	587
B.12.6.2	Grid spacing	589
B.12.6.3	Vertical mixing	589
B.12.6.4	Convergence criterion CG solver	589
B.12.6.5	Defining the input (keywords) for the non-hydrostatic pressure approach	590
B.13	User defined functions	591
B.13.1	Boundary conditions for turbulence models	591
B.13.2	Diagnostic mode	593
B.13.3	Particle wind factor	593
B.14	Domain decomposition	594
B.14.1	Introduction	594
B.14.2	Motivations for domain decomposition	594
B.14.3	Local refinement horizontal and vertical	598
B.14.4	Pre-processing, processing and post-processing	599
B.14.5	Restrictions	601
B.14.6	How to set-up a domain decomposition model	602
B.15	Surfbeat/roller model	603
B.15.1	Domain	603
B.15.2	Formulations	604
B.15.3	Boundary conditions	606
B.15.4	Coupling with other modules	607
B.15.5	Modes of operation	607
B.15.6	Input description	607
B.16	Bedform heights	612
B.17	Trachytopes	615
B.17.1	Trachytype definition file	616
B.17.2	Area files	617
B.18	Creating D-Water Quality input files	618
B.19	Dry run	619
B.20	Reuse temporary files	620
B.21	Change the update frequency of the nodal factors	620
B.22	Bubble screen	620
B.22.1	Entrained water as function of the air injection	621
B.22.1.1	Single nozzle bubble plume	622
B.22.1.2	Bubble screen or line bubble plume	622

B.22.1.3	Bubble plume in stagnant stratified water	623
B.22.2	Theory	624
B.22.3	Numerical implementation	627
B.22.4	Input description	628
B.22.4.1	Generating a file with bubble screen locations	629
B.22.4.2	Extending the discharge locations file with bubble screens .	630
B.22.4.3	Extending the time-series file (<*.dis>) with amount of en- trained water	630
B.22.5	Coupling with other models	630
B.22.6	Model results	631
B.23	1D–3D Coupling	631
B.23.1	Motivation for online Delft3D-SOBEK coupling	632
B.23.2	Implementation of Delft3D-SOBEK coupling	633
B.23.3	Model setup and input (including best practise)	634
B.23.3.1	Preparation of the Delft3D-FLOW and SOBEK models . . .	634
B.23.3.2	Setup of the communication file used by coupling	635
B.23.3.3	Running of the coupled model system	638
B.23.3.4	Best practice with regard to running coupled Delft3D-SOBEK simulations	638
B.23.4	Versions and limitations	639
B.24	Output of Courant number messages	639
B.25	Initialisation of water depth in dry points	639
B.26	Remapping of near-bottom layers for accurate and smooth bottom shear stress in Z-layers	640
B.27	Slope Limiter	641
B.28	Real-time control	641
B.28.1	Run procedure	642
B.28.2	Time-series forcing of controlled structures	642
B.28.3	Data Locations layer	645
B.28.4	Decision layer	645
B.28.5	Measures layer	658
C	Astronomical constituents	661
C.1	Introduction	661
C.2	List of astronomical constituents	661
D	Some modelling guidelines	667
D.1	Introduction	667
D.2	Depth-averaged or 3D model	667
D.3	Selection of the vertical turbulence model	668
D.3.1	Well-mixed	669
D.3.2	Partly mixed	669
D.3.3	Strongly stratified	669
E	Computational grid	671
F	Delft3D-NESTHD	677
F.1	Introduction	677
F.2	How to use NESTHD1	677
F.3	How to use NESTHD2	679
F.4	Example	681

List of Figures

2.1	System architecture of Delft3D	8
3.1	Title window of Delft3D	11
3.2	Main window Delft3D-MENU	12
3.3	Selection window for Hydrodynamics	13
3.4	Select working directory window	13
3.5	Select working directory window to set the working directory to <flow\friesian_tidal_inlet>	13
3.6	The current working directory is not shown in the title bar due to its length	14
3.7	Main window of the FLOW Graphical User Interface	14
3.8	Menu bar of the FLOW-GUI	15
3.9	<i>File</i> drop down menu	15
3.10	Data Group <i>Domain</i> selection and input fields	16
3.11	Save changes window	16
4.1	Main window of the FLOW-GUI	22
4.2	Visualisation Area Window	23
4.3	Possible selections of <i>View</i> → <i>Attributes</i>	24
4.4	Display symbols of all grid related quantities	25
4.5	Data Group <i>Description</i>	26
4.6	Staggered grid of Delft3D-FLOW	27
4.7	Sub-data group <i>Grid parameters</i>	29
4.8	Definition sketch grid system to North orientation	30
4.9	Specifying the layers thickness	30
4.10	Sub-data group <i>Domain</i> → <i>Bathymetry</i>	32
4.11	Dry point at grid location (m, n)	34
4.12	Sub-data group <i>Dry points</i>	35
4.13	Equivalence of <i>v</i> -type thin dams (left) and <i>u</i> -type thin dams (right) with the same grid indices, (M–1 to M+1, N)	36
4.14	Sub-data group <i>Thin dams</i>	37
4.15	Data Group <i>Time frame</i>	38
4.16	Data Group <i>Processes</i>	40
4.17	Sediment definition window	42
4.18	Data group <i>Initial conditions</i>	44
4.19	Sketch of cross-section with 8 grid cells	48
4.20	Main window for defining open boundaries	49
4.21	Open and save window for boundary locations and conditions	50
4.22	Straight channel; location of open and closed boundaries	51
4.23	Specifying astronomical boundary conditions	54
4.24	Contents of a Component set with two tidal constituents having corrections	56
4.25	Specifying harmonic boundary conditions	57
4.26	Specifying QH-relation boundary conditions	59
4.27	Specifying time-series boundary conditions	60
4.28	Transport conditions; Thatcher Harleman time lags	61
4.29	Specifying transport boundary conditions	62
4.30	Specifying the physical constants	64
4.31	Examples of the wind drag coefficient	66
4.32	Sub-data group <i>Roughness</i>	67
4.33	Defining the eddy viscosity and eddy diffusivity	69
4.34	Window with HLES parameters	70
4.35	Sub-data group Heat flux model	73
4.36	Sub-data group <i>Sediment</i> , overall and cohesive sediment parameters	76
4.37	Sub-data group <i>Sediment</i> , cohesive sediment parameters (continued)	77

4.38	Sub-data group <i>Sediment</i> , cohesive sediment parameters (continued)	78
4.39	Sub-data group <i>Sediment</i> , non-cohesive sediment parameters	79
4.40	Sub-data group <i>Morphology</i>	81
4.41	Wind definition window	84
4.42	Nautical definition of the wind direction	85
4.43	Sub-window Tidal forces	86
4.44	Data Group Numerical parameters	87
4.45	Data Group Discharges	91
4.46	Sub-window to define the discharge rate and substance concentrations	93
4.47	Decomposition of momentum released by a discharge station in (m, n)	93
4.48	Sub-data group <i>Dredging and dumping</i>	95
4.49	Sub-window for <i>Monitoring</i> locations	95
4.50	Sub-window for <i>Observation</i> points	96
4.51	Sub-data group <i>Monitoring</i> → <i>Drogues</i>	97
4.52	Sub-data group <i>Monitoring</i> → <i>Cross-Sections</i>	98
4.53	Data Group Additional parameters	100
4.54	Sub-data group <i>Output storage</i>	101
4.55	Sub-data group <i>Output</i> → <i>Storage</i> → <i>Edit WAQ input</i>	104
4.56	Sub-data group <i>Output</i> → <i>Print</i>	106
4.57	Output Specifications window	107
4.58	<i>File</i> drop down menu	107
4.59	Save changes window	109
5.1	Starting window of the FLOW Graphical User Interface	114
5.2	Data Group Description sub-window	115
5.3	Sub-data group Grid	116
5.4	Sub-data group Grid ; filenames, type of co-ordinate system and grid dimensions	116
5.5	Visualisation Area window	118
5.6	Staggered grid used in Delft3D-FLOW	118
5.7	Sub-data group Bathymetry	119
5.8	Location of a dry point at grid indices (m, n)	120
5.9	Sub-data group Dry Points	121
5.10	Sets of thin dams blocking <i>v</i> -velocities (left) and blocking <i>u</i> -velocities (right)	122
5.11	Sub-data group Thin dams	122
5.12	Equivalence of <i>v</i> -type thin dams (left) and <i>u</i> -type thin dams (right) with the same grid indices, $(m - 1$ to $m + 1, n)$	123
5.13	Window Time frame	124
5.14	Processes window	125
5.15	Processes: Pollutants and tracers sub-window	126
5.16	Initial conditions sub-window	127
5.17	Boundaries sub-window	128
5.18	Open/Save Boundaries sub-window	129
5.19	Harmonic boundary conditions	131
5.20	Boundaries: Transport Conditions window	131
5.21	Physical parameters sub-data groups	133
5.22	Physical parameters - Constants sub-window	134
5.23	Roughness sub-window	134
5.24	Viscosity sub-window	135
5.25	Wind sub-window	136
5.26	Filled time table in the Wind sub-window	136
5.27	Numerical parameters sub-window	137
5.28	Data Group Operations; Discharges sub-window	138
5.29	Representation of a discharge in the Visualisation Area window	139
5.30	Discharge Data sub-window	139

5.31	Observation points sub-window	141
5.32	Representation of an observation point in the Visualisation area window	141
5.33	Drogues sub-window	142
5.34	Representation of a drogue in the Visualisation Area window	143
5.35	Cross-sections sub-window	143
5.36	Representation of a v-cross-section in the Visualisation Area window	144
5.37	Output sub-window	145
5.38	Output details sub-window	146
5.39	Select scenario to be executed	148
5.40	Part of the <tri-diag> report file	149
5.41	Computed time-series of the water level, current and salinity in observation point Obs4	151
5.42	Computational grid with drogue Dr4, and contours of water level on 6 August 1990 01:00 hr	152
5.43	Vector velocities and contours of salinity on 6 August 1990 01:00 hr	153
6.1	MENU-window for Hydrodynamics	156
6.2	Select the MDF-file to be verified	157
6.3	Part of the report to the output window	157
6.4	Select a report file for inspection	158
7.1	Hierarchy of GPP	166
7.2	Main window of GPP	167
7.3	Parameters and locations in the <trih-tut_fti.dat> file	168
7.4	Some options to change the plot attributes	169
7.5	Delft3D-QUICKPLOT interface to Delft3D result	170
9.1	Definition of water level (ζ), depth (h) and total depth (H)	183
9.2	Example of σ - and Z -grid	184
9.3	Examples wind drag coefficient depending on wind speed	200
9.4	Hydrodynamic model of coastal area with three open boundaries with offshore boundary (A–B at deep water) and two cross shore boundaries (A–A', and B–B')	205
9.5	Illustration of memory effect for open boundary	209
9.6	Spiral motion in river bend (from Van Rijn (1990))	221
9.7	Vertical profile secondary flow (V) in river bend and direction bed stress	222
9.8	Vertical distribution of turbulent kinetic energy production	230
9.9	Schematic view of non-linear interaction of wave and current bed shear-stresses (from Soulsby <i>et al.</i> (1993b, Figure 16, p. 89))	232
9.10	Inter-comparison of eight models for prediction of mean and maximum bed shear-stress due to waves and currents (from Soulsby <i>et al.</i> (1993b, Figure 17, p. 90))	234
9.11	Overview of the heat exchange mechanisms at the surface	236
9.12	Co-ordinate system position Sun δ : declination; θ : latitude; ωt : angular speed	240
9.13	Effect of tide generating force on the computed water elevation at Venice	250
9.14	Earth ocean tidal interaction (after Schwiderski (1980))	251
10.1	Example of a grid in Delft3D-FLOW	270
10.2	Mapping of physical space to computational space	270
10.3	Grid staggering, 3D view (left) and top view (right)	270
10.4	Example of Delft3D-FLOW model area	272
10.5	Example of Delft3D-FLOW grid	273
10.6	Numerical region of influence for one time step, "Zig-zag channel"	276

10.7 (a) Control Volume for mass for the Flooding scheme,	280
(b) Control Volume for momentum in horizontal and	
(c) vertical direction for the Flooding scheme	
10.8 Layer numbering in σ -model	283
10.9 Illustration of wiggles in vertical direction	288
10.10 Definition bottom depth on FLOW grid	291
10.11 Negative control volume with two positive flow-through heights, MEAN-option	292
10.12 Drying of a tidal flat; averaging approach. The flow-through height is based on the average water level, see Equation 10.63, the velocity point is set dry.	293
10.13 Overtopping of a river bank (weir); averaging approach. The flow-through height is based on the average water level, see Equation 10.63, the velocity point is set dry.	293
10.14 Drying of a tidal flat; upwind approach. The flow-through height is determined by flow direction, see Equation 10.64, the velocity point remains wet.	294
10.15 Overtopping of a river bank; upwind approach. The flow-through height is based on the maximum water level, see Equation 10.64, the velocity point remains wet.	294
10.16 Special limiter for critical flow due to a sudden contraction (Flooding scheme and increase in bottom larger than $DGCUNI$)	295
10.17 Example of a 3D Gate (vertical cross-section)	298
10.18 Computational layer partially blocked at bottom of gate	299
10.19 Example of a hydrostatic consistent and inconsistent grid; (a) $H\delta\sigma > \sigma\frac{\partial H}{\partial x}\delta x$, (b) $H\delta\sigma < \sigma\frac{\partial H}{\partial x}\delta x$	308
10.20 Finite Volume for diffusive fluxes and pressure gradients	309
10.21 Left and right approximation of a strict horizontal gradient	310
10.22 Cold start with damping of eigen oscillations due to bottom friction	312
11.1 Sediment mixing coefficient in non-breaking waves (Source: Van Rijn (1993))	325
11.2 Selection of the k_{mx} layer; where a is Van Rijn's reference height	327
11.3 Schematic arrangement of flux bottom boundary condition	328
11.4 Approximation of concentration and concentration gradient at bottom of k_{mx} layer	329
11.5 Setting of bedload transport components at velocity points	335
11.6 Morphological control volume and bedload transport components	359
12.1 Irregular representation of bottom boundary layer in the Z -model	366
12.2 Vertical computational grid Z -model (left) and σ -model (right)	367
12.3 discretisation along streamlines. Grid points in difference stencil dependent on flow direction	372
12.4 Aggregation of Control volumes in the vertical due to variation position free surface	375
12.5 Horizontal fluxes between neighbouring cells with variation in position free sur- face	375
12.6 Definition bottom depth on Delft3D-FLOW grid	380
12.7 The flow-through height is determined by the flow direction. The bottom is represented as a staircase around the depth in water level points.	381
12.8 left: Cut Cell (definition) and right: defined by shifting (exaggerated) the corner point to boundary.	384
12.9 Flow along staircase boundary.	384
12.10 Reflection of velocities	384
12.11 Example of a 3D Gate (vertical cross-section)	385
12.12 Computational layer partially blocked at the bottom of the 3D gate	385
A.1 Example of computational grid enclosures	417

A.2	Example of thin dams in a model area	419
A.3	Dry points in model area	420
A.4	Definition sketch of wind direction according to Nautical convention	421
A.5	Definition wind components for space varying wind	424
A.6	Cross-sections in model area	469
A.7	Example of the plant input file (<i><name.pla></i>) where the areas are defined with a polygon file, see subsection A.2.25	473
A.8	Example of the plant input file (<i><name.pla></i>) where the area is defined with files according the depth-format, see subsection A.2.4	473
A.9	Example of the plant input file (<i><name.pla></i>) where two different vegetation types are defined.	474
B.1	Example of 3D gates in perspective view (left) and top view (right)	481
B.2	Barriers in model area	482
B.3	Example of CDW in perspective view (left) and top view (right)	484
B.4	Top view of 2D weirs	486
B.5	Local weir	487
B.6	Top view of rigid sheets	495
B.7	Cross-sectional view floating structure The vertical lines are drawn through the velocity points	497
B.8	Illustration of the data to grid conversion for meteo input on a separate curvilinear grid	519
B.9	Wind definition according to Nautical convention	522
B.10	Spiderweb grid definition	522
B.11	Definition of truncation wave numbers due to resolution and numerical damping	528
B.12	Schematic representation of the governing processes between suspension layer and fluid mud layer Winterwerp <i>et al.</i> (1999).	569
B.13	A schematic representation of two Delft3D-FLOW modules running simultaneously simulating a fluid mud problem	572
B.14	Vertical grid construction, Z -model	577
B.15	Inserting appropriate keywords to switch on the Z -grid-model	577
B.16	Defining cut-cell and 45 degree closed boundaries	578
B.17	45 degrees staircase closed boundary	579
B.18	Cut-cell closed boundary (not related to the data specified in the example above)	580
B.19	Schematic representation of the free surface boundary condition for the pressure correction.	586
B.20	Area where a non-hydrostatic pressure is taken into account	588
B.21	Defining the Non-hydrostatic solver using the Z -model in the FLOW-GUI	590
B.22	Example of grid refinement	595
B.23	Example of coupling of models with a different dimension	596
B.24	Schematised island without domain decomposition	596
B.25	Schematised island with domain decomposition	597
B.26	Example of grid refinement in the horizontal direction	599
B.27	Problem layout sketch	604
B.28	Schematic axisymmetric bubble plume with entrainment of water by the rising bubbles	621
B.29	Definition of near-, mid- and far-field of the circulation induced by a bubble screen and the vertical profile of the vertical (downward) velocity in the mid-field circulation cell	622
B.30	Flux of entrained water as function of air flow under atmospheric conditions and height above the nozzle in stagnant non-stratified water based on experiments in (Milgram, 1983) <i>The injected air flux is given in [Nm³/s] where N stands for normal atmospheric conditions.</i>	623

B.31	Schematic overview of the introduction of cold hypolimnion water into the lower part of the warmer epilimnion by a bubble plume. Above the first plunge point the second plunge point creates an intrusion and recirculation inside the epilimnion.	624
B.32	The mid field averaged heat equation with vertical distribution of sources/sinks and vertical (downward) velocity profile and the model equation applied to a single grid box.	629
B.33	Initial temperature profile (blue) and after application of a bubble screen at $z = -16$ m (red line). On horizontal axis the temperature (in $^{\circ}\text{C}$) and on the vertical axis the vertical position in the water column (in m).	631
B.34	Explicit exchange of water levels and discharges between Delft3D-FLOW and SOBEK.	632
B.35	Coupling of the Delft3D-FLOW and SOBEK grids.	633
B.36	Enable Delft3D-FLOW in SOBEK settings and select MDF-file and communication-file.	636
B.37	Remapping of two near-bed layers to an equidistant layering. Figure from Platzek <i>et al.</i> (2012).	640
D.1	Steps to determine if a 3D model is required	668
E.1	Left: items with the same (array) number. Right: a computational control volume	672
E.2	Lower-left (left) and lower right (right) computational grid cell	672
E.3	Definition sketch of a $(12 * 7)$ staggered grid with grid enclosure (thick line) and numerical grid	673
E.4	Location and combination of water level and velocity controlled open boundaries	674
E.5	Straight channel with $10 * 5$ computational grid cells	675
F.1	Hydrodynamics selection window with the <i>Tools</i> option	678
F.2	Additional tools window with the NESTHD1 and NESTHD2 options	678
F.3	Specification of input and output files for NESTHD1	679
F.4	Specification of input and output files for NESTHD2	680
F.5	Overview grids overall and nested models	680

List of Tables

4.1	Overview of the main attribute files	21
4.2	Time step limitations shallow water solver Delft3D-FLOW	39
4.3	Definition of open and closed boundaries.	51
5.1	Overview of attribute files.	113
6.1	Simulation performance on different operating systems	159
9.2	Overview of eddy viscosity options in Delft3D-FLOW	188
9.3	Overview of eddy diffusivity options in Delft3D-FLOW	192
9.4	Frequencies, phases and amplitude on alongshore waterlevel boundary and corresponding frequenties, phases and amplitudes for the cross-shore Neumann boundaries	206
9.5	Fitting coefficients for wave/current boundary layer model	234
9.6	Albedo coefficient and cloud function	241
9.7	Terms of the heat balance used in heat model 1	247
9.8	Terms of the heat balance used in heat model 2	247
9.9	Terms of the heat balance used in heat model 4	248
9.10	Terms of the heat balance used in heat model 5	248
9.11	Summary of time dependent input data of the heat flux models	249
9.12	Constants of major tidal modes	251
10.1	Time step limitations shallow water solver Delft3D-FLOW	278
11.1	Additional transport relations	338
11.2	Overview of the coefficients used in the various regression models (Soulsby <i>et al.</i> , 1993a)	353
11.3	Overview of the coefficients used in the various regression models, <i>continued</i> (Soulsby <i>et al.</i> , 1993a)	353
12.1	Available advection and diffusion schemes in the Z -layer model (for comparison also the options available in the σ -model have been included).	368
12.2	Time step limitations shallow water solver Delft3D-FLOW	371
A.2	Print flags for map-data	408
A.3	Print flags for history-data	408
A.3	Print flags for history-data	409
A.4	Storage flags for map-data	409
A.5	Storage flags for history-data	409
A.5	Storage flags for history-data	410
A.6	Optional output flags under <i>Additional parameters</i>	410
A.7	Dredge and dump input file with keywords	454
A.8	Vegetation input file with keywords	471
B.1	Special features of Delft3D-FLOW	477
B.1	Special features of Delft3D-FLOW	478
B.2	Default parameter settings for the SGS model	528
B.3	Sediment input file with keywords	530
B.4	Options for sediment diameter characteristics	533
B.5	Sediment input file without keywords	534
B.5	Sediment input file without keywords	535
B.6	Morphological input file with keywords	536
B.7	Morphological input file without keywords	545
B.7	Morphological input file without keywords	546

B.8	Additional transport relations	548
B.9	Transport formula parameters	548
B.10	Sediment transport formula input file with keywords	550
B.11	Initial bed composition file keywords	566
B.12	Example of vertical grid refinement	600
B.13	Example of a < <i>name.ddb</i> > file	602
B.14	Bedform keywords in mdf file	612
C.1	Astronomical constituents	661

DRAFT

1 A guide to this manual

1.1 Introduction

This User Manual concerns the hydrodynamic module, Delft3D-FLOW, of the Delft3D software suite. To make this manual more accessible we will briefly describe the contents of each chapter and appendix.

If this is your first time to start working with Delft3D-FLOW we suggest you to read and practice the getting started of [chapter 3](#) and the tutorial of [chapter 5](#). These chapters explain the user interface options and guide you through the definition of your first simulation.

Chapter 2: Introduction to Delft3D-FLOW, provides specifications of Delft3D-FLOW, such as the areas of applications, the standard and specific features provided, the required computer configuration and how to install the software.

Chapter 3: Getting started, explains the use of the overall menu program, which gives access to all Delft3D modules and to the pre- and post-processing tools. Last but not least you will get a first introduction into the FLOW Graphical User Interface (GUI), used to define the input required for a flow simulation.

Chapter 4: Graphical User Interface, provides practical information on the selection of all parameters and the tuning of the model.

Chapter 5: Tutorial, emphasis at giving you some first hands-on experience in using the FLOW-GUI to define the input of a simple problem, in verifying this input, in executing the simulation and in inspecting the results.

Chapter 6: Execute a scenario, discusses how to verify and execute a scenario and provides information on run times and file sizes.

Chapter 7: Visualise results, explains in short the visualisation of results. It introduces the post processing program GPP to visualise or animate the simulation results.

Chapter 8: Manage projects and files, provides a detailed insight into the managing of projects and scenarios.

Chapter 9: Conceptual description, describes the theoretical physics modelled in Delft3D-FLOW.

Chapter 10: Numerical aspects of Delft3D-FLOW, discusses the various grids, grid-numbering etc., as well as all practical matters about the implications of parameter selections.

Chapter 11: Sediment transport and morphology, describes the three-dimensional transport of suspended sediment, bedload transport and morphological updating of the bottom.

Chapter 12: Fixed layers in Z-model, the concept of fixed, horizontal layers in the vertical grid are given.

References, provides a list of publications and related material on the Delft3D-FLOW module.

Glossary of terms, contains a list and explanations of the terms and abbreviations used in this manual.

Appendix A: Files of Delft3D-FLOW, gives a description of the files that can be used in the input of Delft3D-FLOW. Generally, these files are generated by the FLOW-GUI and you need not to be concerned about their internal details. However, in certain cases it can be useful to know these details, for instance to generate them by means of other utility programs.

Appendix B: Special features of Delft3D-FLOW, gives an overview and description of the additional functions provided by Delft3D-FLOW. An additional function provides specific functionalities which are not yet supported by the user interface, but which are recognised by a keyword in the MDF-file with one or more values. This value can be a string of characters referring to a file that contains additional input parameters for this function.

The available features are listed in [Table B.1](#). They include e.g. decay rate for constituents, hydraulic structures, space varying Coriolis coefficients, evaporation and precipitation, space varying wind and pressure, horizontal large eddy simulation, 3D sediment and morphology, fluid mud, Z-model, a non-hydrostatic module, user-defined functions, domain decomposition, surfbeat or roller model and trachytopes.

Appendix C: Astronomical constituents, this appendix gives a complete overview of the astronomical components supported by Delft3D-FLOW. For each component is given: its name, angular frequency, amplitude in the equilibrium tide and the relation if an amplitude coupling exists with other components.

Appendix D: Some modelling guidelines, this appendix discusses some guidelines to determine when you need a 3D computation and which vertical turbulence model you need, given the type of modelling application.

Appendix E: Computational grid, discusses the location of open and closed boundaries on the staggered grid used in Delft3D-FLOW. The definition and use of the grid enclosure is discussed in detail. Reading this appendix is suggested when you want to know all the details of the staggered grid and specific implementation aspects and consequences. For normal use of Delft3D-FLOW you can skip this appendix.

Appendix F: Delft3D-NESTHD, discusses the steps to generate boundary conditions for a nested Delft3D-FLOW model. In case the hydrodynamic and transport boundary conditions of a Delft3D-FLOW model are generated by a larger (overall) model we speak of a nested model. Nesting in Delft3D-FLOW is executed in three steps, using two separate utilities and the Delft3D-FLOW program.

1.2 Manual version and revisions

This manual applies to Delft3D-FLOW version 6.00.00, and FLOW-GUI version 3.43.05.

1.3 Typographical conventions

Throughout this manual, the following conventions help you to distinguish between different elements of text to help you learn about the FLOW-GUI.

Example	Description
Module Project	<p>Title of a window or a sub-window are given in bold. Sub-windows are displayed in the Module window and cannot be moved.</p> <p>Windows can be moved independently from the Module window, such as the Visualisation Area window.</p>
Save	<p>Item from a menu, title of a push button or the name of a user interface input field.</p> <p>Upon selecting this item (click or in some cases double click with the left mouse button on it) a related action will be executed; in most cases it will result in displaying some other (sub-)window.</p> <p>In case of an input field you are supposed to enter input data of the required format and in the required domain.</p>
<\tutorial\wave\swan-curvi> <siu.mdw>	<p>Directory names, filenames, and path names are expressed between angle brackets, <>. For the Linux and UNIX environment a forward slash (/) is used instead of the backward slash (\) for PCs.</p>
"27 08 1999"	<p>Data to be typed by you into the input fields are displayed between double quotes.</p> <p>Selections of menu items, option boxes etc. are described as such: for instance 'select Save and go to the next window'.</p>
delft3d-menu	<p>Commands to be typed by you are given in the font Courier New, 10 points.</p>
	<p>In this User manual, user actions are indicated with this arrow.</p>
[m s ⁻¹] [-]	<p>Units are given between square brackets when used next to the formulae. Leaving them out might result in misinterpretation.</p>

1.4 Changes with respect to previous versions

Version	Description
3.15	<p>Meteo input (wind, pressure, etc.) modified (FLOW v3.60.01.02 and higher), see Appendices A.2.8 and B.7.</p> <p>Bubble screen added.</p> <p>1D–3D Coupling added.</p>

Version	Description
3.14	<p>Section 9.3.3: UNESCO formulation added. Chapter 12.12, References: References on UNESCO formulation added.</p> <p>Equation (9.173) corrected. Equation (11.94): — before max changed to +. Equation (11.97): V replaced by q.</p> <p>Section A.2.15, example <*.bct> file: time function should be time-function. Section A.2.21, example <*.dis> file: time function should be time-function. Section B.8.4, example <*.bcc> file: time function should be time-function. Section B.8.5, example <*.mft> file: time function should be time-function. Section B.9.1.3, description of η and η added. Section B13.4, runprocedure adjusted to use arguments for tdatom.exe and trisim.exe.</p> <p>Section 9.3.1: description horizontal viscosity extended taking into account Subgrid scale viscosity and HLES viscosity. Momentum equation adapted. Section 9.3.2: description horizontal diffusivity extended taking into account Subgrid scale viscosity and HLES diffusivity. Transport equation adapted.</p>
3.13	<p>New keyword (<code>IniSedThick</code>) for <i>Initial sediment layer thickness at the bed</i>. Unit is metre. GUI reads old keyword and converts data to new keyword.</p> <p>New functionality: Neumann boundaries described. New functionality: Time-varying morphological scale factor, see B.8.5. New functionality: Maximum number of constituents (pollutants and/or sediments) increased to 99. New functionality: if FLOW runs online with WAVE, and wind is active in the FLOW simulation, this wind can be used in the WAVE simulation. New functionality: Section 6.5 (Command-line arguments) added. New functionality: Initial conditions from a map-file described. Online Delft3D-WAVE and Online coupling is now possible. GUI output file <attribute_files.runid> changed in <runid.fil> with runid not truncated. Limitation of 4 Gb for NEFIS files added. Maximum number of discharges increased to 500. Units for reflection parameter at open boundaries added. New functionality: reflection parameter for discharge boundaries implemented. MENU screens updated. Astronomical component A0 cannot be corrected by a <*.cor> file. New functionality: The name of the MDF-file may be up to 256 characters, i.e. the runid is not limited anymore to 3 characters. In Data Group <i>Numerical parameters</i>, <i>Depth specified at cell corners or cell centres</i> introduced. Text adjusted accordingly. Chapter 11 updated. Appendix B.8 updated. Appendix B.13: Before using <model_id.ddb> you have to replace the grid filenames by the MDF-filenames. Appendix F.1: Restriction added: The boundaries of the nested model may not consist of a single grid cell.</p>

Version	Description
3.12	<p>Chapter 4: the contribution from the HLES sub-grid model will be added to the background values (FLOW v3.50.09.02 and higher).</p> <p>Chapter 4 updated with online coupling, online Delft3D-WAVE.</p> <p>Chapter 4 updated with latest GUI developments (Cartesian or spherical grids, sediment and morphology, dredging and dumping, specification of astronomical flow boundary conditions changed, no slip condition added, Horizontal Large Eddy Simulations, heat flux model parameters added, specification of wind moved to Physical parameters, specification of tide generating forces moved to Physical parameters, additional drying and flooding options, momentum solver options).</p> <p>In chapter 4 emphasized that the stop time may not be too far ahead of the Reference date.</p> <p>Also in chapter 4 the description of <i>In-out</i> discharges improved.</p> <p>In chapter 7 Delft3D-GIS updated to GISVIEW.</p> <p>Theoretical background of Sediment and morphology updated and moved from Appendix B.8 to a new chapter 11.</p> <p>Inclusion of Z-model (horizontal layers) and Non-hydrostatic approach; Chapters 9 and 10 are amended. New Chapter, 12, is dedicated for specifics of the Z-model.</p> <p>C(urrent)D(eflection)W(all)-features added (for σ- and Z-model).</p> <p>Special treatment for 'staircase' closed boundary added for Z-model.</p> <p>Reference and Glossary updated and renamed to chapter 12.12 and 12.12 respectively.</p> <p>Appendix B.10: Input description for Z-model.</p>
3.11	<p>Update of MENU windows with second postprocessing tool Delft3D-QUICKPLOT.</p> <p>Operation of <i>Change working directory</i> updated in chapter 3.</p> <p>In chapter 10: description of the <i>Flooding Scheme</i>.</p> <p>Section 10.4.3: Description of AOI method removed.</p> <p>Section 10.6.2: Equation (10.35) 0 and 1 interchanged.</p> <p>Appendix B.8: Update of Sediment and Morphology latest developments.</p> <p>Appendix B.14 added: Surfbeat/roller model.</p>
3.10	Reference version for these change notes.

DRAFT

2 Introduction to Delft3D-FLOW

Deltares has developed a unique, fully integrated computer software suite for a multi-disciplinary approach and 3D computations for coastal, river and estuarine areas. It can carry out simulations of flows, sediment transports, waves, water quality, morphological developments and ecology. It has been designed for experts and non-experts alike. The Delft3D suite is composed of several modules, grouped around a mutual interface, while being capable to interact with one another. Delft3D-FLOW, which this manual is about, is one of these modules.

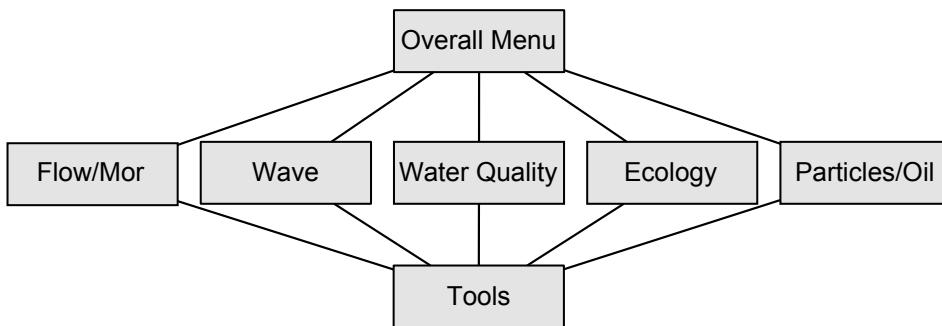
Delft3D-FLOW is a multi-dimensional (2D or 3D) hydrodynamic (and transport) simulation program which calculates non-steady flow and transport phenomena that result from tidal and meteorological forcing on a rectilinear or a curvilinear, boundary fitted grid. In 3D simulations, the vertical grid is defined following the σ co-ordinate approach.

2.1 Areas of application

- ◊ Tide and wind-driven flows (i.e. storm surges).
- ◊ Stratified and density driven flows.
- ◊ River flow simulations.
- ◊ Simulations in deep lakes and reservoirs.
- ◊ Simulation of Tsunamis, hydraulic jumps, bores and flood waves.
- ◊ Fresh-water river discharges in bays.
- ◊ Salt intrusion.
- ◊ Thermal stratification in lakes, seas and reservoirs.
- ◊ Cooling water intakes and waste water outlets.
- ◊ Transport of dissolved material and pollutants.
- ◊ Online sediment transport and morphology.
- ◊ Wave-driven currents.
- ◊ Non-hydrostatic flows.

2.2 Standard features

- ◊ Tidal forcing.
- ◊ The effect of the Earth's rotation (Coriolis force).
- ◊ Density driven flows (pressure gradients terms in the momentum equations).
- ◊ Advection-diffusion solver included to compute density gradients with an optional facility to treat very sharp gradients in the vertical.
- ◊ Space and time varying wind and atmospheric pressure.
- ◊ Advanced turbulence models to account for the vertical turbulent viscosity and diffusivity based on the eddy viscosity concept. Four options are provided: k- ε , k-L, algebraic and constant model.
- ◊ Time varying sources and sinks (e.g. river discharges).
- ◊ Simulation of the thermal discharge, effluent discharge and the intake of cooling water at any location and any depth.
- ◊ Drogue tracks.
- ◊ Robust simulation of drying and flooding of inter-tidal flats.

**Figure 2.1:** System architecture of Delft3D

2.3 Special features

- ◊ Various options for the co-ordinate system (rectilinear, curvilinear or spherical).
- ◊ Built-in automatic switch converting 2D bottom-stress coefficient to 3D coefficient.
- ◊ Built-in anti-creep correction to suppress artificial vertical diffusion and artificial flow due to σ -grids.
- ◊ Built-in switch to run the model in either σ -model or in Z -model.
- ◊ Various options to model the heat exchange through the free water surface.
- ◊ Wave induced stresses and mass fluxes.
- ◊ Influence of waves on the bed shear stress.
- ◊ Optional facility to calculate the intensity of the spiral motion phenomenon in the flow (e.g. in river bends) which is especially important in sedimentation and erosion studies (for depth averaged — 2DH — computations only).
- ◊ Optional facility for tidal analysis of output parameters.
- ◊ Optional facility for special points such as 3D gates, Current Deflecting Wall (CDW) floating structures, bridges, culverts, porous plates and weirs.
- ◊ Optional facility to switch between a number of advection solvers.
- ◊ Optional facility for user-defined functions.
- ◊ Domain decomposition.

2.4 Coupling to other modules

The hydrodynamic conditions (velocities, water elevations, density, salinity, vertical eddy viscosity and vertical eddy diffusivity) calculated in the Delft3D-FLOW module are used as input to the other modules of Delft3D, which are (see [Figure 2.1](#)):

module	description
Delft3D-WAVE	short wave propagation
D-Water Quality	far-field water quality
D-Waq PART	mid-field water quality and particle tracking
Delft3D-ECO	ecological modelling
Delft3D-SED	cohesive and non-cohesive sediment transport

2.5 Utilities

For using Delft3D-FLOW the following utilities are important:

module	description
RGFGRID QUICKIN	for generating curvilinear grids for preparing and manipulating grid oriented data, such as bathymetry or initial conditions for water levels, salinity or concentrations of constituents.
Delft3D-TRIANA	for performing off-line tidal analysis of time series generated by Delft3D-FLOW
Delft3D-TIDE	for performing tidal analysis on time-series of measured water levels or velocities
Delft3D-NESTHD	for generating (offline) boundary conditions from an overall model for a nested model
GPP Delft3D-QUICKPLOT	for visualisation and animation of simulation results a second tool for visualisation and animation of simulation results

For details on using these utility programs you are referred to the respective User Manual.

2.6 Installation and computer configuration

See the Delft3D Installation Manual.

DRAFT

3 Getting started

3.1 Overview of Delft3D

The Delft3D program suite is composed of a set of modules (components) each of which covers a certain range of aspects of a research or engineering problem. Each module can be executed independently or in combination with one or more other modules. The information exchange between modules is provided automatically by means of a so-called communication file; each module writes results required by another module to this communication file and reads from the file the information required from other modules. Other, module-specific, files contain results of a computation and are used for visualisation and animation of results.

Delft3D is provided with a menu shell through which you can access the various modules. In this chapter we will guide you through some of the input screens to get the look-and-feel of the program. In [chapter 5, Tutorial](#), you will learn to define and run a simple scenario.

3.2 Starting Delft3D

To start Delft3D:

- ◊ On an MS Windows platform: select Delft3D in the *Programs* menu or click on the Delft3D icon on the desktop.
- ◊ On Linux machines: type `delft3d-menu` on the command line.

Next the title window of Delft3D is displayed, [Figure 3.1](#):



Figure 3.1: Title window of Delft3D

After a short while the main window of the Delft3D-MENU appears, [Figure 3.2](#).

Several menu options are shown. In [Figure 3.2](#) all options are sensitive.

For now, only concentrate on exiting Delft3D-MENU, hence:

- ◊ Click on the *Exit* push button.

The window will be closed and you are back in the Windows Desktop screen for PCs or on the command line for Linux workstations.

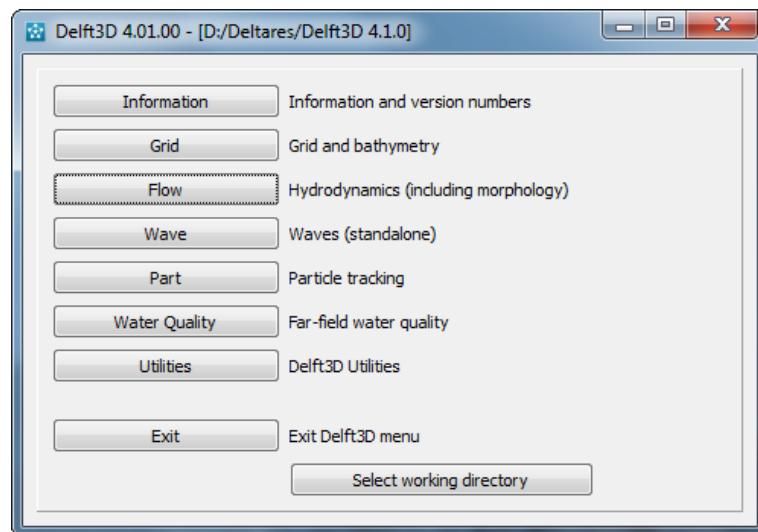


Figure 3.2: Main window Delft3D-MENU



Remark:

- ◊ In this and the following chapters several windows are shown to illustrate the presentation of Delft3D-MENU and Delft3D-FLOW. These windows are grabbed from the PC-platform. For Linux workstations the content of the windows is the same, but the colours may be different. On the PC-platform you can set your preferred colours by using the Display Properties.

3.3 Getting into Delft3D-FLOW

To continue restart the menu program as indicated above.

- ◊ Click the *Flow* button.

Next the selection window for Hydrodynamics is displayed for preparing a flow input (MDF-)file or wave input (MDW-)file, to execute a computation in foreground (including online WAVE or online coupling), to inspect the report files with information on the execution and to visualise the results: [Figure 3.3](#).

Before continuing with any of the selections of this **Hydrodynamics (including morphology)** window, you must select the directory in which you are going to prepare scenarios and execute computations:

- ◊ Click the *Select working directory* button.

Next the **Select working directory** window, [Figure 3.4](#), is displayed (your current directory may differ, depending on the location of your Delft3D installation).

- ◊ Browse to and open the <Tutorial> sub-directory of your Delft3D Home-directory.
- ◊ Open the <flow> directory.
- ◊ Enter the <friesian_tidal_inlet> sub-directory and close the **Select working directory** window by clicking *OK*, see [Figure 3.5](#).

Next the **Hydrodynamics (including morphology)** window is re-displayed, but now the chan-

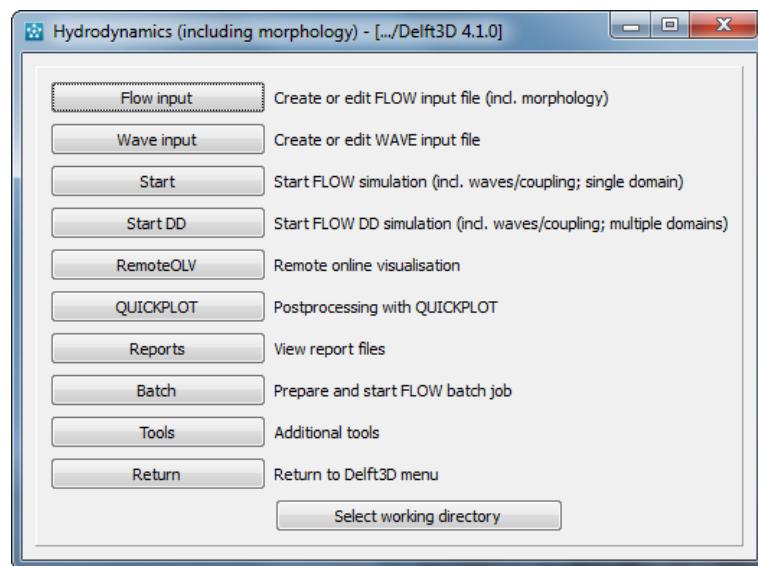


Figure 3.3: Selection window for Hydrodynamics

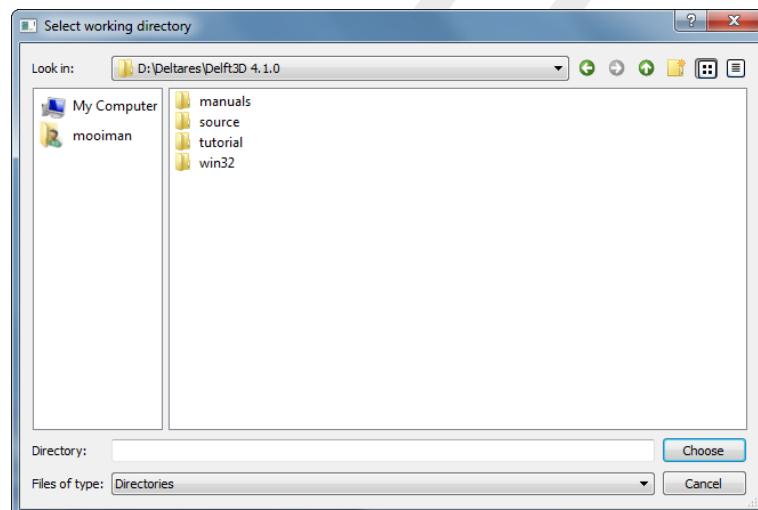


Figure 3.4: Select working directory window

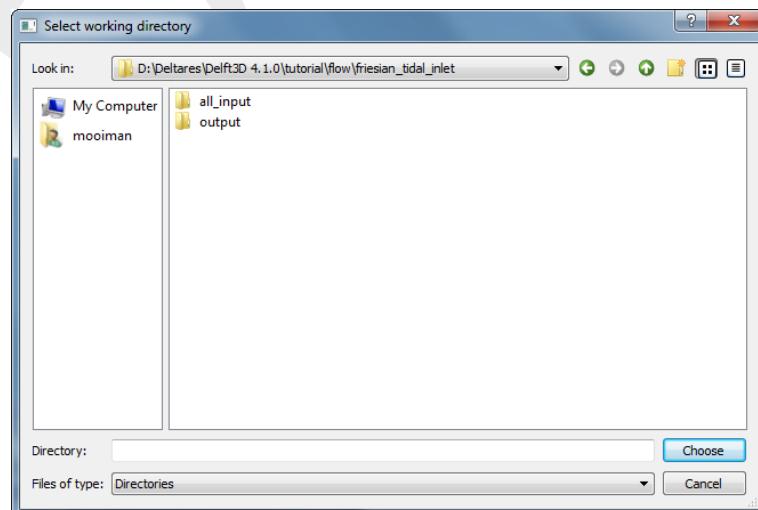


Figure 3.5: Select working directory window to set the working directory to <flow\friesian_tidal_inlet>



Figure 3.6: The current working directory is not shown in the title bar due to its length

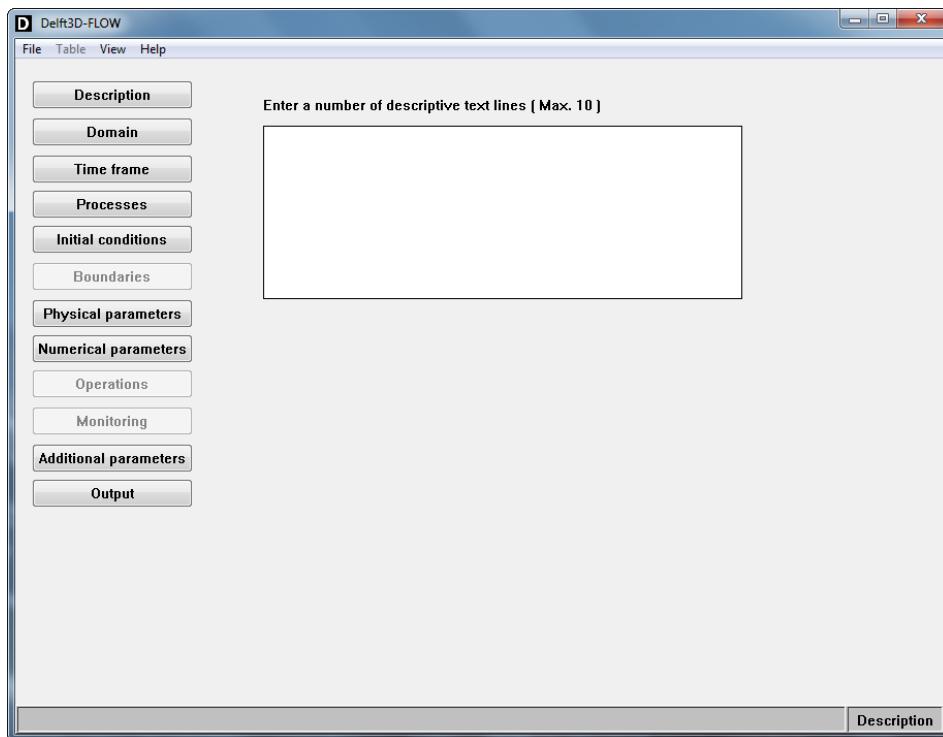


Figure 3.7: Main window of the FLOW Graphical User Interface

ged current working directory is displayed in the title bar (if the name is not too long), see [Figure 3.6](#).



Remark:

- ◊ In case you want to start a new project for which no directory exists yet, you can select  in the **Select working directory** window to create a new folder.

In the main **Hydrodynamics (including morphology)** menu, [Figure 3.3](#), you can define, execute and visualise a scenario. In this guided tour through Delft3D-FLOW we limit ourselves to inspecting some windows of the FLOW Graphical User Interface (GUI).

Hence:

- ◊ Click on *Flow input*.

The FLOW-GUI is loaded and the primary input screen is opened, [Figure 3.7](#).

The purpose of this FLOW-GUI is to create the input file of Delft3D-FLOW, also called the “Master Definition Flow” file (MDF-file) which contains all information to execute a flow simulation.



Figure 3.8: Menu bar of the FLOW-GUI

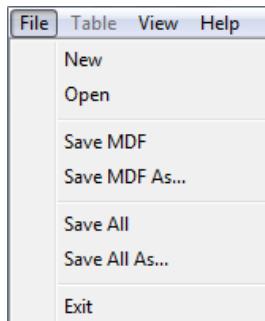


Figure 3.9: File drop down menu

3.4 Exploring some menu options

The menu bar of the FLOW-GUI displays four options:

- | | |
|--------------|--|
| <i>File</i> | select and open an MDF-file, save an MDF-file, save an MDF-file under a different name, save attribute files or ‘exit’ the FLOW-GUI. |
| <i>Table</i> | tool to change table oriented data by adding or deleting rows or values. |
| <i>View</i> | visualisation area or list of attribute files. |
| <i>About</i> | About information. |

Each option provides one or more selections; for instance, clicking on *File* enables the selections:

- | | |
|--------------------|--|
| <i>New</i> | to clean-up the internal data structure and start with a new scenario. |
| <i>Open</i> | to open an existing MDF-file. |
| <i>Save MDF</i> | to save the MDF-data under its current name. |
| <i>Save MDF As</i> | to save the MDF-data under a new name. |
| <i>Save All</i> | to save all attribute data in the current attribute files + MDF-file. |
| <i>Save All As</i> | to save all attribute data under a new name + MDF-file. |
| <i>Exit</i> | to exit the FLOW-GUI and return to the Hydrodynamics (including morphology) window. |

The input parameters that define a hydrodynamic scenario are grouped into Data Groups. These Data Groups are represented by the large grey buttons at the left of the main window. Upon starting the FLOW-GUI, [Figure 3.7](#) is displayed with the Data Group *Description* selected and displayed. The area to the right of the Data Groups is called the canvas area. This canvas area will be dynamically filled with input fields, tables, or list boxes to define the various kinds of input data required for a simulation. In [Figure 3.7](#) the *Description* text box is displayed in the canvas area.

Click on a Data Group and see what happens. For example, clicking the *Domain* button and next the sub Data Group *Grid parameters*, will result in the window shown in [Figure 3.10](#). The Tutorial in [chapter 5](#) will make you become fully acquainted with the various input windows that result from this main window.

You are encouraged to explore the various Data Groups and sub-windows to get a first impression on the items the Data Groups are composed of. Though several input items are

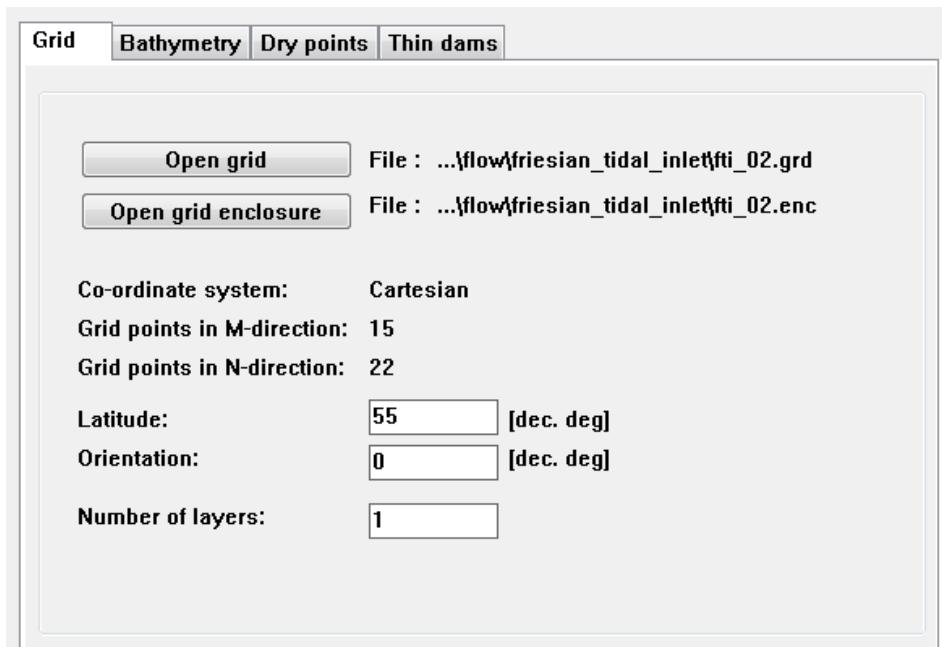


Figure 3.10: Data Group Domain selection and input fields

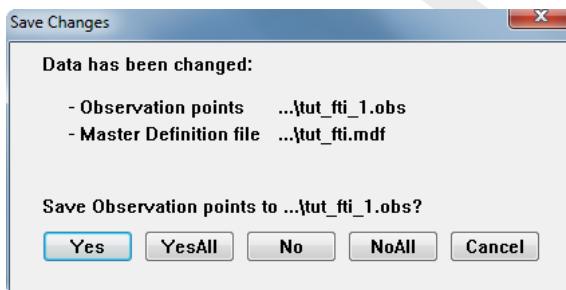


Figure 3.11: Save changes window

related there is no fixed or prescribed order in defining the input data. Occasionally you might get a warning or error message that some data is not saved or not consistent with earlier defined data; during this introduction you can neglect these messages and press the *Ignore* button if requested. No harm will be done on existing input files as you are not going to save the input data of this exercise.

3.5 Exiting the FLOW-GUI

To exit the FLOW-GUI:

- ◊ From the *File* menu, select *Exit*.

If you have made any change to any input field and have not explicitly saved both the attribute data and/or the MDF-data [Figure 3.11](#) is displayed.

In this case only the MDF-data was not saved; if you have changed data that must be saved into a so-called attribute file, the unsaved attribute files will be listed.

Select one of the options displayed:

Yes save the first unsaved data item, request for a file name if not yet

	defined and proceed to the next unsaved data item.
YesAll	save all unsaved data items and request a file name if not yet defined.
No	don't save the first unsaved data item; proceed to the next unsaved data.
NoAll	exit without saving any unsaved data item.
Cancel	abort this <i>Exit</i> action and return to the FLOW-GUI.

Neglect any unsaved data and exit:

- ◊ Click *NoAll*.

You will be back in the **Hydrodynamics (including morphology)** window of the Delft3D-MENU program, [Figure 3.3](#).

Ignore all other options and just:

- ◊ Click *Return* to return to the main window of Delft3D-MENU, [Figure 3.2](#).
- ◊ Click *Exit*.

The window is closed and the control is returned to the desktop or the command line.

In this Getting Started session you have learned to access the FLOW-GUI and to load and inspect an existing input (MDF-)file.

We encourage new users next to run the tutorial described in [chapter 5](#).

DRAFT

4 Graphical User Interface

4.1 Introduction

In order to set up a hydrodynamic model you must prepare an input file. All parameters to be used originate from the physical phenomena being modelled. Also from the numerical techniques being used to solve the equations that describe these phenomena, and finally, from decisions being made to control the simulation and to store its results. Within the range of realistic values, it is likely that the solution is sensitive to the selected parameter values, so a concise description of all parameters is required. The input data defined is stored into an input file which, as you may recall, is called the Master Definition Flow file or MDF-file.

If you have not used Delft3D-FLOW before, or if you are not familiar with the FLOW Graphical User Interface (GUI) we suggest you to execute the tutorial given in [chapter 5](#) first and then return to this chapter.

In [section 4.2](#) we discuss some general aspects of the MDF-file and its attribute files. In [section 4.3](#) we discuss shortly the filenames and their extension. In [section 4.4](#) we discuss working with the FLOW-GUI and the **Visualisation Area** window. In [section 4.5](#) we discuss all input parameters, including their restrictions and their valid ranges or domain. In many cases we give a short discussion on the criteria to determine a parameter or to select a certain formulation, such as the turbulence closure model for the vertical turbulent eddy viscosity and turbulent eddy diffusivity. In [section 4.6](#) we discuss saving the MDF-file and exiting the FLOW-GUI. Finally, we discuss in [section 4.7](#) the aspect of importing, removing and exporting of data and their references in the MDF-file.

4.2 MDF-file and attribute files

The Master Definition Flow file (MDF-file) is the input file for the hydrodynamic simulation program. It contains all the necessary data required for defining a model and running the simulation program. In the MDF-file you can define attribute files in which relevant data (for some parameters) is stored. This will be particularly the case when parameters contain a large number of data (e.g. time-dependent or space varying data). The MDF-file and all possible user-definable attribute files are listed and described in [Appendix A](#).

Although you are not supposed to work directly on the MDF-file it is useful to have some ideas on it as it reflects the idea of the designer on how to handle large amounts of input data and it might help you to gain a better idea on how to work with this file.

The basic characteristics of an MDF-file are:

- ◊ It is an ASCII file.
- ◊ Each line contains a maximum of 300 characters.
- ◊ Each (set of) input parameter(s) is preceded by a (set of) *keyword(s)*.

The MDF-file is an intermediate file between the FLOW-GUI and the hydrodynamic simulation program. Being an ASCII-file, it can be transported to an arbitrary hardware platform. Consequently, the hydrodynamic simulation program and the FLOW-GUI do not necessarily have to reside on the same hardware platform. Currently, Delft3D does not support remote or distributed processing, but you can easily write a couple of scripts to run your pre- and postprocessing on one hardware platform and run the computational intensive simulation at an other hardware platform. The results of all modules are written to platform independent binary files, so also these result files you can transfer across hardware platforms without any conversion. Contact our support manager if you need remote or distributed computational

functionalities.

The MDF-file is self contained, i.e. it contains all the necessary information about the model concerned. It can therefore be used as model archive. To maintain a good overview of the file, its length is restricted to 300 columns.

As you will see in [chapter 5](#), after having specified certain types of input parameters you can store them in attribute files. The MDF-file only contains permanent input parameters and references to these attribute files. An overview of the attribute files is given in Section [4.3](#).

If you wish to create attribute files in advance, since supplying long time-series manually is not very practical, you must make sure that the correct input formats are used. The formats of all attribute files (and of the MDF-file itself) are described in detail in [Appendix A](#).

4.3 Filenames and conventions

The names of the MDF-file and its attribute files have a specific structure, some aspects are mandatory while others are only advised or preferred.

MDF-file

The name of an MDF-file must have the following structure: <*runid.mdf*>.

The *runid* may consist of up to 256 alpha-numeric characters and may not contain blanks. The *runid* part of the filename is used as a run-id in the names of the result files to safeguard the link between an MDF-file and the result files. When you have many computations we suggest to use a combination of one alpha- and two numeric-characters followed by a useful name of your project.

Example: <r01-calibration.mdf>.

This file could indicate the flow-input file of the first calibration run of a project.

Result files

The results of a Delft3D-FLOW computation are stored in several types of files:

- ◊ communication file: <com-*runid.def*> and <com-*runid.dat*>.
- ◊ history file: <trih-*runid.def*> and <trih-*runid.dat*>.
- ◊ map file: <trim-*runid.def*> and <trim-*runid.dat*>.
- ◊ drogue file: <trid-*runid.def*> and <trid-*runid.dat*>.
- ◊ restart files <tri-rst.*runid.date.time*>.

The result files are stored in the working or project directory.



Restrictions:

- ◊ Each scenario must have a unique run-id; when you have many computations we suggest to use a character followed by a two digit number.
- ◊ Avoid spaces in a filename, use an underscore instead, i.e. <r01_test.mdf> instead of <r01 test.mdf>.
- ◊ The extension mdf is *mandatory*.

The communication file contains results that are required by other modules, such as the water quality module. The history file contains results of all computed quantities in a number of

Table 4.1: Overview of the main attribute files

Quantity	Filename and mandatory extension
Astronomic correction factors	<name.cor>
Bathymetry or depth	<name.dep>
Bottom roughness	<name.rgh>
Constituents boundary conditions	<name.bcc>
Cross-sections	<name.crs>
Curvilinear grid	<name.grd>
Discharge locations	<name.src>
Discharges rates	<name.dis>
Dredge and dump characteristics	<name.dad>
Drogues or floating particles	<name.par>
Dry points	<name.dry>
Flow boundary conditions (astronomic)	<name.bca>
Flow boundary conditions (harmonic)	<name.bch>
Flow boundary conditions (QH-relation)	<name.bcq>
Flow boundary conditions (time-series)	<name.bct>
Fourier analysis input file	<name.fou>
Grid enclosure	<name.enc>
Horizontal eddy viscosity and diffusivity	<name.edy>
Initial conditions	<name.ini>
Morphology characteristics	<name.mor>
Observation points	<name.obs>
Open boundaries	<name.bnd>
Sediment characteristics	<name.sed>
Temperature model parameters	<name.tem>
Thin dams	<name.thd>
Wind	<name wnd>

user-defined grid points at a user-defined time interval. The map file contains results of all quantities in all grid points at a user-specified time interval.

Attribute files

Attribute files contain certain input quantities, such as monitoring points or time dependent input data such as wind. The names of the main attribute files are basically free, but their extension is mandatory as indicated in [Table 4.1](#).

The name of an attribute file must have the following structure: <*name.extension*>.

Where:

- ◊ <*name.extension*> may consist of up to 256 alpha-numeric characters and may contain (sub-) directories, i.e. the full path.
- ◊ There is no limitation other than the platform dependent limitations; you are referred to your hardware platform manual for details. We suggest to add some continuation character, for instance *-number* to the *name* to distinguish between various updates or modifications of the file, i.e. <*name-number.extension*>.
- ◊ The *extension* is mandatory as indicated in [Table 4.1](#).

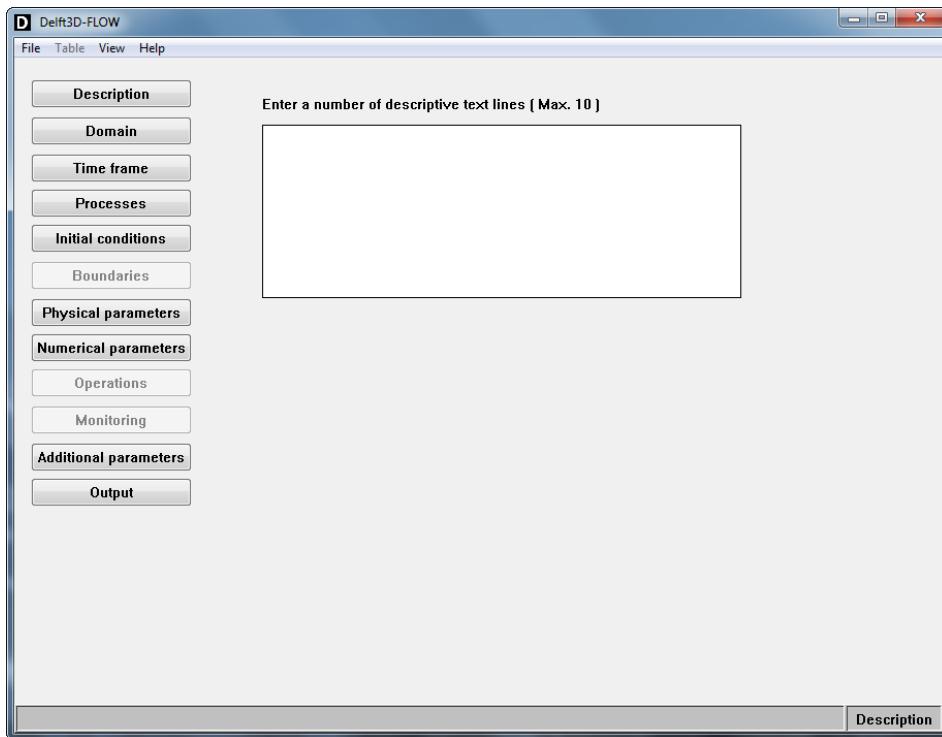


Figure 4.1: Main window of the FLOW-GUI

4.4 Working with the FLOW-GUI

4.4.1 Starting the FLOW-GUI

The purpose of the FLOW-GUI is to provide a graphical user interface which simplifies the preparation of an MDF-file. For your convenience, in this section we briefly recapitulate how to work with the FLOW-GUI (alternatively, consult Chapters 3 and 5). With respect to the parameters in the MDF-file, the FLOW-GUI follows either one of the following options:

- ◊ A single parameter is updated and included in the MDF-file.
- ◊ A reference to an attribute file is updated and included in the MDF-file. An attribute file can be created by the FLOW-GUI if the required data was specified.

To start the FLOW-GUI you must in short execute the following commands, see [chapter 3](#) for details:

- Click the *Delft3D-MENU* icon on the desktop (PC) or execute the command `delft3d-menu` on the command line (Linux and UNIX).
- Click the menu item *FLOW*.
- Change to your project or working directory.
- Click the menu item *Flow input*; the FLOW-GUI will be started and the main window will be opened, see [Figure 4.1](#).

You are now ready to start defining or modifying all input parameters grouped into so-called data groups. In the menu bar you can choose from the following options:

File

For opening and saving an MDF-file, or saving an MDF-file with another name, for saving attribute files under the same name or under a new name, for cleaning up the internal data structure and for exiting the FLOW-GUI. Sub-menus are: *New*, *Open*, *Save MDF*, *Save MDF*

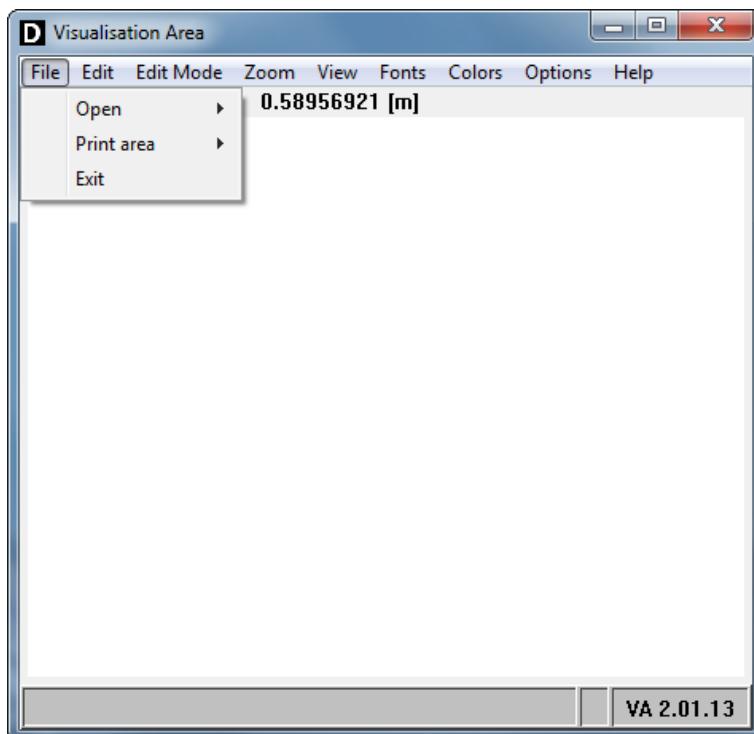


Figure 4.2: Visualisation Area Window

As, Save All, Save All As and Exit.

Table To change attribute files by adding or deleting rows or values in table quantities. Sub-menus are: *Insert row above*, *Copy row above*, *Delete row* and *Copy value to all rows*.

View For viewing the grid related parameters or for listing the attribute files used (only their referenced name, not their contents). Sub-menus are: *Visualisation Area* and *Attribute files*.

Help For getting online and context sensitive help. Sub-menus are: *Contents* and *About*. The first is not implemented yet.

4.4.2 Visualisation Area window

Most grid related data specified in the MDF-file can be visualised and defined in the **Visualisation Area** window. These grid related data are: dry points, thin dams, observation points, drogues, discharges, cross-sections and open boundaries.

Upon selecting *View* → *Visualisation Area* the **Visualisation Area** window will pop up, [Figure 4.2](#).

The visualisation area is still blank, but after you have defined or selected a grid it will display the grid and several grid related quantities.

The main and sub-menus of the **Visualisation Area** window are:

File To *Open* some of the grid defining files, to print (*Print area*) the **Visualisation Area** window or *Exit* (close) the **Visualisation Area** window.

Edit To select one of the grid related quantities that can be visualised.

Edit Mode To *Add*, *Delete*, *Modify* (move) and *View* the quantities selected in

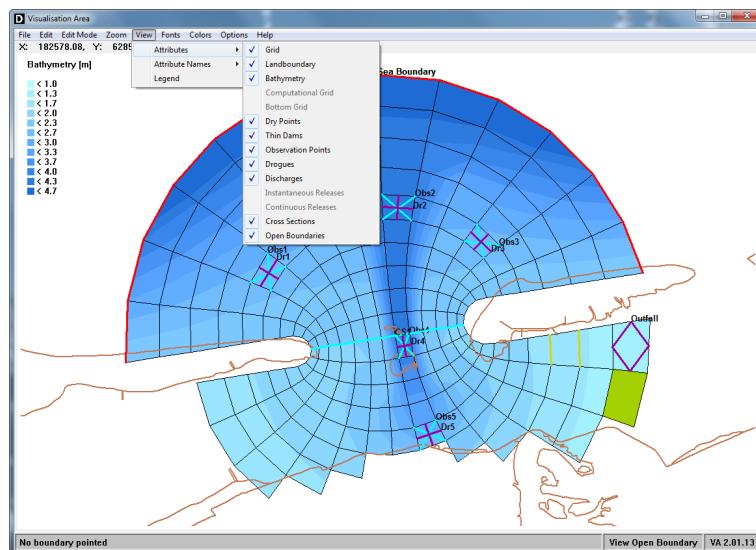


Figure 4.3: Possible selections of View → Attributes

Zoom

Edit.

To *Zoom In* and *Zoom Out* the whole visualisation area, *Zoom Box* in a user-defined area and *Zoom Reset* to return to the initial situation.

View

To switch on or off viewing attributes and/or attribute names. The various selections of *View* → *Attributes* are displayed in [Figure 4.3](#). You can activate (display) or de-activate (hide) the various attributes.

Fonts

To set the font, size, etc. of the attribute names.

Colors

To set the colours for visualising the bathymetry.

Options

To select which quantities will be displayed in the visualisation area and to refresh the display.

Help

For online information of using the **Visualisation Area** window.



Remarks:

- ◊ The *Edit* and *Edit Mode* options make it possible to graphically define, delete, modify (move) or just view quantities of a certain type, without having to type locations manually.
- ◊ If a quantity can have a user-defined name you can fill that in after having defined its location graphically.
- ◊ You can save the data in an attribute file before you change the quantity to be worked on, but you can also postpone this until the end of your input session.
- ◊ The *Edit Mode* remains in its selected mode as long as you are working in the **Visualisation Area** window. This allows you to define (Add) or delete (Delete) all kind of quantities without having to set and reset the *Edit Mode*. The *Edit Mode* shifts back to *View* mode as soon as you leave the **Visualisation Area** window, to prevent unintended modifications of your grid related quantities.
- ◊ Make sure the *Edit Mode* is selected properly, if *not* you might accidentally move a quantity without notice!

To see how each quantity is represented in the **Visualisation Area** display the legend:

➤ Select *View* → *Legend*

and next [Figure 4.4](#) is displayed.



Remark:

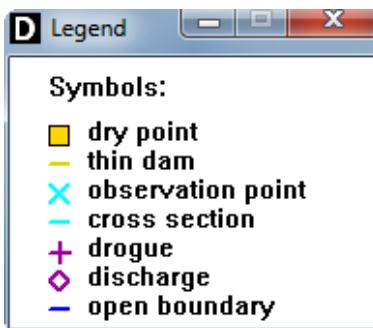


Figure 4.4: Display symbols of all grid related quantities

- ◊ Symbols are grouped in colours, but with different representation to support easy recognition.

4.5 Input parameters of MDF-file

In this section all input parameters of the MDF-file will be described in the order as they appear in the FLOW-GUI. After starting the FLOW-GUI data groups become available for defining or changing the input parameters in the MDF-file.

A data group is a coherent set of input parameters that together define a certain type of input data. For instance, in the Data Group *Operations* you can define all aspects related to a discharge, such as its name, its location, its discharge rate, if the momentum of the discharge is to be taken into account and if so in which direction and last but not least the concentration of all substances released. Several of these items can be specified as a function of time, where the time-series can be specified manually or read from a file.

Some data groups are organised in sub-data groups, such as the Data Group *Domain*, that consists of four sub-data groups: *Grid parameters*, *Bathymetry*, *Dry points* and *Thin dams*.

We will now describe all data groups in consecutive order. For each input quantity we give:

- ◊ A short description of its meaning. In many cases we add a more comprehensive discussion to put the quantity and its use in perspective.
- ◊ The restrictions on its use.
- ◊ The range of allowed values, called its domain, and its default value (if applicable).

Remark:



- ◊ Before you can define grid related quantities, you must define or select the grid structure. When editing an existing input file there is no preferred or mandatory order in which to address the various data groups.

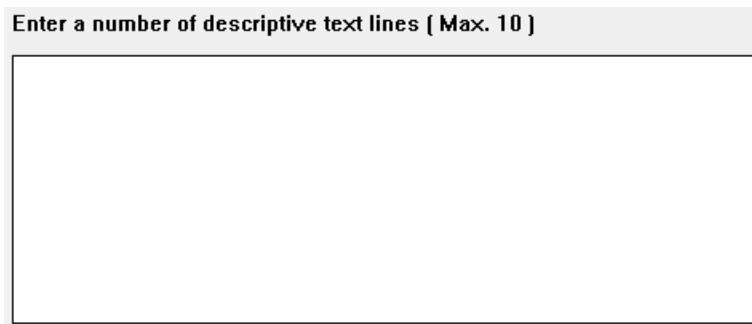


Figure 4.5: Data Group Description

4.5.1 Description

The Data Group *Description* is a text box of up to 10 lines of text, which you can use to describe the purpose of the present model and for discriminating the present run from the (possibly) other runs with the same model. The description is only used for reference. Upon selecting the data group [Figure 4.5](#) is displayed.

If you started from an existing MDF-file its name is displayed above in the title bar.

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Descriptive text	Any printable character		Empty lines	none



Restriction:

- ◊ 10 lines of text each containing a maximum of 30 characters.

4.5.2 Domain

The Data Group *Domain* contains the following sub-data groups: *Grid parameters*, *Bathymetry*, *Dry points* and *Thin dams*.

4.5.2.1 Grid parameters

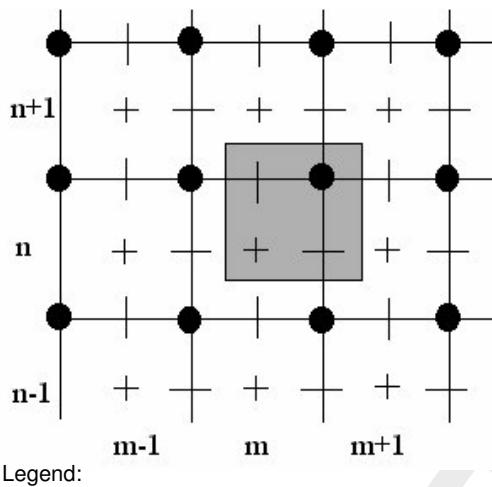
In the sub-data group *Grid parameters* you specify the grid used, the latitude of the model area and the number of (vertical) layers.

Before continuing with discussing the *Grid parameters* you should familiarise yourself with the concept of the staggered grid applied in Delft3D-FLOW.

In a staggered grid not all quantities, such as the water level, the depth, the velocity components or concentration of substances, are defined at the same location in the numerical grid (and thus in the physical space).

The staggered grid applied in Delft3D is given in [Figure 4.6](#).

Closed boundaries are defined through *u*- or *v*-points; open boundaries through either *u*-, *v*- or water level (ζ -) points depending on the type of boundary condition such as velocity or water level.



Legend:

- full lines** the numerical grid
- grey area** items with the same grid indices (m, n)
 - + water level, concentration of constituents, salinity, temperature
 - horizontal velocity component in ξ -direction (also called u - and m -direction)
 - | horizontal velocity component in η -direction (also called v - and n -direction)
- depth below mean (still) water level (reference level)

Figure 4.6: Staggered grid of Delft3D-FLOW

The location of other grid related quantities, such as discharges and observation points are given when appropriate.

In Delft3D-FLOW we support two types of co-ordinate systems in the horizontal:

- ◊ Cartesian: the co-ordinates are in metres
- ◊ Spherical: the co-ordinates are in decimal degrees

For a Cartesian grid you have to specify the latitude of the model area; this will be used to calculate a fixed Coriolis force for the entire area. For a spherical grid the Coriolis force is calculated from the latitude co-ordinates in the grid file and thus varies in the latitude direction. Typically, you use spherical co-ordinates for large areas, such as a regional model.

The type of co-ordinate system is stored in the grid file, together with the number of grid points in both directions, and the co-ordinates of the grid points.

The construction of a suitable curvilinear grid is not a simple task, because the grid must fulfill the following criteria:

- ◊ It must fit as closely as possible to the land-water boundaries (in short land boundaries) of the area to be modelled.
- ◊ It must be orthogonal, i.e. grid lines must intersect perpendicularly.
- ◊ The grid spacing must vary smoothly over the computational region to minimise inaccuracy errors in the finite difference operators.

The Delft3D modelling suite contains the grid generator program RGFGRID that allows you to generate a curvilinear grid (in Cartesian or spherical co-ordinates) with the required resolution and properties. The actual construction of a grid is realised in an iterative procedure allowing

for a stepwise generation of the model grid, working from a coarse version of the grid to finer versions until the required resolution is achieved. RGFGRID provides all kind of features to develop a grid, such as refine or de-refine the grid globally or locally, delete or add locally individual grid cells, define separately a grid in a sub-area and glue it to the overall grid and orthogonalise the grid. RGFGRID provides features to inspect the quality of the grid.

The quality of a grid is to a large extent determined by its orthogonality and the rate with which certain properties change over the area to be modelled (smoothness). A measure for the orthogonality is the angle, or the cosine of the angle, between the grid lines in ξ - and η -direction. A measure for the grid smoothness is the aspect ratio of grid cells (ratio of the grid cell dimension in ξ - and η -direction) and the ratio of neighbouring grid cell dimensions.

As a guideline we suggest the following overall quality criteria:

- ◊ Orthogonality: $\cos(\varphi) < 0.02$, where φ is the angle between the grid lines.
- ◊ Aspect ratio: must be in the range [1 to 2], unless the flow is predominantly along one of the grid lines.
- ◊ Ratio of neighbouring grid cells: should be less than 1.2 in the area of interest up to 1.4 far away.

You can use RGFGRID to inspect these and other properties graphically. For details see the RGFGRID User Manual.

The horizontal resolution of the grid depends on the characteristic length scale of the bathymetry and the land-water boundary and of flow patterns you want to resolve. To resolve an important geometrical or hydrodynamic phenomenon you will need at least 5 grid cells; to resolve a horizontal circulation the grid size should be 1/10th or less of the size of the circulation.

**Remark:**

- ◊ Flooding and drying is less accurate if the grid size increases.

Opening a grid and enclosure file

- ◊ Select *Grid parameters* → *Open grid*, see [Figure 4.7](#). A file window opens in which you can browse to the required directory, and open a <*.grd> file.

After the grid file has been opened, the co-ordinate system used is displayed, as well as the number of grid points in both directions. Next

- Select *Grid parameters* → *Open grid enclosure*, see [Figure 4.7](#). A file window opens in which you can browse to the required directory, and open the <*.enc> file that belongs to your grid file.

A grid enclosure is a closed polygon specified on a grid through the water level points. Its purpose is to define the active or potentially active (i.e. wet) computational cells in the computational domain and the location of the open and closed boundaries. The grid enclosure is generated by the grid generator RGFGRID; its use is under all practical conditions completely transparent to you and you do not need to be aware of the definition and implementation details.

However, if you want to inspect certain files, such as the grid file or the bathymetry file and want to check all details in relation to values and locations, you must be aware of certain details related to definition and implementation. You can find these details in [chapter 10](#) as

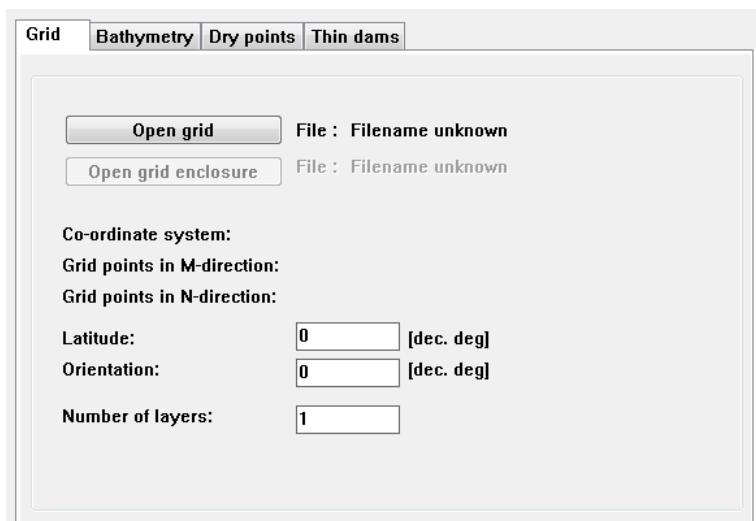


Figure 4.7: Sub-data group Grid parameters

far as related to the numerical scheme, in subsection A.2.3 related to the file structure and content and in Appendix E for a detailed discussion of the grid enclosure and the location of open and closed boundaries.

Remarks:

- ◊ We *strongly suggest* generating the grid and the grid enclosure *in all cases* with the grid generator program RGFGRID.
 - ◊ If no grid enclosure is specified a default polygon is generated through the four corner points of the numerical grid.

Latitude and orientation of the model area

If the grid is defined in Cartesian co-ordinates you have to specify the latitude and orientation of the model:

Latitude Latitude location of the model on Earth.

The Coriolis force is determined by the location of the model area on the Earth's globe, i.e. the angle of latitude (in degrees North). In the northern hemisphere you must enter a positive value; in the southern hemisphere you must enter a negative value.

Remark:

- ◊ The Coriolis force in a spherical model varies in the North-South direction and is determined by the actual latitude.

Orientation

The model orientation is defined as the angle between the true North and the y-axis of the Cartesian co-ordinate system. The angle is positive if the rotation is clockwise, see [Figure 4.8](#).

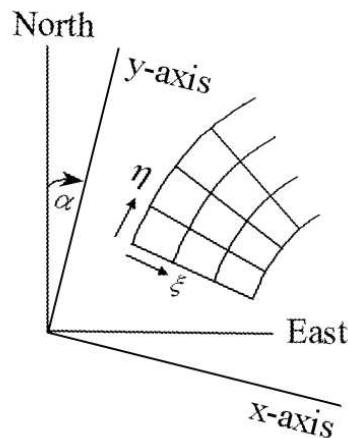


Figure 4.8: Definition sketch grid system to North orientation

Number of layers

For a depth-averaged computation you must set the *Number of layers* to 1. For a 3D computation, the *Number of layers* is larger than 1. Furthermore, you must specify the thickness of the layers in the vertical. Upon setting the *Number of layers* larger than 1, an additional input field **Layer thickness** is displayed, see [Figure 4.9](#).

Co-ordinate system:	Cartesian	
Grid points in M-direction:	15	
Grid points in N-direction:	22	
Latitude:	<input type="text" value="0"/> [dec. deg]	
Orientation:	<input type="text" value="0"/> [dec. deg]	
Number of layers:	<input type="text" value="5"/>	

Layer thickness [%]
120
220
320
420
520

Total: 100 [%]

Figure 4.9: Specifying the layers thickness

In the vertical direction, two types of vertical grid with distinctive layer thickness characteristics are supported:

- ◊ With the σ -grid in the vertical the layer thickness varies with the depth, and the number of active layers is constant (denoted in this manual as the σ -model) and
- ◊ Z -grid; here the layer thickness is fixed and the number of active layers varies with the depth. The layer thickness at the top is however determined by the actual water level and at the bottom by the local topography. The model using this grid is referred to as the Z -model.

To achieve a constant number of layers in the σ -model, a σ co-ordinate transformation in the vertical is used. You can specify an arbitrary number or distribution as long as the total sum of the layers is 100 %.

The thickness of a layer is defined as:

- ◊ a percentage of the initial water depth for the Z -model and

- ◊ a percentage of the, time varying, water depth for the σ -model.

You can specify an arbitrary number or distribution as long as the total sum of the layers is 100 %. See Chapters 9, 10 and 12 for more details of the vertical co-ordinate system.

Remark:

- ◊ In the σ -model layer 1 corresponds to the surface layer while in the Z -model layer 1 refers to the bottom layer.



To resolve the logarithmic profile of the horizontal velocity components in the vertical the thickness of the bottom layer should be small. It is recommended to choose the bed layer thickness to be about 2 % of the water depth. The variation in the layer thickness should not be large, i.e. the layer thickness must have a smooth distribution. An indicative value for the variation-factor for each layer is 0.7 to 1.4. Going from bottom to surface the suggested layer thickness should not exceed 3 %, 4.5 %, 6.75 %, etc. of the water depth. For a ten σ -layers example the suggested layer thickness is {2, 3, 4, 6, 8, 10, 12, 15, 20, 20} %.

If wind is one of the important driving forces also the layer thickness near the surface should not exceed 2 % of the water depth. Thus, going down from the surface the layer thickness should not exceed 3 %, 4.5 %, 6.75 % etc. of the water depth.

A similar argument holds when you want to resolve sharp density gradients in the vertical: you must have a sufficiently fine grid to resolve the vertical profile.

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Latitude	-90.0	90.0	0.0	degrees North
Orientation	0.0	360.0	0.0	degrees
Number of layers	1	100	1	-
Thickness	0.01	100.0	100.0	% total depth

Restrictions:

- ◊ The direction of the line segments in the polygon of the grid enclosure must form a multiple of a 45 degree angle with the numerical grid axis.
- ◊ A line segment may not intersect or touch another line segment.
- ◊ The grid enclosure (polygon) must be closed.
- ◊ The sum of the layer thickness must be equal to 100 %.
- ◊ The maximum number of layers is 100.



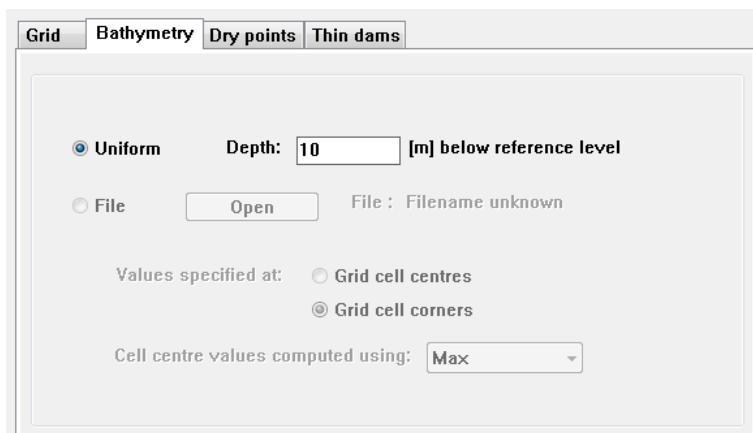


Figure 4.10: Sub-data group Domain → Bathymetry

4.5.2.2 Bathymetry

The depth schematisation may be uniform or non-uniform across the model area. A non-uniform (space-varying) bathymetry is given in an attribute file with the extension <dep>. For a uniform bathymetry the sub-window is given in [Figure 4.10](#).

You can either select:

Uniform

To enter a uniform depth value for the whole model area. Upon selecting *Uniform* an input field *Depth* is displayed.

Depth

To enter a uniform depth (positive downward).

Remark:

- ◊ The reference level of the depth is a horizontal plane. A negative value defines a depth *above* the reference plane.

File

To specify a file with bathymetry data. Upon selecting *File* an *Open* button is displayed.

Open

To open and read the bathymetry file with extension <dep>; see [Appendix A](#) for its file format.

Values specified at

Select if the depth values have been generated at cell centres (*Grid cell centres*) or at cell corners (*Grid cell corners*).

Remark:

- ◊ Default QUICKIN generates data at the grid cell corners, but you can also choose to generate data at cell centres.

You can either select (see also [subsection 10.8.1](#)):

- ◊ *Max* The depth at the cell centre is the maximum of the 4 surrounding depths at the cell corners.
- ◊ *Mean* The depth at the cell centre is the mean of the 4 surrounding depths at the cell corners.
- ◊ *Min* The depth at the cell centre is the minimum of the 4 surrounding depths at the cell corners.

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Depth	-1,000	20,000	10.0	m

The task of assigning depth values to grid points can be split in two main components. The

first one is the gathering of the raw bathymetric data, the second one is the actual interpolation of these raw data on the structured grid.

You can obtain the bathymetric data by:

- ◊ Digitising bathymetric charts (Admiralty Charts, Fair Sheets).
- ◊ Extracting the bottom schematisation of the area to be modelled from the bottom schematisation of an overall coarser hydrodynamic model.
- ◊ Using available measurements (echo-soundings).

Remark:

- ◊ Do pay special attention to the reference levels of different raw bathymetric data sources.



These options can be combined to obtain the most elaborate depth data. However, you must be careful when combining depth data originating from different sources (different chart datums). Corrections might be required to ensure that all depth values refer to only one reference level. The combined bathymetric data may not all be of the same resolution, neither of the same quality with respect to accuracy, nor may they cover the complete area of the grid. If all data are simply stacked into one file, there will be the problem that high quality data becomes contaminated with low quality data, thus spoiling interpolation results that might have been good if properly dealt with. Hence, you must carefully evaluate the quality of the various bathymetric data sets, before deciding to either include or discard it.

The interpolation of these data to the depth points of the grid should result in a bathymetry that resembles the natural bathymetry as closely as possible. However, this does not mean that the best bathymetry is obtained by always assigning the actual depths to all grid points. Since the FLOW module calculates averaged flow velocities and water levels, equality of averaged bathymetric features is more important than equality of bathymetric features at discrete grid points. Therefore, you should adopt a volume-preserving interpolation method that uses all data points if there is redundancy of data in a given grid cell.

In this way, the integral bathymetric features are best accounted for. In the opposite situation, when there are less data points than grid points in a given area, you will apply some kind of interpolation method on a triangulation network.

You can apply the utility program QUICKIN that enables you to select a sequence of data files and to control the interpolation areas and the interpolation method. The triangulation network is designed in such a way, that minimum triangle side lengths are achieved. Thus, a maximum correlation between the numerical bathymetry and known bathymetric data points is then obtained. The resulting bathymetry on the numerical grid is shown by way of iso-lines. You can correct interactively depth values of individual samples or grid points.

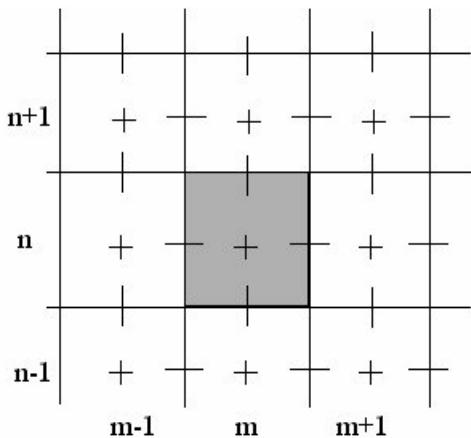


Figure 4.11: Dry point at grid location (m, n)

Bathymetry in relation to drying and flooding

Drying and flooding is a discontinuous process. It may generate high frequency disturbances in the results. In general, the magnitude of the disturbance will depend on the grid size and the bottom topography. The disturbances are small if the grid size is small and the bottom has smooth gradients. If the bottom has steep gradients across a large area on a tidal flat, a large area may be taken out of the flow domain in just one half integration time step. This can produce many (short wave length or wave period) disturbances (noise) in the simulation results. You can avoid this by smoothing the bottom gradients. You should also pay attention to the fact that depth values at points near closed boundaries are used in the drying and flooding procedure. Finally, you must ensure that the topography for points near tidal flats in general, and for discharge points near tidal flats in particular, are appropriately schematised to prevent the cells to be set wet and dry at each integration time step. For details of the numerical aspects of drying and flooding see [chapter 10](#). Finally, you must avoid drying and flooding at open boundaries.

4.5.2.3 Dry points

Dry points are grid cells centred around a water level point that are permanently dry during a computation, irrespective of the local water depth and without changing the water depth as seen from the wet points. Dry points are specified as a line of dry points; a single dry point is specified as a line of unit length.

In [Figure 4.11](#) a single dry point is defined at location (m, n) . The depth at the corner points, i.e. at (m, n) , $(m, n - 1)$, $(m - 1, n - 1)$ and $(m - 1, n)$ remain unchanged, i.e. as defined by the bathymetry. As a result the water depth in the surrounding water level points (+) are not influenced by the presence of the dry point.

You can specify dry points either manually, graphically in the **Visualisation Area** window or by reading from an attribute file (file extension <dry>).

Upon selecting the Data Group *Dry points* the sub-window of [Figure 4.12](#) is displayed.

Dry points are characterised by their (m, n) grid indices. You can apply one or more of the following options:

Add

To add one or a line of dry points either:

➤ Click Add.

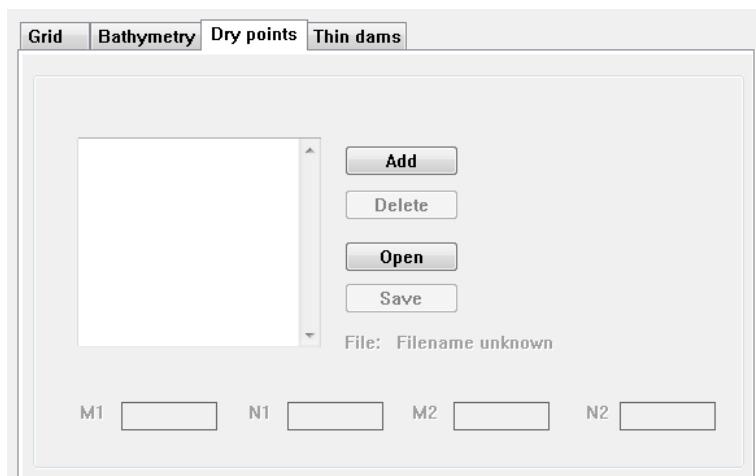


Figure 4.12: Sub-data group Dry points

- Specify the grid indices of the begin and end points, i.e. (m_1, n_1) and (m_2, n_2).

or

- Use the **Visualisation Area** window (see subsection 4.4.2 on how to use the **Visualisation Area** window).

Delete

To delete a single dry point or a line of dry points either:

- Select the dry points to be deleted in the list box.
- Click *Delete*.

or

- Use the **Visualisation Area** window.

Open

To read dry points from an attribute file with extension <dry>.

Remark:

- ◊ If you want to combine dry points read from a file and inserted manually you must read the file first and then add the manually defined dry points. In reverse order the manually defined dry points are overwritten by those of the file.



Save

To save all dry points in the same or a new attribute file.

Remark:

- ◊ You can save the dry points here or postpone it to the end of the input definition when you save the MDF-file.



Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Indices (m, n)	Anywhere in the computational domain		none	none



Restrictions:

- ◊ Dry points may only be specified along line segments which form a 45 degrees angle, or the multiple of it, with the computational grid axis.
- ◊ If dry points are read from file and defined manually, the file must be read first. Upon saving all dry points are stored in the (new) attribute file.

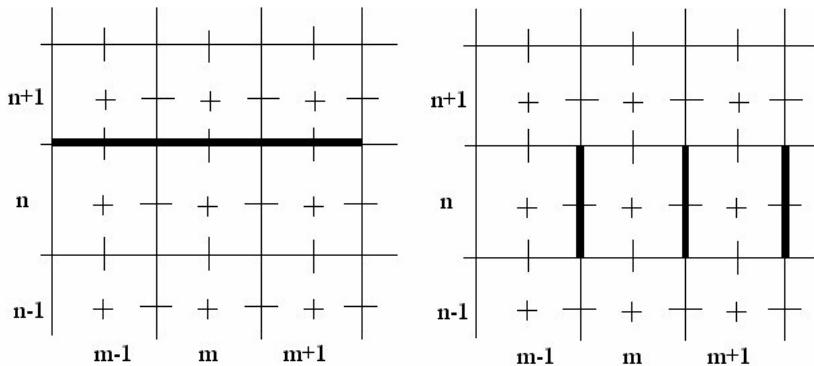


Figure 4.13: Equivalence of *v*-type thin dams (left) and *u*-type thin dams (right) with the same grid indices, ($M-1$ to $M+1$, N)

4.5.2.4 Thin dams

Thin dams are infinitely thin objects defined at the velocity points which prohibit flow exchange between the two adjacent computational cells without reducing the total wet surface and the volume of the model. The purpose of a thin dam is to represent small obstacles (e.g. breakwaters, dams) in the model which have sub-grid dimensions, but large enough to influence the local flow pattern. Thin dams are specified as a line of thin dams; a single thin dam is specified as a line of unit length. The line of thin dams is defined by its indices of begin and end point, (m_1, n_1) and (m_2, n_2) , respectively, and the direction of thin dam (*u*- or *v*-direction). Thin dams can be specified either manually or via an imported file with mask `<thd>`.

In Figure 4.13 the location is shown for three single *u*-thin dams (left) and a line of three *v*-thin dams (right). Note that these thin dams have the same grid indices; they only differ in the direction.



Remark:

- ◊ Thin dams separate the flow on both side, but they do not separate the bathymetry on both sides. Depth points are located at the thin dam and so this depth is used on both sides of the thin dam. If you need to apply a different depth on both sides you cannot apply a thin dam, but you should use a line of dry points instead.

Upon selecting *Thin dams* the sub-window of Figure 4.14 is displayed.

You can apply one or more of the following selections:

- | | |
|---------------|---|
| <i>Add</i> | To add a single or a line of thin dams either: |
| | <ul style="list-style-type: none"> ➤ Click <i>Add</i>. ➤ Specify the grid indices of the begin and end point, i.e. (m_1, n_1) and (m_2, n_2). |
| | or |
| | <ul style="list-style-type: none"> ➤ Use the Visualisation Area window, see subsection 4.4.2. |
| <i>Delete</i> | To delete a single or a line of thin dams either: |
| | <ul style="list-style-type: none"> ➤ Select the thin dam to be deleted in the list box. ➤ Click <i>Delete</i>. |
| | or |
| | <ul style="list-style-type: none"> ➤ Use the Visualisation Area window, see subsection 4.4.2. |
| <i>Open</i> | To read thin dams from an attribute file with extension <code><thd></code> . |

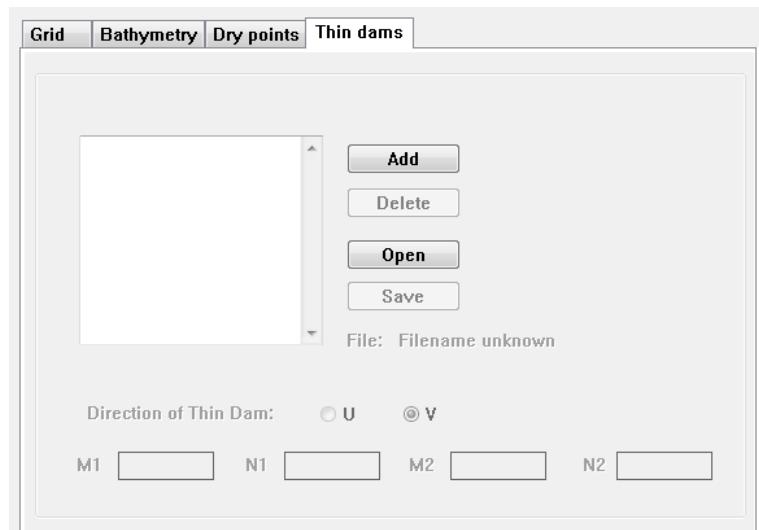


Figure 4.14: Sub-data group Thin dams

Remark:

- ◊ If you want to combine thin dams read from a file and inserted manually you must read the file first and then add the manually defined thin dams. In reverse order the manually defined thin dams are overwritten by those of the file.



Save

To save the thin dams to the same or a new attribute file.

Remark:

- ◊ You can save the thin dams here or postpone it to the end of the input definition when you save the MDF-file.



Direction of Thin Dam

A thin dam can either be defined as blocking the flow in *u*- or *v*-direction; select either direction.

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Indices (<i>m, n</i>)	Anywhere in the computational domain		none	none



Restriction:

- ◊ Thin dams can only be specified along lines parallel to one of the numerical grid axes or along lines which form a 45 degrees angle with the numerical grid axis.



Remarks:

- ◊ Defining thin dams at the grid boundaries (other than open boundaries) does not make sense since these boundaries are already closed per definition.
- ◊ Thin dams perpendicular to open boundaries are allowed.
- ◊ Thin dams aligned along an open boundary section are strongly discouraged. You can better subdivide such a section in sub-sections excluding the thin dams. If done so these thin dams do not make sense anymore, see the first remark.

Time frame

Reference date	23 03 2013 [dd mm yyyy]
Simulation start time	23 03 2013 00 00 00 [dd mm yyyy hh mm ss]
Simulation stop time	25 03 2013 00 00 00 [dd mm yyyy hh mm ss]
Time step	1.0 [min]
Local time zone (LTZ)	0 +GMT
GMT = Local time - LTZ	

Figure 4.15: Data Group Time frame

4.5.3 Time frame

In the Data Group *Time frame* you define the relation between the time axis of the real world and that of the simulation. All time dependent input is defined by the date and time as [dd mm yyyy hh mm ss], but in the simulation a time is determined by its number of time steps after the simulation reference date at time [00 00 00]. Upon selecting the Data Group *Time frame* the sub-window given in [Figure 4.15](#) is displayed.

You must specify the following input data:

Reference date

The reference date of the simulation.
The reference date defines the (arbitrary) $t = 0$ point for all time-series as used in the simulation. All time-series are specified in minutes after this $t = 0$ point. This reference date is also given in the header of files containing time-series, such as boundary conditions of type *Time-Series*, [subsection A.2.15](#), or flow rate and concentrations of discharges, [subsection A.2.22](#). By default the reference date is set equal to the current date.

Simulation start time

The start date and time of the simulation.

Simulation stop time

The stop date and time of the simulation.

Time step

The time step used in the simulation in minutes.

Generally, you can choose the time step based on accuracy arguments only, in most cases stability is not an issue. The accuracy is, among several other parameters, such as the reproduction of the important spatial length scales by the numerical grid, dependent on the Courant-Friedrichs-Lowy number (CFL), defined by:

$$CFL = \frac{\Delta t \sqrt{gH}}{\{\Delta x, \Delta y\}} \quad (4.1)$$

where Δt is the time step (in seconds), g is the acceleration of gravity, H is the (total) water depth, and $\{\Delta x, \Delta y\}$ is a characteristic value (in many cases the minimal value) of the grid spacing in either direction.

Generally, the Courant number should not exceed a value of ten, but for problems with rather small variations in both space and time the Courant number can be taken substantially larger. See [section 10.4](#) for a discussion on the numerical scheme used in Delft3D-FLOW in relation to stability and accuracy. For your convenience we repeat

Table 4.2: Time step limitations shallow water solver Delft3D-FLOW

Points per wave period T	$\Delta t \leq \frac{1}{40}T$
Accuracy ADI for barotropic mode for complex geometries	$C_t = 2\Delta t \sqrt{gH \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right)} < 4\sqrt{2}$
Stability baroclinic mode	$2\Delta t \sqrt{\frac{\Delta \rho}{\rho} gH \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right)} < 1$
Explicit algorithm flooding	$\frac{u\Delta t}{\Delta x} < 2$
Stability horizontal viscosity term	$\Delta t \nu_H \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) < 1$

the summary of that section. Let Δx and Δy be horizontal grid sizes. For a curvilinear grid $\Delta x = \sqrt{G_{\xi\xi}}$ and $\Delta y = \sqrt{G_{\eta\eta}}$. In the time step limitations are given for the shallow water code Delft3D-FLOW. Which of the limitations is most restrictive is dependent on the kind of application: length scale, velocity scale, with or without density-coupling etc.

For the Z-model the following additional limitations apply:

$$\Delta t \leq \frac{\min(\Delta x, \Delta y)}{\max(|u|, |v|)} \quad (4.2)$$

and

$$\Delta t \leq \frac{\min(\Delta x, \Delta y)}{\sqrt{\frac{\Delta \rho}{\rho} gH}} \quad (4.3)$$

Remark:

- ◊ You should check the influence of the time step on your results at least once.



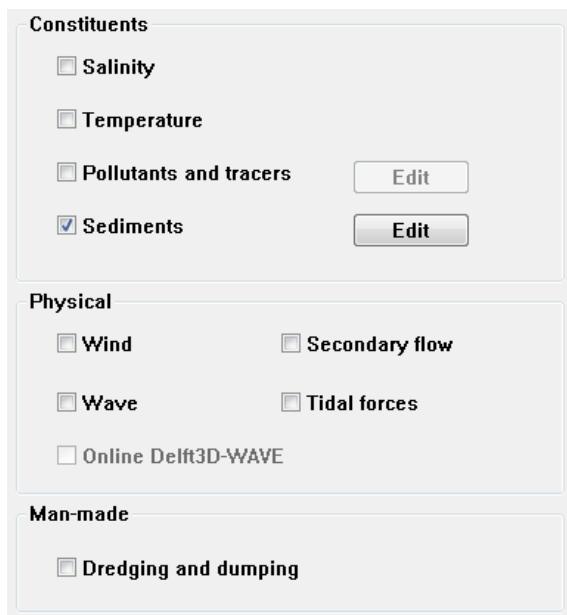
Local time zone

The time difference between local time and UTC.

The time zone is defined as the time difference (in hours) between the local time (normally used as the time frame for Delft3D-FLOW) and Coordinated universal time (UTC). The local Time Zone is used for two processes:

- ◊ To determine the phases in local time of the tidal components when tide generating forces are included in the simulation, see Data Group Processes.
- ◊ To compare the local time of the simulation with the times at which meteo input is specified, e.g. wind velocities and atmospheric pressure. These can be specified in a different time zone.

If the *Local Time Zone* = 0.0 then the simulation time frame will refer to UTC.

**Figure 4.16:** Data Group Processes**Domain:**

Parameter	Lower limit	Upper limit	Default	Unit
Date			Current date	none
Time	00 00 00	23 59 59	00 00 00	hours minutes seconds
Time step	0.0	999.0	1.0	minutes
Local Time Zone	-12.0	12.0	0.0	hours

**Restrictions:**

- ◊ If open boundaries are used with forcing type *Astronomic* then dates may not be before 1 January 1900.
- ◊ The start and stop date must be equal to or larger than the reference date.
- ◊ The start and stop time must be integer multiples of the time-step.
- ◊ The simulation stop time must be equal to or larger than the simulation start time.
- ◊ The time step must be positive.

4.5.4 Processes

In the Data Group *Processes* you specify which processes or quantities that might influence the hydrodynamic simulation are taken into account. Here you only define which processes you are going to apply; the parameters required for these processes are defined in other Data Groups, such as *Initial conditions* or *Boundaries*.

➤ Click the Data Group *Processes*, upon which [Figure 4.16](#) is displayed.

In [Figure 4.16](#) the process *Sediments* has been selected; this enables the process *Dredging and dumping*.

Remarks:

- ◊ To include *Waves* (offline) you need a communication file with wave data.
- ◊ To run FLOW Online with *WAVE* you have to write the communication file, and you have to prepare an MDW-file prior to execute these simulations
- ◊ *Secondary flow* is only displayed for depth averaged computations using the σ -model, but not for the Z -model
- ◊ *Tidal forces* is only displayed if the grid is in spherical co-ordinates.



You can select any combination of the processes by:

➤ Ticking off its check box and specifying names if required.

Salinity The salinity concentration will be taken into account including its influence on the water density and other processes.

Temperature The temperature is computed and the influence of the temperature on the water density is taken into account. Several heat flux models are available; see the sub-data group *Physical parameters* → *Heat flux model*.

Salinity and temperature affect the density of the water. As a consequence, a horizontal and vertical density gradient may occur which induces density driven flows. The density is calculated at cell centres.

Pollutants and tracers The spreading of up to 99 constituents (pollutants and/or sediments) can be simulated simultaneously. If *Pollutants* are selected a window appears to let you specify the names of the constituents applied. The names of the constituents can be up to 20 characters long.

Remark:

- ◊ You can specify a first order decay rate for each of the constituents. See [section B.2](#) for a description on how to use this option.



Sediments This functionality includes the transport of suspended sediments (cohesive and non-cohesive), bedloads and optionally updating the bathymetry. If *Sediments* are selected a window appears to let you specify the names of the cohesive and/or non-cohesive sediments applied, see [Figure 4.17](#).

The names of sediment substances have to start with "Sediment". Furthermore, the tabs *Sediment* and *Morphology* are available in the Data Group *Physical parameters* where you can define the sediment and morphology characteristics.

Remark:

- ◊ See [chapter 11](#) and [section B.9](#) for details on how to use the Sediment and morphology functionality.



Wind To include a wind field and its influence on the flow and spreading of

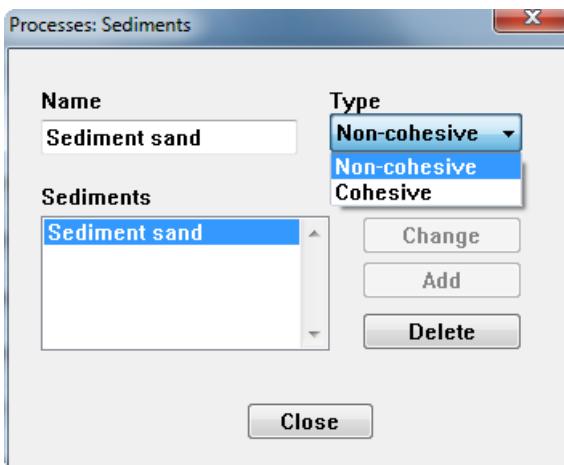


Figure 4.17: Sediment definition window

substances. If wind is selected a *Wind* tab is available in the Data Group *Physical parameters*. There you can specify a global time varying wind field, or read a time and space varying wind field and pressure from a file. The latter is useful for the simulation of storm surges or typhoons in a regional model.

Remark:

- ◊ If FLOW runs online with WAVE, the wind specified in FLOW can be used in WAVE.

Wave

To take into account the influence of short waves on the bed-stress and the radiation stress on the overall momentum transport. See [section 9.7](#) for details on how to take the wave-current interaction into account. For details of modelling waves you are referred to the User Manual of Delft3D-WAVE.

Remark:

- ◊ You first have to run the WAVE module to create a communication file with the required wave information.

Restriction:

- ◊ The Wave process is only available for the σ -model.

You select the formulation for the influence of short waves on the bottom roughness in the Data Group *Physical parameters* → *Roughness*.

Online Delft3D-WAVE

An extension to the previous process is Online Delft3D-WAVE. In this case there is a direct coupling with Delft3D-WAVE. Every time the communication file is written, subsequently a WAVE simulation is performed. Then FLOW resumes, using the latest WAVE results. Prior to starting the FLOW simulation with online Delft3D-WAVE you also have to prepare a WAVE input file. For details of modelling waves you are referred to the User Manual of Delft3D-WAVE.

Secondary flow

The process *Secondary flow* adds the influence of helical flow, such as in river bends, to the momentum transport, see [section 9.6](#) for details of the formulations.

The influence of spiralling flow can either be taken into account through an equilibrium formulation or as an evolutionary process by solving an advection-diffusion equation.

Remark:

- ◊ In 3D simulations the check box of *Secondary flow* is invisible, i.e. cannot (and need not) be selected. The influence of helical flow is resolved in the 3D formulation itself.

Restriction:

- ◊ *Secondary flow* is only available for the σ -model.

Tidal forces



To specify the parameters for tide generating forces. If tidal forces is selected a tab *Tidal forces* is available in Data Group *Physical parameters*. For details see [section 9.9](#). Tidal forces can only be specified for spherical co-ordinates.

Dredging and dumping

When the process *Sediments* is switched on, you can use the functionality to dredge and dump sediments. This feature can also be used for sand mining (only dredging, no associated dumping within the model domain) and sediment nourishment (only dumping, no associated dredging within the model domain). The file with dredging and dumping sites and dredge characteristics must be opened in the Data Group *Operations* → *Dredging and dumping*, see [subsubsection 4.5.9.2](#).

Restrictions:

- ◊ The process *Wave* is not available for the Z -model.
- ◊ *Secondary flow* is not available for the Z -model.
- ◊ *Tidal forces* can only be applied for spherical grids.



4.5.5 Initial conditions

In the Data Group *Initial conditions* you can specify the initial values the computation will start with.¹ Initial conditions are required for all dependent variables, such as water level, flow velocity components, salinity and/or temperature in the case of an inhomogeneous computation, and for the secondary flow and constituents if included in the simulation. The processes salinity, temperature, secondary flow and constituents are selected in the Data Group *Processes*.

Initial conditions can be very simple, such as a uniform value in the whole area, or more complex as a space varying value obtained from 'somewhere' or taken from a previous run.

You can select the initial conditions arbitrarily within the domain specified below. The optimum values depend on a combination of the values of the boundary conditions at the start time of the simulation, the values of the variables in the (dynamic) equilibrium of the model and to some extent the topography.

A large discrepancy between the initial condition and the boundary conditions at the simulation start time can result in short wave disturbances that propagate into the model area. If the discrepancy is so large that the flow velocity becomes close to or above the critical flow velocity the simulation might even become unstable. The time to reduce these short wave disturbances by internal dissipation such as bottom friction might be quite large, in tidal situations one or even several tidal cycles.

The effects of a discrepancy between the initial condition and the boundary conditions at the start time of the simulation can be substantially reduced by applying a transition period from the initial condition to the actual boundary conditions and determine the intermediate values by a smooth interpolation; see the Data Group *Numerical parameters*.

For complicated topographies with large drying and flooding areas we suggest to start at high water and define an initial condition to this situation. Mismatches between the initial condition and the boundary condition are washed out of the system instead of propagating inward.

¹Initial condition file is not yet available for the Z -model

Concentration [kg/m ³]		
Conservative spill	1	
Sediment sand	0	
Sediment mud	0	

Figure 4.18: Data group Initial conditions

If you do not use optimum initial conditions, the final results will be the same, but the transient period to reach the (dynamic) equilibrium will increase.

To display the *Initial conditions* window:

- Click the Data Group *Initial conditions*. Next, [Figure 4.18](#) is displayed.



Remark:

- ◊ For the layout to appear as displayed in [Figure 4.18](#) we have selected a depth averaged computation (*Number of layers* = 1 in the Data Group *Domain* → *Grid parameters*), selected all processes in the Data Group *Processes*, defined one pollutant, two sediments and set the option in [Figure 4.18](#) to *Uniform*. If you specify less processes less data fields will be displayed.

You can select one of the following options:

Uniform

To specify a uniform initial condition for the water level and each of the constituents defined in the Data Group *Processes*.



Remark:

- ◊ The uniform initial condition for the velocity components is by default set to zero.

For uniform initial conditions you must specify one or more of the following quantities, depending on the processes you have selected in the Data Group *Processes*:

Water level

Specify the initial condition above the reference plane (i.e. the still water level) in [m]. Above the reference plane is positive.

Salinity

Specify the initial (sea)water salinity in [ppt].

Temperature

Specify the initial water temperature in [°C].

Secondary flow

Specify the initial secondary flow in [m/s]. This option can only be applied when you have selected a depth averaged simulation (*Number of layers* = 1 in Data Group *Domain* → *Grid parameters*) and the σ -model.

Constituents

Specify the initial constituent (pollutant, sediment) concentrations in [kg/m³].

When you select either of the following options the input fields of the uniform values, displayed in the lower half of [Figure 4.18](#) are not displayed.

<i>Initial conditions file</i>	The initial conditions are read from an attribute file that you somehow obtained, for instance from a previous run, created by a separate program or constructed from measurements. Upon selecting this option you must specify the file.
<i>Restart file</i>	The initial conditions are read from a restart file from a previous run. To identify this file you must select either a <tri-rst.runid> or a <tri-rst.runid.date.time> file.
<i>Map file</i>	The initial conditions are read from a map file from a previous run. To identify this file you must select a <trim-runid> file.

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Water level	-1,000	1,000	0.0	m
Salinity	0.0	100.0	31.0	ppt
Temperature	0.0	60.0	15.0	°C
Secondary flow	-10.0	10.0	0.0	m/s
Constituent	0.0	1,000	0.0	kg/m ³

Restrictions:

- ◊ The uniform initial conditions for the flow velocity components in both horizontal directions are zero by default and cannot be changed.
- ◊ The initial conditions for salinity, temperature, secondary flow and constituents need only be specified if the corresponding process is selected in the Data Group *Processes*.
- ◊ For the *Restart file* and *Map file* options to work properly:
 - ◊ the restart-runid must differ from the current runid
 - ◊ the restart file <tri-rst.*> or map file <trim-*> must reside in the current working directory.
 - ◊ the simulation start time of the current scenario must coincide with the time instance at which the restart file is written in the selected scenario; see Data Group *Output → Storage*; or, when using a map-file: the start time of the current simulation must occur on the map file.
 - ◊ the same processes must be used in both computations.
 - ◊ in case of a 3D-application both computations must have the same number of layers and layer thickness.



Remark:



- ◊ Delft3D-FLOW does not check for all these requirements.

4.5.6 Boundaries

In the Data Group *Boundaries* you can define the open boundaries, their location, type and all input data related to driving the simulation. At an open boundary the flow and transport boundary conditions are required. These conditions represent the influence of the outer world, i.e. the area beyond the model area which is not modelled. The flow may be forced using water levels, currents, water level gradients, discharges (total or per grid cell) and the Riemann invariant which is a combination of water level and current. The hydrodynamic forcing can be prescribed using harmonic or astronomical components or as time-series. For water level forcing the boundary conditions can also be specified in terms of QH-relations. The transport of salinity, temperature and/or constituents is prescribed by specifying the inflow concentrations. You can specify the boundary conditions for these parameters as time-series only.

Depending upon the variability of the parameters along the open boundary, you can divide an open boundary into sections. A boundary section is characterised by a name, indices of the begin and end point of the boundary section, the type of boundary forcing and last but not least the data itself.

The following types of boundary conditions are available:

- ◊ Water level
- ◊ Velocity
- ◊ Neumann (water level gradient)
- ◊ Discharge or flux (total or per grid cell)
- ◊ Riemann or weakly reflective boundaries²

The choice of the type of boundary condition used depends on the phenomena to be studied. For instance, when you are modelling water levels at the inland side of an estuary, you will prescribe the known water levels at the entrance of the estuary. However, the same internal solution may be achieved by prescribing flow velocities, fluxes or weakly reflective conditions. But, the latter three yield a much weaker control over the water level, since velocities are only weakly coupled to water levels, especially for the more complex flow situations. Other examples are:

- ◊ When modelling river flow, the upstream flux is prescribed in combination with the downstream water level.
- ◊ When modelling cross flow in front of a harbour, velocity components are prescribed.

The Neumann type of boundary is used to impose the alongshore water level gradient. Neumann boundaries can only be applied on cross-shore boundaries in combination with a water level boundary at the seaward boundary, which is needed to make the solution of the mathematical boundary value problem well-posed.

The Riemann type of boundary is used to simulate a weakly reflective boundary. The main characteristic of a weakly reflective boundary condition is that the boundary up to a certain level is transparent for outgoing waves, such as short wave disturbances. Outgoing waves can cross the open boundary without being reflected back into the computational domain as happens for the other types of boundaries.

A weakly reflective form of the other boundary types is obtained by specifying a reflection coefficient. This coefficient Alfa must be ≥ 0 (recommended value for tidal computations is 50 or 100). For more details about the weakly reflective boundary condition see [Verboom and](#)

²This feature has only been tested for the σ -model

[Slob \(1984\)](#), [Verboom and Segal \(1986\)](#) and [chapter 9](#).

If there is more than one open boundary in the model area, you should *avoid* applying the same type of boundary condition at *all* boundaries. For instance, two velocity boundaries at both ends of a straight channel may lead to continuity problems, i.e. the channel will eventually dry up or will overflow, if the fluxes that are a result of these velocity components and their respective water levels are not compatible. In this case it is better to prescribe the normal velocity component at one end of the channel and the water level at the other. A physically stable result is then obtained. When modelling tidal flow in a large basin, forcing by prescribing water levels only is generally a sound procedure.

In practice, the type of boundary condition applied often depends mainly on the available data. For example, most of the larger sea models are likely driven by water level boundaries only since these are the only quantities known with some accuracy.

When prescribing water level boundaries, you should keep in mind that water level is a globally varying quantity. There is a substantial correlation between water levels in locations being not too far apart, meaning that a small error in the prescription of water levels can only be compensated by a large response in the velocity components. The area of influence of this phenomenon is not limited to a certain number of grid points near the boundary, but rather to the entire physical area. Since we do not want small errors in the boundary conditions to significantly influence the model results, we *locate boundaries as far away from the areas of interest as possible*.

The boundary condition is prescribed at two so-called boundary support points (Begin or A, and End or B), which divide a complete boundary into several segments. Points that lie in between these two support points are calculated by linear interpolation of the forcing at both ends. The actual signal at the support points may either be presented to the program as a harmonic type of signal (frequencies, amplitudes and phases) or directly as a time-series.

Often, the signal itself will be a tidal signal composed of amplitudes and phases at astronomic frequencies. For large area models (horizontal dimensions of the order of the tidal wave length or larger), the amplitudes and phases can be interpolated from nearby ports. However, the small relative error introduced by the interpolation procedure becomes more and more significant and important when the method is applied for smaller area models (e.g. coastal models). For these models, the boundary condition should be obtained by nesting the model into a calibrated larger area model.

Remark:

- ◊ The interpolation between the begin and end point of a boundary section are obtained by linear interpolation of the values at these (support) points. For boundary conditions expressed in amplitudes and phases the interpolation is executed on these quantities (and not on the computed time dependent boundary values).



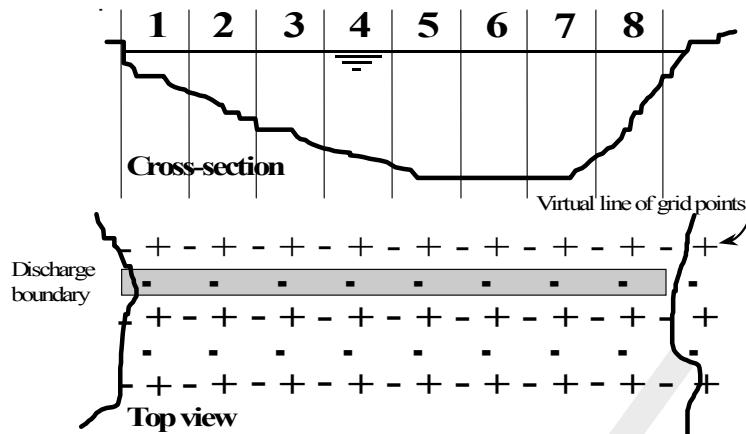


Figure 4.19: Sketch of cross-section with 8 grid cells

Discharge per grid cell and total discharge

To specify boundary conditions of type *Discharge per cell* you must apply a similar procedure as for the other boundary types, i.e. the discharge is prescribed at both ends of a boundary section and intermediate values are determined by linear interpolation. However, the discharge prescribed in an end point is the discharge through the grid cell related to that specific end point. If you want to prescribe a certain discharge through the total cross-section you must translate this total discharge into discharges at the end points. This is a rather difficult task if the bathymetry changes appreciably over the cross-section. A better approach is to define boundary sections of just one grid cell and to determine the boundary conditions per grid cell (still given at its end points A and B, which now are the same) using the following approach:

$$Q_i = \frac{H_i^{3/2}}{\sum_{j=1}^N H_j^{3/2}} Q \quad (4.4)$$

where Q_i , H_i , Q and N are the discharge through section i , the average depth in section i , the total discharge through the cross-section and the total number of sections in the cross-section, respectively.



Remark:

- ◊ The depth is defined at the grid lines between the velocity, or discharge points, see Figure 4.19 details.

In the present release of Delft3D-FLOW the procedure mentioned above is incorporated in the FLOW-GUI, so you can prescribe the discharge through a cross-section using the type of boundary *Total discharge*.

Upon selecting the Data Group *Boundaries* the following sub-window is displayed, see Figure 4.20. You can select one or more of the following options:

Add

To Add an open boundary section either:

- Click *Add*. The section is given the default name “-Unnamed-”, displayed in the *Section name* field.
- Replace the name “-Unnamed-” by a useful name.
- Specify the grid indices of the begin and end point given by (M1, N1) and (M2, N2).

or

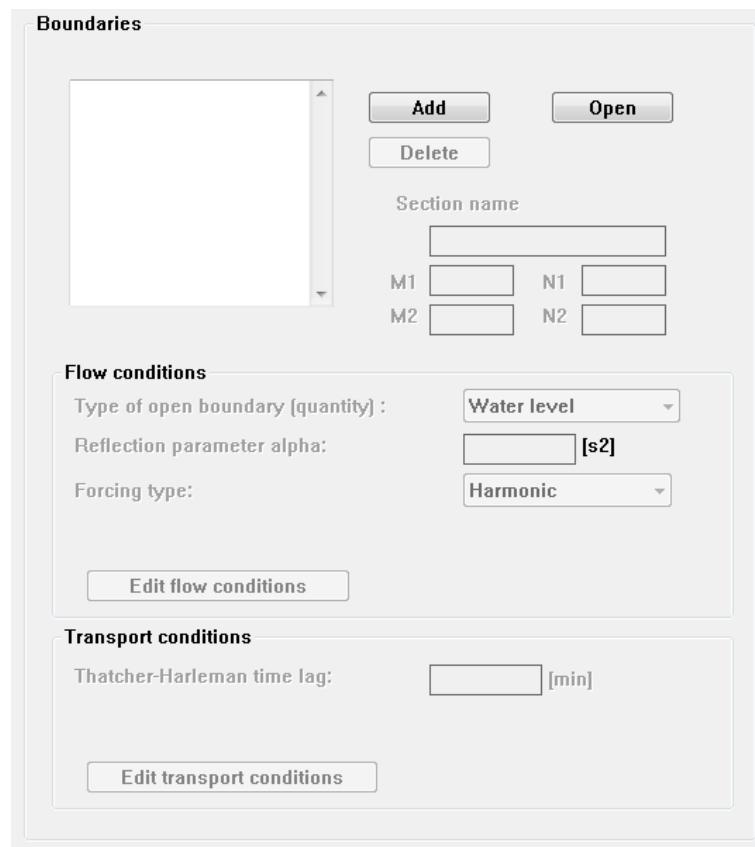


Figure 4.20: Main window for defining open boundaries

- Use the **Visualisation Area** window.

Remarks:



- ◊ Open boundary sections typically extend over more than just one grid cell, so you have to specify their begin and end point.
- ◊ There is no preferred direction in the definition of a boundary section.
- ◊ Only an open boundary of the type *Water level* may be defined along a diagonal grid line.

Delete

To *Delete* an open boundary section either:

- Select the boundary section to be deleted in the list box.
- Click *Delete*.

or

- Use the **Visualisation Area** window.

Open/Save

Open or Save the definitions (location, type of open boundary, reflection parameter, forcing type) of the open boundary sections in an attribute file with extension *bnd* in the working directory. Upon selecting *Open / Save* [Figure 4.21](#) is displayed.

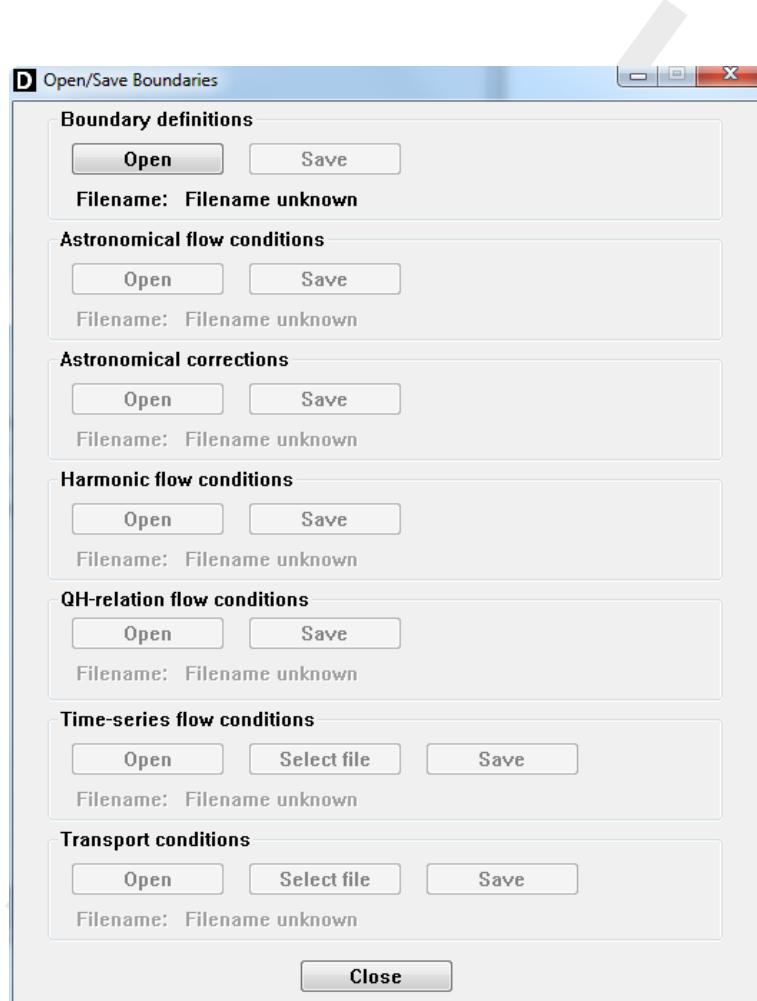


Figure 4.21: Open and save window for boundary locations and conditions

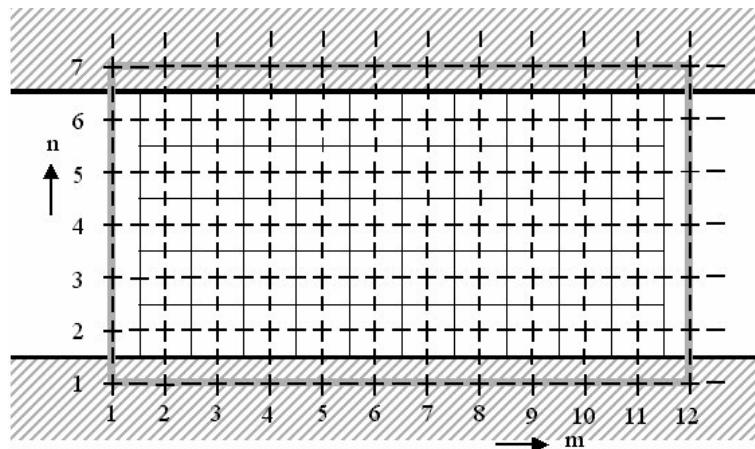


Figure 4.22: Straight channel; location of open and closed boundaries

Table 4.3: Definition of open and closed boundaries.

Boundary	Indices of begin and end point	
	Displayed in FLOW-GUI	Used in computation
West boundary	(1, 2) and (1, 6)	(1, 2) and (1, 6)
South boundary	(2, 1) and (11, 1)	(2, 1) and (11, 1)
East boundary	(12, 2) and (12, 6)	(12, 2) and (12, 6) ζ -boundary (11, 2) and (11, 6) u -boundary
North boundary	(2, 7) and (11, 7)	(2, 7) and (11, 7) ζ -boundary (2, 6) and (12, 6) v -boundary

Discussion:

The location and numbering of open boundaries need some clarification and illustration. Both open and closed boundaries are *defined* at the grid enclosure (through the water level points). However, physically closed and open velocity controlled open boundaries are located at the nearest velocity point inside the grid enclosure.

To illustrate this we take the example given in [Appendix E](#) for a straight rectilinear channel of $10 * 5$ grid cells or $11 * 6$ grid points, see [Figure 4.22](#).

The definition of the boundary sections, both as displayed in the FLOW-GUI and as used in the computation, are given in [Table 4.3](#).

Remark:

- ◊ Much confusion on the location and numbering of open and closed boundaries can be avoided by thinking in terms of computational grid cells. Water level boundaries are located just outside the first or last computational grid cell and velocity boundaries are located at the edge of the first or last computation cell. See [Appendix E](#) for a discussion and illustration.



Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Name boundary	Any printable character	-	-Unnamed-	none
(m, n)	At the grid enclosure	-	0	none
Alfa	0.0	10,000.0	0.0	none

**Restrictions:**

- ◊ A boundary section name must be non-blank, and can contain up to 20 characters.
- ◊ A boundary section must be located on the grid enclosure.
- ◊ The number of boundary sections is limited to 300.

Next you must specify for each boundary section the type of forcing and the forcing itself.

Type of open boundary Select either one of the options:

- ◊ *Water level*
- ◊ *Current*
- ◊ *Neumann*
- ◊ *Discharge per cell*
- ◊ *Total discharge*
- ◊ *Riemann*

**Remarks:**

- ◊ Riemann type of boundary has only been tested for the σ -model.
- ◊ Only an open boundary of the type *Water level* may be defined along a diagonal grid line.

Reflection parameter
Alfa

Specify the amount by which you want to make the open boundary less reflective for short wave disturbances that propagate towards the boundary from inside the model.

Remark:

- ◊ The reflection parameter does not apply for *Neumann* or *Riemann* controlled boundary sections.



Forcing type

Select either one of the options:

- ◊ *Astronomic*, the flow conditions are specified using tidal constituents, amplitudes and phases. The nodal amplitude and phase factors are determined at the start of the simulation and updated after each NodalT time interval.
- ◊ *Harmonic*, the flow conditions are specified using user-defined frequencies, amplitudes and phases.
- ◊ *QH-relation*, (only in combination with water level boundary type) the water level is derived from the computed discharge leaving the domain through the boundary.
- ◊ *Time-series*, the flow conditions are specified as time-series.

Vertical profile for hydrodynamics

The vertical profile does not apply for *Water level* or *Neumann* controlled boundary sections, neither for a 2D simulation. The profile can be:

- ◊ *Uniform* The velocity profile is uniform over the water depth.
- ◊ *Logarithmic* The velocity profile is a logarithmic function over the

water depth.

- ◊ *Per layer* The velocity profile is embedded in the boundary condition file taken from a larger area model by nesting.

Remark:

- ◊ This option is activated if you open a boundary location <bnd> file with boundary type '3-d profile'. In the current implementation you must add this type manually in the <bnd>-file; see [subsection A.2.10](#) and [A.2.15](#) for details.

Remark:

- ◊ The *Vertical profile for hydrodynamics* forms also part of the definition and it will be saved in the <bnd> file as well.

Edit flow conditions

To specify the boundary data for the hydrodynamic simulation. Details are specified in a separate window, see [subsubsection 4.5.6.1](#).

Edit transport conditions

To specify the boundary data for the constituents transported with the flow and only for those constituents specified in the Data Group *Processes*. Details are specified in a separate window, see [subsubsection 4.5.6.2](#).

To define the boundary condition data press the *Edit flow conditions* button. The sub-window displayed depends on the *Type of open boundary* selected.

4.5.6.1 Flow boundary conditions

For flow boundary conditions you can either select:

- ◊ Astronomic
- ◊ Harmonic
- ◊ QH-relation (for water levels only)
- ◊ Time-series

The window to specify the boundary data depends on the forcing type selected.

Astronomic boundary conditions

You can specify the boundary conditions in terms of astronomical components. Specifying astronomical components in a manner similar as harmonic components is a laborious activity, as tens of components can be prescribed. These astronomical components are mostly obtained from some type of tidal analysis program, such as Delft3D-TIDE or Delft3D-TRIANA. The boundary conditions are stored in an attribute file with extension <bca>.

The observed tidal motion can be described in terms of a series of simple harmonic constituent motions, each with its own characteristic frequency ω (angular velocity). The amplitudes A and phases G of the constituents vary with the positions where the tide is observed.

In this representation by means of the primary constituents, compound and higher harmonic constituents may have to be added. This is the case in shallow water areas for example, where advection, large amplitude to depth ratio, and bottom friction give rise to non-linear interactions. For a list of primary and compound constituents, see [Appendix C](#).

The general formula for the astronomical tide is:

$$H(t) = A_0 + \sum_{i=1}^k A_i F_i \cos(\omega_i t + (V_0 + u)_i - G_i) \quad (4.5)$$

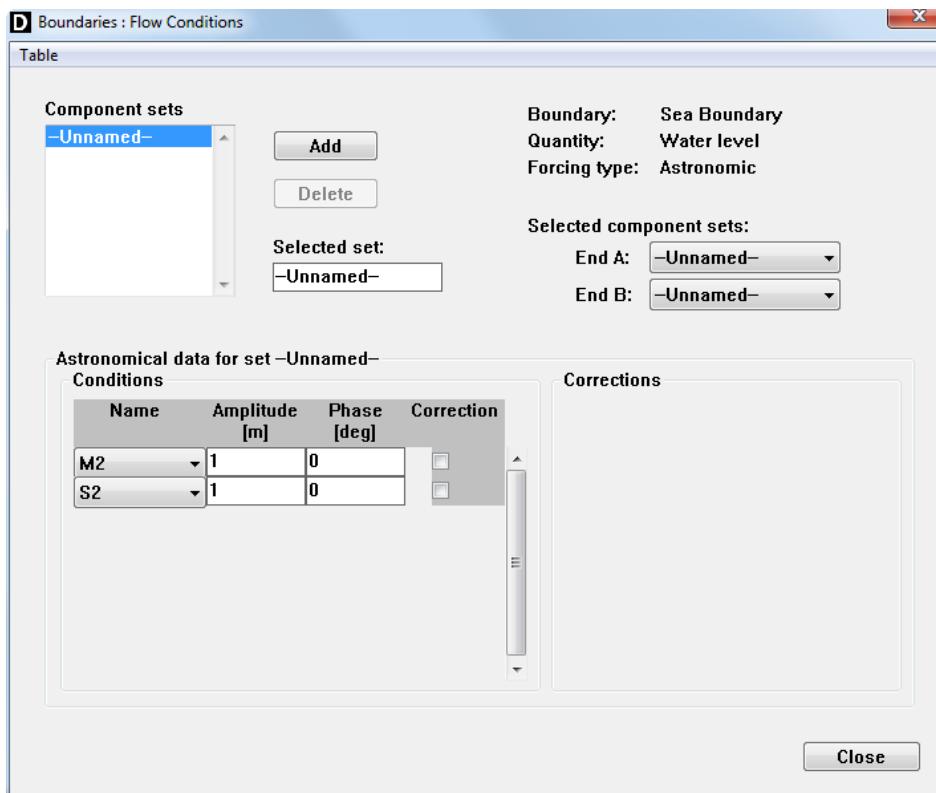


Figure 4.23: Specifying astronomical boundary conditions

in which:

$H(t)$	water level at time t
A_0	mean water level over a certain period
k	number of relevant constituents
i	index of a constituent
A_i	local tidal amplitude of a constituent
F_i	nodal amplitude factor
ω_i	angular velocity
$(V_0 + u)_i$	astronomical argument
G_i	improved kappa number (= local phase lag)

F_i and $(V_0 + u)_i$ are *time-dependent* factors which, together with ω , can easily be calculated and are generally tabulated in the various tidal year books. V_0 is the phase correction factor which relates the local time frame of the observations to an internationally agreed celestial time frame. V_0 is frequency dependent. F_i and u_i are slowly varying amplitude and phase corrections and are also frequency dependent. For most frequencies they have a cyclic period of 18.6 years. A_0 , A_i and G_i are *position-dependent*: they represent the local character of the tide. You only have to provide the amplitudes, phases and frequencies (in terms of constituent names). By default, Delft3D-FLOW re-calculates the nodal amplitude factors and astronomical arguments every 6 hours. See [section B.21](#) for changing this default.

Upon selecting *Astromonic* and next *Edit flow conditions* Figure 4.23 is displayed.

Each open boundary section has a begin (A) and an end (B) point. At each end you have to assign a *Component set*. A *Component set* consists of a number of tidal constituents with their *Name*, *Amplitude* and *Phase*.

To add a *Component set*:

- Click *Add*. The set is given the default name ‘-Unnamed-’.
- Replace the name ‘-Unnamed-’ in the *Selected set* field by a useful name, reflecting the boundary and the end point (A or B) for which this set will be used.

Remark:



- ◊ When adding a *Component set*, this set will have the same tidal constituents as the first set. The amplitudes will be 1 and the phases 0.

To add a tidal constituent to a *Component set*:

- First select the required set from the list of available *Component sets*.
- Click in the *Amplitude* or *Phase* field.
- Use the menu item *Table* in the menu bar to insert, copy or delete a row, or to copy a value.
- In the *Name* field: select from the dropdown list the required tidal constituent.
- Specify the *Amplitude* and *Phase*.

Remark:



- ◊ The mean value (A0) has no phase.

To assign a *Component set* to a boundary:

- First select the required open boundary.
- Click *Edit flow conditions*.
- Under *Selected component sets*, select from the dropdown list the required set for *End A*.
- Under *Selected component sets*, select from the dropdown list the required set for *End B*.

You can apply corrections to the defined astronomic components. These corrections can be applied to a sub-set of the boundary sections and to a sub-set of the astronomic components applied. These corrections are typically applied during the calibration phase of a model. Both the amplitude (multiplicative) and the phase (additive) can be corrected, see [Figure 4.24](#). This attribute file has extension <cor>.

Remarks:



- ◊ The correction does not apply for the A0 astronomic component.
- ◊ If the type of open boundary is *Total discharge*, only one *Component set* is needed for this open boundary. Specify the same set for End A and End B.
- ◊ Be aware that amplitudes and phases for intermediate points at the open boundary are linearly interpolated from the values at the end points. For instance, if you have a phase of 1 degree at End A, and a phase of 359 degrees at End B, an intermediate point will have a phase of 180 degrees. If you do not want this, you can specify 361 degrees at End A.

To save the component sets, the corrections and the specification of the sets to boundary end points:

- ◊ Use *Open / Save* in [Figure 4.20](#) to enter the **Open/Save Boundaries** window of [Figure 4.21](#).

Remarks:



- ◊ The components sets are stored in a file with extension <bca>.
- ◊ The information about which set is assigned to which boundary end point is stored in the <bnd> file.

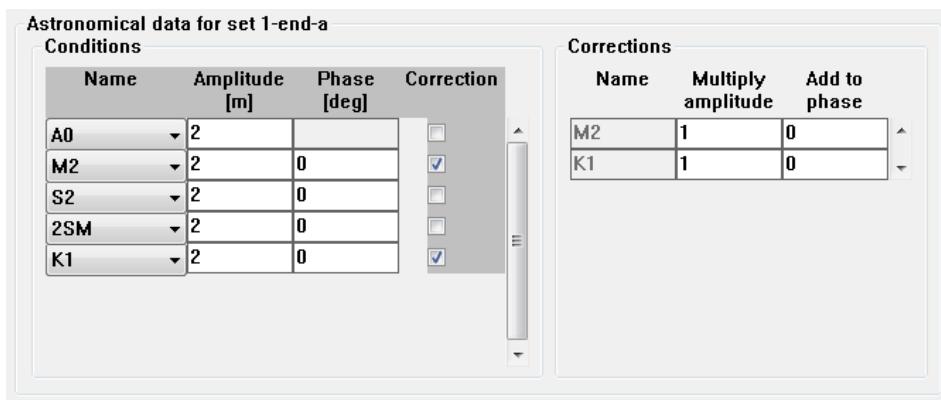


Figure 4.24: Contents of a Component set with two tidal constituents having corrections

You can import astronomic boundary conditions by selecting *Open/Save* in [Figure 4.20](#) and *Open Astronomical flow conditions* in [Figure 4.21](#). In the window of [Figure 4.23](#) the component sets from the imported file are listed under *Component sets*. You can extend, delete or modify them at will.

To assign a Component set to an end point of an open boundary section, follow the procedure above.

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Water level	0.0	100.0	0.0	m
Neumann	0.0	100.0	0.0	-
Current	0.0	10.0	0.0	m/s
Discharge	0.0	1.0E+06	0.0	m ³ /s
Riemann	0.0	100.0	0.0	m/s
Phase	-360.0	360.0	0.0	degrees
Alpha	0.0	10,000	0.0	none



Restrictions:

- ◊ Astronomical forcing can not be mixed with harmonic forcing.
- ◊ Boundary conditions of type *Astronomic* must precede boundary conditions of type *QH-table* and *Time-series*.
- ◊ The maximum number of frequencies is 234.

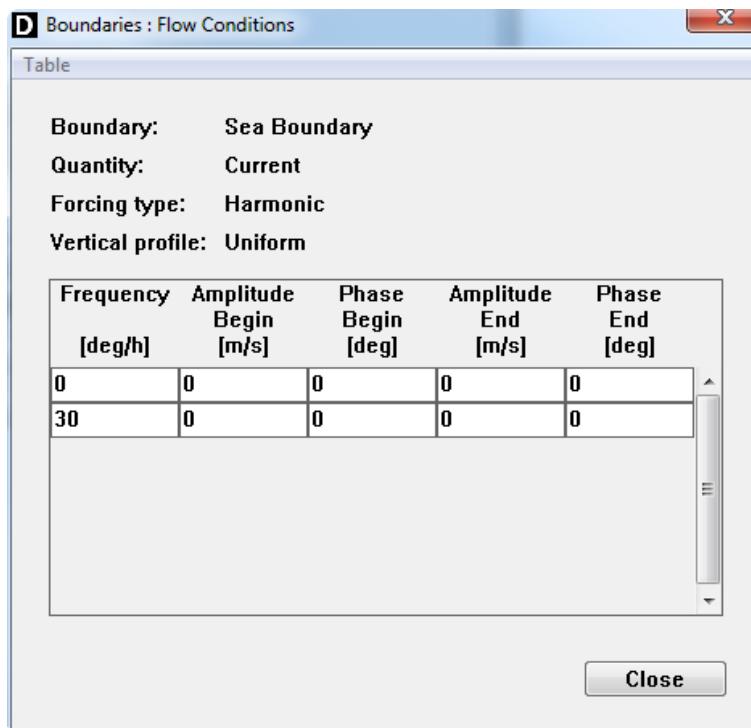


Figure 4.25: Specifying harmonic boundary conditions

Harmonic boundary conditions

Upon selecting *Harmonic* and next *Edit flow conditions* Figure 4.25 is displayed.

Remark:

- ◊ If the type of open boundary is *Total discharge*, the amplitudes and phases for the whole section are required.



The boundary signal $F(t)$ is constructed by super-imposing the following individual components:

$$F(t) = \sum_{i=1}^N A_i \cos(\omega_i t - \varphi) \quad (4.6)$$

where N is the number of frequency components. The unit of the amplitudes depends on the quantity prescribed, they are respectively [m] for water elevations, [-] for Neumann, [m/s] for velocities, [m^3/s] for fluxes, and [m/s] for Riemann boundaries.

For each of the boundary sections you can either import the boundary conditions by selecting *Open/Save*, or specify the harmonic conditions manually in a table.

Open

Import the harmonic boundary conditions from an attribute file with extension <bch>.



Remark:

- ◊ The data you import here will be included in the <bch> attribute file when you save the boundary locations and conditions, see Figure 4.21.

In the table you must specify:

Frequency

The frequency of the harmonic component in [degrees/hour].

Amplitude The amplitude of the harmonic component in [m], [-], [m^3/s] or [m/s] at the *Begin* point A and at the *End* point B defined at grid indices (M1, N1) and (M2, N2), respectively.

Phase The phase of the harmonic component in [degrees] at the *Begin* point A and at the *End* point B defined at grid indices (M1, N1) and (M2, N2), respectively.

Use the menu item *Table* in the menu bar to insert, copy or delete a row, or to copy a value.



Remarks:

- ◊ The mean value is specified by zero frequency and zero phase. The mean value must *always* be specified.
- ◊ All boundaries with *Harmonic* forcing must have the same frequencies. If not taken care of by you, the GUI will save the frequencies of the last *Harmonic* boundary.
- ◊ Be aware that amplitudes and phases for intermediate points at the open boundary are linearly interpolated from the values at the end points. For instance, if you have a phase of 1 degree at End A, and a phase of 359 degrees at End B, an intermediate point will have a phase of 180 degrees. If you do not want this, you can specify 361 degrees at End A.



Restrictions:

- ◊ Harmonic forcing may not be mixed with astronomical forcing and must precede QH and time-series forcing.
- ◊ The maximum number of frequencies is 234.

If you have defined all data for the currently selected boundary section:

- Close the sub-window and select another boundary section to define its boundary conditions.

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Water level	0.0	100.0	0.0	m
Neumann	0.0	100.0	0.0	-
Current	0.0	10.0	0.0	m/s
Discharge	0.0	1.0E+06	0.0	m^3/s
Riemann	0.0	100.0	0.0	m/s
Phase	-360.0	360.0	0.0	degrees

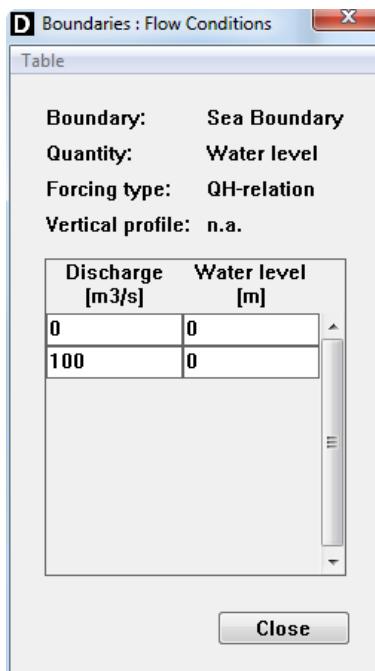


Figure 4.26: Specifying QH-relation boundary conditions

QH-relation boundary conditions (for water level boundaries only)

For QH-relations you must specify the relation between the outflowing discharge and the water level at the boundary. Intermediate values are determined by linear interpolation. You cannot change the type of interpolation. Above the highest specified discharge and below the lowest specified discharge the water level is kept constant.

Upon selecting *QH-relation* and next *Edit flow conditions* Figure 4.26 is displayed.

The sub-window displayed has just one column for the discharge and one column for the water level. Use the menu item *Table* in the menu bar to insert, copy or delete a row, or to copy a value. First click in either the Discharge or Water level input field.

If you have defined all data for the currently selected boundary section:

- Close the sub-window and select another boundary section to define its boundary conditions.

A relaxation parameter can be specified in the data group *Numerical parameters*.

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Discharge	-1.0E+6	1.0E+06	0.0	m ³ /s
Water level	-100.0	100.0	0.0	m

Restrictions:

- ◊ The discharges should be specified in increasing order. That is, for positive discharges from small to large and for negative discharges from large to small.



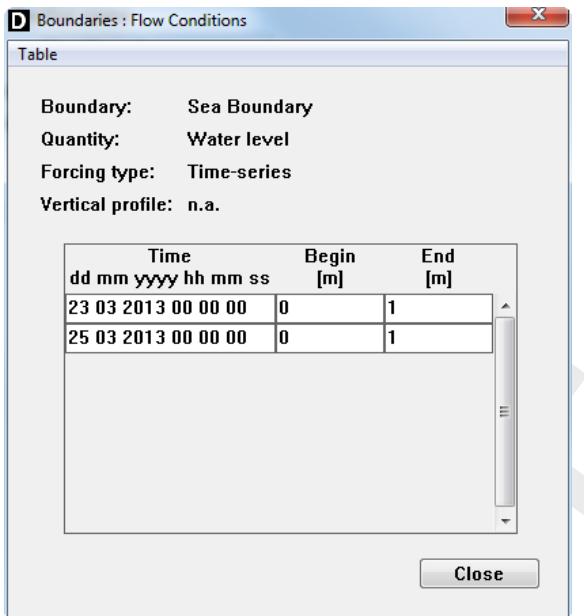


Figure 4.27: Specifying time-series boundary conditions

- ◊ Intermediate values are determined by linear interpolation. Above the highest specified discharge and below the lowest specified discharge the water level is kept constant.
- ◊ Boundaries with forcing by a *QH relation* must precede time-series boundaries and they must follow boundaries with harmonic and astronomical forcing.

Time-series boundary conditions

Upon selecting *Time-series* and next *Edit flow conditions* Figure 4.27 is displayed.



Remark:

- ◊ If the type of open boundary is *Total discharge*, only one (1) discharge is required for the whole section.

For time-series conditions you must specify the boundary conditions in terms of values of the selected kind, i.e. water level, Neumann, current, discharge or Riemann, at time breakpoints for both the *Begin* point A and the *End* point B, respectively. Intermediate values are determined by **linear** interpolation. You cannot change the type of interpolation.

Use the menu item *Table* in the menu bar to insert, copy or delete a row, or to copy a value.

If you have defined all data for the currently selected boundary section:

- Close the sub-window and select another boundary section to define its boundary conditions.



Figure 4.28: Transport conditions; Thatcher Harleman time lags

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Water level	-100.0	100.0	0.0	m
Neumann	-100.0	100.0	0.0	-
Current	-10.0	10.0	0.0	m/s
Discharge	-1.0E+6	1.0E+06	0.0	m ³ /s
Riemann	-100.0	100.0	0.0	m/s
Alfa	0.0	10,000	0.0	none

Restrictions:

- ◊ The time breakpoints must differ from the simulation start time by an integer multiple of the time-step (Data Group *Time frame*).
- ◊ The first time breakpoint must be before or equal to the simulation start time, and the last time breakpoint must be equal or later than the simulation stop time (see Data Group *Time frame*).
- ◊ Time breakpoints must be given in ascending order.
- ◊ Intermediate values are determined by linear interpolation.
- ◊ Time-series forcing *must follow* astronomic, harmonic or QH forcing when these type of boundary conditions are applied in the same model area.



4.5.6.2 Transport boundary conditions

To specify the boundary conditions for all quantities defined in the Data Group *Processes* you must for each boundary section:

- ◊ Select the boundary section in the list box, see [Figure 4.20](#).
- ◊ Specify if required *Thatcher Harleman time lags*, see [Figure 4.28](#).

Thatcher-Harleman time lag

The return time for concentrations from their value at outflow to their value specified by the boundary condition at inflow. For 3D simulations you can specify different values at the surface and at the bottom to represent the influence of stratification and the different flow conditions above and below the interface.

At the sea-side boundary a common problem for numerical models of estuarine areas is encountered when the boundary conditions for a constituent are to be prescribed. In a physical (unbounded) world, the inflowing water mass immediately after low water slack originates from the outflowing water mass a moment earlier. Consequently, the concentration of the inflowing

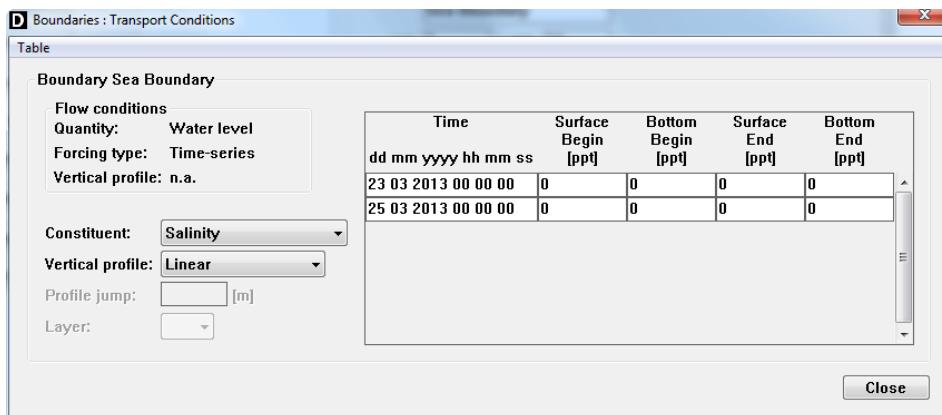


Figure 4.29: Specifying transport boundary conditions

water is commonly not equal to the concentration C_{\max} which has been prescribed along this open boundary. It will take some time before the concentration along this open boundary reaches the C_{\max} value. In numerical models, this time lag (return time) is often modelled by means of a “Thatcher-Harleman” boundary condition (Thatcher and Harleman, 1972). The return time depends on the flow conditions outside the estuary. If there is a strong circulation the return time is short. The return time must be specified for each open boundary section, one for the top layer and the other for the bottom layer. The return times for the layers in between will be determined through a linear interpolation of these two values. In a 2D model only the first value is used, the value of the upper layer.



Remark:

- ◊ When you specify a negative value for the time lag the boundary treatment for constituents, salinity and temperature will be the same as at outflow, but now for all times. Thus, the solution for these parameters will be fully determined by the processes inside the model area. This negative value has *no* physical meaning or background; it only offers the option to fully exclude the influence of the outer world on the inner solution.
- Select *Edit transport conditions*.

Next the window **Boundaries: Transport Conditions** is displayed, see [Figure 4.29](#). The window displayed is for a 3D simulation; for a 2D simulation all input fields related to 3D are either absent or insensitive. In a 3D simulation you can specify a surface and a bottom value, in a 2D simulation you need to specify only the surface value (applied over the whole depth).

Transport boundary conditions *always* are specified as time-series.

You can specify one or more of the following options, depending whether you have a 2D or a 3D simulation; for each of the quantities selected in the Data Group *Processes* you must:

Constituent

Select the quantity to be defined next in the table.

In the table you must specify time breakpoints and concentrations at the *Surface* and *Bottom* layer for the *Begin* point A and the *End* point B of the current boundary section, defined at grid indices $(m1, n1)$ and $(m2, n2)$, respectively. Use the menu item *Table* in the menu bar to insert, copy or delete a row, or to copy a value.

Vertical profile

To specify the vertical profile of the concentration at the current boundary section. You can either select:

- ◊ *Uniform* The value is applied over the whole water depth; only

- the surface values need to be specified.
- ◊ *Linear* Surface and bottom values are linearly interpolated for intermediate depth.
 - ◊ *Step* You can define a jump in the vertical profile. Upon selecting this option you can specify the depth (*Profile jump*) below the free surface at which the profile jump occurs. The surface value is maintained down to this depth and the bottom value is used from the jump depth down to the bottom.
 - ◊ *Per layer* The vertical profile is embedded in the boundary condition file taken from a larger area model by nesting.

If a constituent (i.e. salinity or temperature, etc) is included in a 3D model then the vertical profiles for these quantities must be prescribed, besides the usual boundary condition values. The options for the vertical profile for these quantities are respectively: *Uniform*, *Linear* and *Step* (discontinuous profile). In the last case the location of the profile jump, measured in metres from the water surface, must be prescribed. During initialisation the jump in the profile will be translated into a layer number. *Once determined, this layer number will be fixed throughout the simulation; so this layer number depends on the initial water level.*

Remark:



- ◊ When the initial conditions are read from a restart file (hot start), the position of the jump will be determined using the water elevation read from this file. Consequently, a mismatch between your boundary condition and the concentration distribution in the vertical, just inside the open boundary, may occur. This mismatch can produce spurious oscillations in your result.

If you have defined all data for the currently selected boundary section:

- Close the sub-window and select another boundary section to define its transport boundary conditions.

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Salinity	0.0	100.0	0.0	ppt
Temperature	0.0	60.0	0.0	°C
Pollutant or sediment	0.0	1,000	0.0	kg/m ³

Restrictions:



- ◊ The time breakpoints must differ from the simulation start time by an integer multiple of the time-step (Data Group *Time frame*).
- ◊ The first time breakpoint must be before or equal to the simulation start time, and the last time breakpoint must be equal or later than the simulation stop time (see Data Group *Time frame*).
- ◊ Time breakpoints must be given in ascending order.

Hydrodynamic constants			
Gravity	9.81	[m/s ²]	
Water density	1000	[kg/m ³]	
Air density	1	[kg/m ³]	
Temperature	15	[°C]	
Beta_c	0.5	H	
Equilibrium state	<input type="checkbox"/>		

Wind drag coefficients			
Breakpoints	Coefficient	Wind speed	
A	0.00063	H	0 [m/s]
B	0.00723	H	100 [m/s]
C	0.00723	H	100 [m/s]

Figure 4.30: Specifying the physical constants

4.5.7 Physical parameters

In the Data Group *Physical parameters* you can select or specify a number of parameters related to the physical condition of the model area. The physical parameters can be split into several classes; some are always needed and some are related to the processes switched on or off.

The Data Group *Physical parameters* contains several tabs. The tabs for *Constants*, *Roughness* and *Viscosity* are always visible. The other tabs are only available if the associated processes are switched on:

- ◊ *Heat flux model* (requires *Temperature* process activated)
- ◊ *Sediment* and *Morphology* (requires *Sediments* process activated)
- ◊ *Wind* (requires *Wind* process activated)
- ◊ *Tidal forces* (requires grid in spherical co-ordinates and *Tidal forces* process activated)

4.5.7.1 Constants

Upon selecting *Physical parameters* → *Constants* the sub-window given by Figure 4.30 is displayed. You can specify one or more of the following constants:

<i>Gravity</i>	The gravitational acceleration in [m/s ²].
<i>Water density</i>	The background water density in [kg/m ³].
	This value is only required for a homogeneous simulation, i.e. the processes temperature and salinity are not selected.
<i>Air density</i>	The density of air in [kg/m ³], used in the wind stress formulation.
<i>Temperature</i>	The background water temperature in [°C], used in the equation of state. This value is only required if the process <i>Temperature</i> is not selected, and the process <i>Salinity</i> is.
<i>Salinity</i>	The background water salinity in [ppt], used in the equation of state. This value is only required if the processes <i>Salinity</i> is not selected, and the process <i>Temperature</i> is.
<i>Beta_c</i>	Fraction of the shear stress due to secondary flow taken into account

Equilibrium state

in the momentum equation.

Tick off if you want to apply the equilibrium state for the secondary flow process (only for depth-averaged flow):

- ◊ Yes Apply the equilibrium state.
- ◊ No Compute the secondary flow as a dynamical process.

The flow in a river bend is basically three-dimensional. The velocity has a component in the plane perpendicular to the river axis. This component is directed to the inner bend along the river bed and directed to the outer bend near the water surface. This so-called ‘secondary flow’ (spiral motion) is of importance for the calculation of changes of the river bed in morphological models. This spiral motion is automatically taken into account in a 3D simulation, but in 2D simulations the flow equations must be extended to take this effect into account. The intensity of the secondary flow is a measure for the magnitude of the component of the secondary flow normal to the depth averaged flow. The vertical distribution of this horizontal velocity component is assumed to be a universal function of the vertical co-ordinate. The actual local velocity distribution originates from a multiplication of this function with the spiral motion intensity. The spiral motion intensity, I , can be used to determine the deviation of the direction of the bed shear stress from the direction of the depth-averaged flow; this is done in the morphological module if the secondary flow is included as a process. The β_c parameter described here influences the extent to which the effect of spiral motion in a depth-averaged model is accounted for. The secondary flow may be computed from a local equilibrium approach or from an advection/diffusion equation. The parameter $Beta_c$ specifies to which extent the shear stresses due to the secondary flow are included in the momentum equation ($Beta_c = 0$ means no inclusion). See [section 9.6](#) for full details.

Remark:

- ◊ You do not need (can not) prescribe boundary conditions for the transport equations that describe the secondary flow. The underlying assumption is that the secondary flow is locally determined inside the model area and not transported into the model area through the open boundaries. This assumption might be violated in models that cover only a very small area.

Wind drag

You can specify three *Wind drag coefficient* values and three *Wind speed* values at which these coefficients apply.

The wind drag coefficient may be linearly dependent on the wind speed, reflecting increasing roughness of the water surface with increasing wind speed. The three wind drag coefficients determine three breakpoints in the piece-wise linear function of wind drag coefficient and wind speed, see [Figure 4.31](#). The first two coefficients, *Breakpoint (A)*, determine the constant wind drag coefficient from zero wind speed up to the wind speed specified. The last two coefficients, *Breakpoint (C)*, specify the constant wind drag coefficient from the specified wind speed and higher. Inbetween the breakpoints there is a linear interpolation applied. As a result, you can specify a constant wind drag coefficient, a linearly increasing wind drag coefficient or a piece-wise linear function of wind speed, as shown in [Figure 4.31](#).

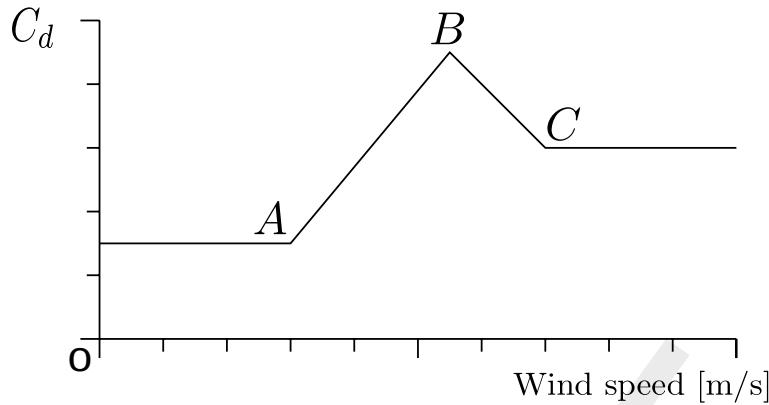


Figure 4.31: Examples of the wind drag coefficient

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Gravity	9.5	12.0	9.81	m/s ²
Water density	900.0	1,500	1,000	kg/m ³
Air density	0.5	1.5	1.0	kg/m ³
Temperature	0.0	60.0	15.0	°C
Salinity	0.0	100.0	31.0	ppt
Beta_c	0.0	1.0	0.5	none
Coefficient A	0.0	0.1	0.00063	none
Coefficient B	0.0	0.1	0.00723	none
Coefficient C	0.0	0.1	0.00723	none
Wind speed A	0.0	100.0	0.0	m/s
Wind speed B	0.0	100.0	100.0	m/s
Wind speed C	0.0	100.0	100.0	m/s

Roughness

In the sub-data group *Roughness* you can specify the bottom roughness and the roughness of the side walls.

The bottom roughness can be computed with several formulae and you can specify different coefficients for x - (or ξ -) and y - (or η -) direction; either as a uniform value in each direction, or space-varying imported from an attribute file.

Upon selecting *Physical parameters* → *Roughness* the window given by [Figure 4.32](#) is displayed.

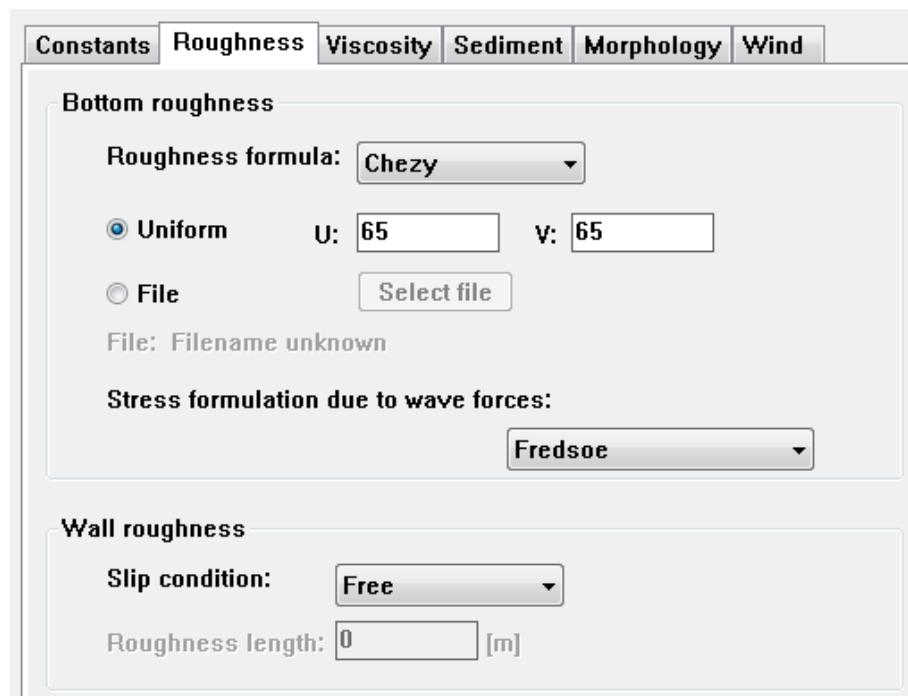


Figure 4.32: Sub-data group Roughness

Bottom roughness

You can select and define the following options and parameters:

Roughness formula The bottom roughness can be computed according to:

- ◊ Manning.
- ◊ White-Colebrook.
- ◊ Chézy.
- ◊ Z_0 (for 3D simulations only).

In the Manning formulation the Manning coefficient, n , must be specified. The Chézy friction coefficient is calculated from:

$$C = \frac{H^{1/6}}{n} \quad (4.7)$$

where H is the water depth. A typical Manning value is $0.02 [\text{s}/\text{m}^{1/3}]$. In the White-Colebrook formulation you must specify the equivalent geometrical roughness of Nikuradse, k_s [m]. The Chézy friction coefficient is calculated from:

$$C = 18 \cdot 10 \log \left(\frac{12H}{k_s} \right) \quad (4.8)$$

A first estimate of the bed roughness length z_0 [m] in 3D computations can be derived from the equivalent geometrical roughness of Nikuradse, k_s :

$$z_0 = \frac{k_s}{30} \quad (4.9)$$

Typical values of k_s range from $0.15 [\text{m}]$ for (river) beds with sediment transport down to $0.01 [\text{m}]$ or less for (very) smooth surfaces; see subsection 9.4.1 for more details.

A first estimate of the Chézy friction coefficient is given by:

$$C = 25 + H \quad (4.10)$$



Uniform File

Stress formulation due to wave forces

Remark:

- ◊ The Chézy friction coefficient discussed above applies to a depth averaged computation.

Specify the values in u - and v -direction in the input fields.

Select the space varying coefficients in u - and v -direction from an attribute file (extension <rgh>). See [subsection A.2.19](#) for a format description of the roughness file. The file is not read.

The bottom stress due to wave forces can be computed with several formulae. You can select one of the following formulae, see also [subsection 9.7.5](#):

- ◊ Fredsøe (1984)
- ◊ Myrhaug and Slaattelid (1990)
- ◊ Grant and Madsen (1979)
- ◊ Huynh-Thanh and Temperville (1991)
- ◊ Davies *et al.* (1988)
- ◊ Bijker (1967)
- ◊ Christoffersen and Jonsson (1985)
- ◊ O' Connor and Yoo (1988)
- ◊ Van Rijn *et al.* (2004)

Wall roughness

The friction of side walls is computed from a ‘law of the wall’ formulation if selected:

Slip condition

You can select one of the following:

- ◊ *Free*; zero tangential shear stress
- ◊ *Partial*; specify the *roughness length (m)*.
- ◊ *No*; zero velocity at the wall assumed.

In large-scale hydrodynamic simulations, the tangential shear stress for all lateral boundaries or vertical walls can be safely neglected (free slip). For simulations of small-scale flow (e.g. laboratory scale) the influence of the side walls on the flow can be significant and this flag can be activated. The partial slip only applies to the side walls of *permanently* dry areas, including dry points. The tangential shear stress is calculated based on a logarithmic law of the wall where the roughness length needs to be specified. See [chapter 9](#) for details.

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Manning	0.0	0.04	65.0	$m^{-1/3}s$
White-Colebrook	0.0	10.0	65.0	m
Chézy	0.0	1,000	65.0	$m^{1/2}/s$
Z_0	0.0	1.0	65.0	m



Remark:

- ◊ The default value is set for the Chézy-formulation.

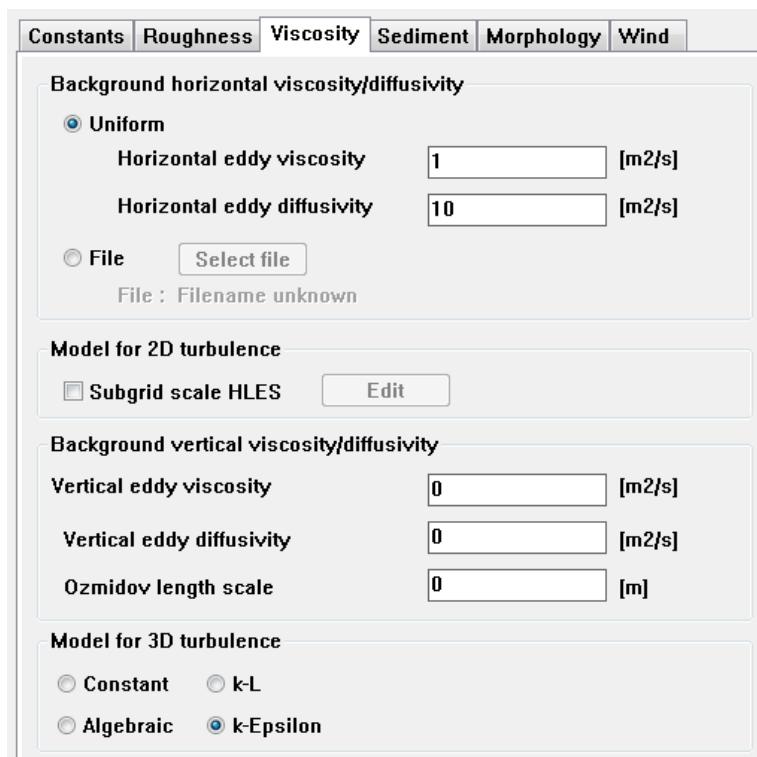


Figure 4.33: Defining the eddy viscosity and eddy diffusivity

4.5.7.2 Viscosity

The background horizontal eddy viscosity and horizontal eddy diffusivity are user-defined. The viscosity and diffusivity calculated with the Horizontal Large Eddy Simulation (HLES) sub-grid model will be added to the background values (for FLOW v3.50.09.02 and higher).

Remark:

- ◊ If you use an older version of FLOW, or if you re-run an old MDF-file with FLOW v3.50.09.02 or higher, be aware of this adding operation.



The user-defined parameters can be entered as one uniform value for each quantity (uniform for both directions) or as non-uniform values read from a user-specified file with file extension <ed>.

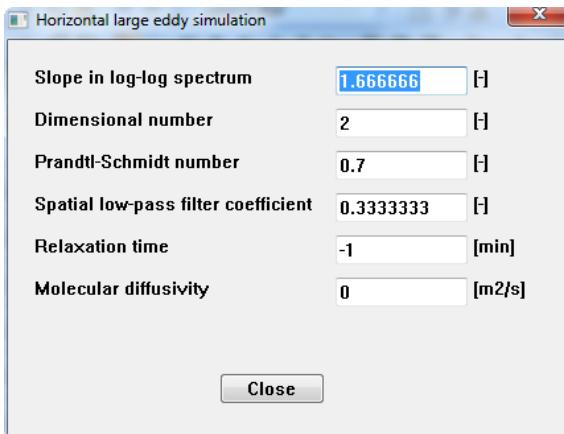
If the model is 3D, you may specify constant background values for the turbulent vertical eddy viscosity. If constituents are modelled you may also specify a constant background value for the vertical turbulent eddy diffusivity and the Ozmidov length scale.

Finally the model for 3D turbulence can be specified. For details see [section 9.5](#).

Upon selecting *Physical parameters* → *Viscosity* [Figure 4.33](#) will be displayed.

Horizontal eddy viscosity and diffusivity

You can select or specify the following parameters:

**Figure 4.34:** Window with HLES parameters**Background values***Uniform*

A uniform horizontal eddy viscosity and eddy diffusivity. The values are defined in the separate input fields.

Horizontal eddy viscosity

Specify a uniform horizontal eddy viscosity in [m²/s].

Horizontal eddy diffusivity

Specify a uniform horizontal eddy diffusivity in [m²/s].

File

Select the horizontal eddy viscosity and eddy diffusivity from an attribute file with extension <edt>. The data is not read.

Model for 2D turbulence*Subgrid scale HLES*

Upon selecting [Figure 4.34](#) will be displayed. Details about the input parameters and theory can be found in [section B.8](#).

**Remarks:**

- ◊ The contribution from the HLES model will be added to the background values.
- ◊ If the relaxation time is less than 0.0, the high-pass filtering operation will be cancelled.

Background vertical eddy viscosity and diffusivity

You can select or specify the following parameters:

Vertical eddy viscosity Specify a uniform vertical turbulent eddy viscosity in [m²/s].*Vertical eddy diffusivity* Specify a uniform vertical turbulent eddy diffusivity in [m²/s].

The resulting viscosity is the maximum of the user-defined background value and the calculated value from the turbulence model. Similar for the diffusivity.

Ozmidov length scale

The Ozmidov length scale determines the minimal value for the vertical diffusivity from the calculated vertical diffusivity and the equation using Brunt-Väisälä frequency and Ozmidov length scale; see [chapter 9](#) for details.

Model for 3D turbulence

Turbulence Model For a 3D simulation you can select the type of turbulence formulation used to determine the vertical turbulent eddy viscosity and the vertical turbulent eddy diffusivity additional to the background values. You can select from:

- ◊ *Constant* Only the background values will be applied.
- ◊ *Algebraic* The coefficients are determined by algebraic equations for the turbulent energy and the mixing length.
- ◊ *k-L* The coefficients are determined by a transport equation for the turbulent kinetic energy k and mixing length L .
- ◊ *k-epsilon* The coefficients are determined by transport equations for both the turbulent kinetic energy and the turbulent kinetic energy dissipation.

Remark:



- ◊ You do not need (can not) prescribe boundary conditions for the transport equations used in the selected turbulence model. The underlying assumption is that the turbulence is mainly generated inside the model area and not transported into the model area through the open boundaries. For very small area models this assumption might be violated and a special add-on to the flow module can be provided to prescribe the required boundary conditions.

In 3D-simulations the vertical turbulent eddy viscosity and turbulent diffusivity are computed using a turbulence closure model. The four models differ in their prescription of the turbulent kinetic energy, the dissipation of energy and/or the mixing length.

Constant eddy viscosity and diffusivity: you must specify background values that are used in both (horizontal) directions.

Algebraic model: the first closure model uses an algebraic formula for k and L .

k depends on the (friction) velocities, whereas for the mixing length L a function following Bakhtmetev (1932) is used. In the case of vertical density gradients, the turbulent exchanges are limited by buoyancy forces and the mixing length L must be corrected. This correction involves a so-called damping function. The approach is sufficiently accurate to be applied in cases where weakly stratified flow conditions exist. See [subsection 9.5.1](#) for full details.

$k-L$ model: the second closure model is a first-order turbulence closure model.

In this model an identical approach is applied for the mixing length L . To find the turbulent kinetic energy k a transport equation is solved. See [subsection 9.5.2](#) for full details.

$k-\varepsilon$ model: the third model is a $k-\varepsilon$ model which is a second-order turbulence closure model.

Here, both the turbulent kinetic energy k and the turbulent kinetic dissipation ε are prescribed by a transport equation. From k and ε the mixing length and the vertical turbulent eddy viscosity value are determined. The mixing length is now a property of the flow, and in the case of stratification no damping functions are needed. The eddy diffusivity is derived from the eddy viscosity. See [subsection 9.5.3](#) for full details.

Remark:



- ◊ Both parameters, the vertical eddy viscosity and eddy diffusivity, are defined at the layer interfaces. Therefore, when you plot or print these values you will obtain one value more

than the number of layers.

The value for both the horizontal eddy viscosity and the horizontal eddy diffusivity depends on the flow and the grid size used in the simulation. For detailed models where much of the details of the flow are resolved by the grid, grid sizes typically tens of metres or less, the values for the eddy viscosity and the eddy diffusivity are typically in the range of 1 to 10 [m²/s]. For large (tidal) areas with a coarse grid, grid sizes of hundreds of metres or more, the coefficients typically range from 10 to 100 [m²/s]. Both coefficients are so-called calibration parameters; their value must be determined in the calibration process.

The use of the uniform vertical eddy viscosity and eddy diffusivity coefficient depends on the selected turbulence closure model:

- ◊ If a *constant* turbulence closure model is selected the prescribed (constant) background values are applied.
- ◊ In all other cases the uniform values are used as a background (minimum) value for the turbulent contribution. The molecular viscosity and diffusivity are added to the turbulent part.

In case of strongly stratified flow, a background vertical eddy viscosity may be needed to damp out short oscillations generated by e.g. drying and flooding, boundary conditions, wind forcing etc. Without this background eddy viscosity these short oscillations will only be damped in the bottom layer. The layers behave as if they were de-coupled. The value suggested for the background eddy viscosity depends on the application. The value must be small compared to the vertical eddy viscosity calculated by the turbulence closure model. Our experience is that for stratified estuaries and lakes a value of 10⁻⁴ to 10⁻³ [m²/s] is suitable.

For the eddy diffusivity similar arguments hold as for the eddy viscosity with one exception. Actually, a background eddy diffusivity is not necessary and will generate additional vertical mixing of matter through the pycnocline. However, the access to this value is retained for reason of compatibility with the eddy viscosity and for experimental purposes.

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Horizontal eddy viscosity	0.0	100.0	10..0	m ² /s
Horizontal eddy diffusivity	0.0	1,000	10.0	m ² /s
Vertical eddy viscosity	0.0	100.0	1.0E-6	m ² /s
Vertical eddy diffusivity	0.0	1,000	1.0E-6	m ² /s
Ozmidov length scale	0.0	Max. number	0.0	m



Restriction:

- ◊ Due to the manner in which the horizontal shear stresses containing cross derivatives are numerically implemented (explicit integration) the, otherwise unconditionally stable, numerical scheme for the viscosity term will become conditionally stable. Please check the stability criterion in [chapter 10](#).

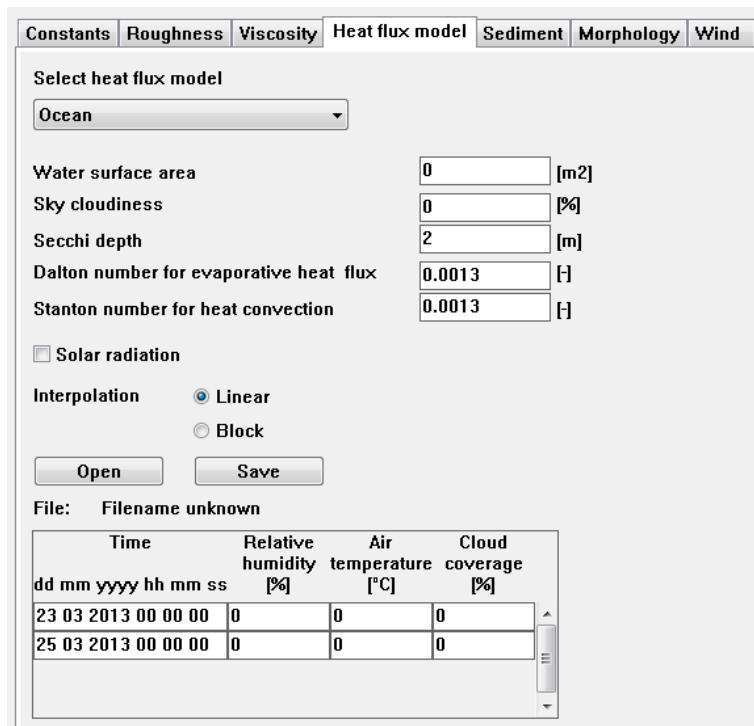


Figure 4.35: Sub-data group Heat flux model

4.5.7.3 Heat flux model

If you have switched on the *Temperature* process, you can specify how the exchange of heat through the free surface is modelled.

Upon selecting the tab **Heat flux model** in the Data Group *Physical parameters* the window given by [Figure 4.35](#) is displayed.

In [Figure 4.35](#) the *Ocean* heat flux model is selected. Depending on the selected heat flux model parameters are displayed or not.

Heat flux model

You can select one of the following options:

- | | |
|---|---|
| <i>No flux</i> | The background temperature is used throughout the model area. |
| <i>Absolute flux, total solar radiation</i> | The absolute temperature is computed. The relative humidity, air temperature and the (short wave) solar radiation for a clear sky have to be prescribed. The net atmospheric (long wave) radiation and the heat losses due to evaporation and convection are computed by the model. |
| <i>Absolute flux, net solar radiation</i> | The absolute temperature is computed. The relative humidity, air temperature and the combined net (short wave) solar and net (long wave) atmospheric radiation have to be prescribed. The terms related to heat losses due to evaporation and convection are computed by the model. |
| <i>Excess temperature</i> | The heat exchange flux at the air-water interface is computed; only the background temperature is required. |
| <i>Murakami</i> | The relative humidity, air temperature and the net (short wave) solar radiation have to be prescribed. The effective back radiation, and the |

heat losses due to evaporation and convection are computed by the model. The incoming radiation is absorbed as a function of depth. The evaporative heat flux is calibrated for Japanese waters.

Ocean

The relative humidity, air temperature and the fraction of the sky covered by clouds is prescribed (in %). The effective back radiation and the heat losses due to evaporation and convection are computed by the model. Additionally, when air and water conditions are such that free convection occurs, free convection of latent and sensible heat is computed by the model and added to the forced convection. This model formulation typically applies for large water bodies.

The absolute temperature models compute the heat fluxes through the water surface by considering incoming radiation, back radiation, evaporation and convection. Evaporation and convection depend on air temperature, water temperature, relative humidity and wind speed.

When modelling the effect of a thermal discharge, it is often enough to compute the excess temperature distribution and decay. Excess heat is transferred from the water surface to the atmosphere by evaporation, long-wave radiation and convection. This total heat flux is proportional to the excess temperature at the surface. The heat transfer coefficient depends mainly on the water temperature and the wind speed.



Remarks:

- ◊ If you have selected a heat flux model which includes evaporation, you can specify if the evaporated water should be taken into account. Specify `Maseva = #Y#` in Data Group *Additional parameters*.
- ◊ Precipitation and evaporation is an add-on of FLOW independent if you model temperature or not. See [section B.6](#) for how you can use this add-on. If you use this add-on together with the process *Temperature*, then the user-defined evaporation overrules the computed evaporation from the heat flux model. The user-defined evaporated water is always included in the continuity equation.

Global parameters for the heat flux models

You can define:

<i>Water surface area</i>	The water area exposed to wind in [m ²]. Required for heat flux models 1, 2 and 3.
---------------------------	--

A secondary effect is included in the heat models by taking into account the surface area of the water body exposed to the wind. Wind strongly affects the evaporation and convection process. The value for the surface area can differ from the effective water surface in the model. Its value depends on the model geometry and on the direction of the wind ([Sweers, 1976](#)). Effectively, the surface area can be regarded as a calibration parameter.

<i>Sky cloudiness</i>	The fractional blocking of the solar radiation by clouds in [%]. Required by heat flux models 1, 4 and 5.
-----------------------	---

The incoming solar and atmospheric radiations are affected by clouds. The *Sky cloudiness* parameter describes the percentage of the solar radiation blocked by the clouds; 0 % refers to clear sky condition whereas a value equal to 100 % implies that most of the solar radiation will be absorbed by the atmosphere. Only 35 % (excluding the reflection) of the specified solar radiation will reach the water surface. Consequently, due to increased absorption the atmosphere will ultimately radiate more energy.

<i>Secchi depth</i>	Required by heat flux models 4 and 5.
---------------------	---------------------------------------

<i>Dalton number</i>	for evaporative heat flux. Required by heat flux models 4 and 5.
<i>Stanton number</i>	for heat convection. Required by heat flux model 5.

Time dependent heat flux model input

<i>Interpolation</i>	The type of interpolation used for the time-series at intermediate time instances. You can select either <i>Linear</i> or <i>Block</i> interpolation.
<i>Open / Save</i>	To open and read or save the time-series from or to an attribute file with extension <tem>.

In the table you must specify time-series for the following parameters:

- ◊ Incoming solar radiation for a clear sky in J/(m²s) (only for heat flux model 1)
- ◊ Net (short wave) solar radiation in J/(m²s) (heat flux model 4)
- ◊ Combined net (short wave) solar and net (long wave) atmospheric radiation in J/(m²s) (heat flux model 2)
- ◊ Background temperature in °C (*Excess* heat flux model)
- ◊ Air temperature in °C (all heat flux models except the *Excess* model)
- ◊ Relative humidity in % (all heat flux models except the *Excess* model)
- ◊ Fraction of the sky covered by clouds (in %) (*Ocean* heat flux model)

The background temperature used in the excess temperature heat model is already prescribed in Data Group *Physical parameters* → *Constants*.

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Water surface	0.0	Max. Number	0.0	m ²
Sky cloudiness	0.0	100.0	0.0	%
Secchi depth	0.0		2.00	m
Dalton number	0.0	1.0	0.00130	-
Stanton number	0.0	1.0	0.00130	-
Temperature	0.0	60.0	0.0	°C
Humidity	0.0	100.0	0.0	%
Solar radiation	0.0	Max. Number	0.0	W/m ²

Restrictions:

- ◊ The time breakpoints must differ from the simulation start time by an integer multiple of the time-step (Data Group *Time frame*).
- ◊ The first time breakpoint must be before or equal to the simulation start time, and the last time breakpoint must be equal or later than the simulation stop time (see Data Group *Time frame*).
- ◊ Time breakpoints must be given in ascending order.



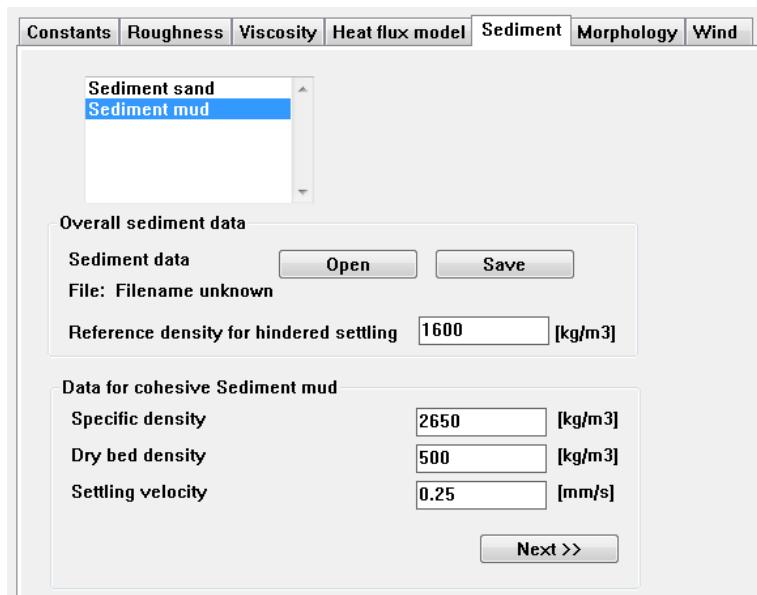


Figure 4.36: Sub-data group Sediment, overall and cohesive sediment parameters

4.5.7.4 Sediment

Sediment transport offers the following functionalities (for details see [chapter 11](#)):

- ◊ Cohesive sediment transport, including the effect of salinity on flocculation.
- ◊ Non-cohesive suspended sediment (sand) transport.
- ◊ Bedload transport, including effect of wave asymmetry.
- ◊ Influence of waves and hindered settling.
- ◊ Updating the bed-level and feedback to the hydrodynamics.
- ◊ Up to 99 fractions of sand or combination of sand and mud.
- ◊ Numerous sediment transport formulations

If you have switched on the *Sediments* process, you can specify the characteristics of the (non-) cohesive sediments, the critical shear stresses for sedimentation and erosion (for cohesive sediments) and the initial sediment at the bed.

Upon selecting the tab **Sediment** in the Data Group *Physical parameters* the window given by [Figure 4.36](#) is displayed.

Open / Save The sediment characteristics can be read from or saved to the file <*.sed>. See [section B.9](#) for a detailed description.

Overall sediment data

Reference density for hindered settling In high concentration mixtures, the settling velocity of a single particle is reduced due to the presence of other particles. In order to account for this hindered settling effect [Richardson and Zaki \(1954\)](#) are followed. The reference density is a parameter in their formulation. See [subsection 11.1.4](#).

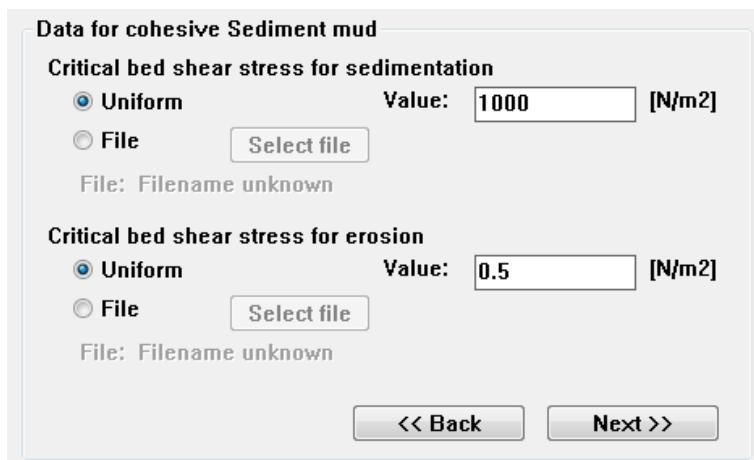


Figure 4.37: Sub-data group Sediment, cohesive sediment parameters (continued)

Cohesive sediment

Specific density

For sediment transport the Eckart relation is extended to include the density effect of sediment fractions in the fluid mixture, see [subsection 11.1.3](#). The specific density of each sediment fraction is part of this formulation.

Dry bed density

The thickness of the sediment above the fixed layer is calculated by dividing the mass of sediment available at the bed by the user specified dry bed density.

Settling velocity

The settling velocity of the cohesive sediment.

Remarks:

- ◊ The settling velocity has to be specified in mm/s in the GUI but is written in [m/s] to the <*.sed> file
- ◊ In salt water cohesive sediment tends to flocculate to form sediment “flocs”, with the degree of flocculation depending on the salinity of the water. This has been implemented, see [subsection 11.2.1](#). Though the GUI does not support this formulation, you can still edit the <*.sed> file to use this functionality.



➤ Press *Next* to browse to the next cohesive input parameters, showing [Figure 4.37](#).

For cohesive sediment fractions the fluxes between the water phase and the bed are calculated with the well-known Partheniades-Krone formulations ([Partheniades, 1965](#)):

Critical bed shear stress for sedimentation

If the bed shear stress is larger than the critical stress, no sedimentation takes place. If the bed shear stress is smaller than the flux is calculated following Partheniades-Krone, see [subsection 11.2.3](#).

Critical bed shear stress for erosion

If the bed shear stress is smaller than the critical stress, no erosion takes place. If the bed shear stress is larger than the flux is calculated following Partheniades-Krone, see [subsection 11.2.3](#).

Space-varying shear stress files can be made with QUICKIN.

➤ Press *Next* to browse to the next cohesive input parameters, showing [Figure 4.38](#).

Sediment erosion rate Erosion parameter in the formulation of Partheniades-Krone, see [subsection 11.2.3](#).

Initial sediment layer thickness at bed The initial sediment layer thickness at the bed in metre.

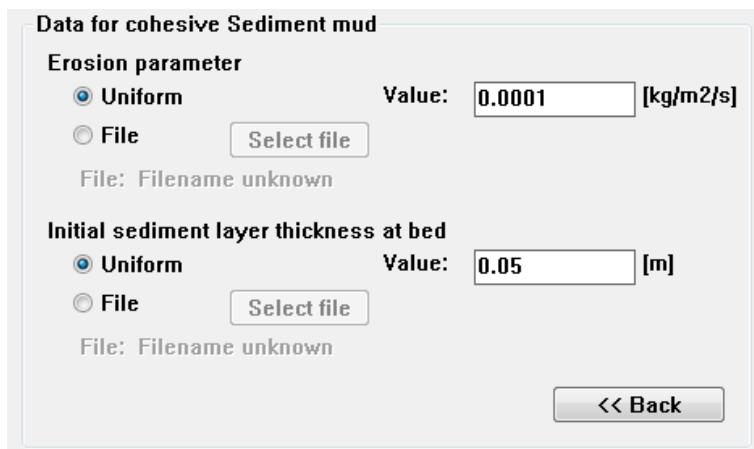


Figure 4.38: Sub-data group Sediment, cohesive sediment parameters (continued)



Remarks:

- ◊ The initial bed composition should be specified at the cell centres. You can use QUICKIN with the *Operations → Data in Cell Centre* option on to prepare the file with space-varying initial sediment layer thickness at the bed.
- ◊ GUI v3.39.25 and higher writes a new keyword (`IniSedThick`) to the `<*.sed>` file. The sediment now is in metre, before it was in kg/m^2 . Old `<*.sed>` files will be converted by the GUI.

Non-cohesive sediment

If you have defined a non-cohesive sediment fraction, you have to specify the following parameters, see [Figure 4.39](#).

Specific density

For sediment transport the Eckart relation is extended to include the density effect of sediment fractions in the fluid mixture, see [subsection 11.1.3](#). The specific density of each sediment fraction is part of this formulation.

Dry bed density

The thickness of the sediment above the fixed layer is calculated by dividing the mass of sediment available at the bed by the user specified dry bed density.

Median sediment diameter

The settling velocity of a non-cohesive ("sand") sediment fraction is computed following the method of [Van Rijn \(1993\)](#). The formulation used depends on the diameter of the sediment in suspension: See [subsection 11.3.1](#).

Remark:

- ◊ The sediment diameter has to be specified in μm in the GUI but is written in m to the `<*.sed>` file.

The initial available sediment at the bed.

Remarks:

- ◊ The initial bed composition should be specified at the cell centres. You can use QUICKIN with the *Operations → Data in Cell Centre* option on to prepare the file with space-varying initial sediment layer thickness at the bed.
- ◊ GUI v3.39.25 and higher writes a new keyword (`IniSedThick`) to the `<*.sed>` file. The sediment now is in metre, before it was in kg/m^2 . Old `<*.sed>` files will be converted by the GUI.



Initial sediment layer thickness at bed



Data for non-cohesive Sediment sand

Specific density	2650	[kg/m ³]
Dry bed density	1600	[kg/m ³]
Median sediment diameter (D ₅₀)	200	[μm]
Initial sediment layer thickness at bed	<input checked="" type="radio"/> Uniform Value: <input type="text" value="5"/> [m]	
<input type="radio"/> File	<input type="button" value="Select file"/>	
File: Filename unknown		

Figure 4.39: Sub-data group Sediment, non-cohesive sediment parameters

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Reference density	100.0	Max. Number	1600.0	kg/m ³
Specific density	100.0	4000.0	2650.0	kg/m ³
Dry bed density	mud: 100.0 sand: 500.0	3000.0 3000.0	mud: 500.0 sand: 1600.0	kg/m ³ kg/m ³
Settling velocity	> 0.0	30.0	0.25	mm/s
Median sediment diameter	64.0	2000.0	200	μm
Critical shear stress for sedimentation	0.0	1000.0	1000.0	N/m ²
Critical shear stress for erosion	0.001	100.0	0.5	N/m ²
Sediment erosion rate	0.0	1.0	0.0001	kg/(m ² s)
Initial sediment layer at bed	0.0	mud: 10.0 sand: 50.0	mud: 0.05 sand: 5.0	m

Remarks:

- ◊ The settling velocity value is written in [m/s] to the file <*.sed>.
- ◊ The mean sediment diameter value is written in [m] to the file <*.sed>.



4.5.7.5 Morphology

With the feedback of bottom changes to the hydrodynamic computation you can execute a full morphodynamic computation. You can include the influence of waves by running this version of Delft3D-FLOW in combination with the Delft3D-WAVE module. See [section 11.6](#) for details.

➤ Selecting the tab **Morphology** in the Data Group *Physical parameters* opens [Figure 4.40](#).

Open / Save The morphology characteristics can be read from or saved to the file <*.mor>. See [section B.9](#) for a detailed description.

General

Update bathymetry during FLOW simulation If you want to take into account the feedback of bottom changes on the hydrodynamics, tick off this option.



Remark:

◊ The use of this update option only affects the updating of the depth values (at ζ and velocity points) used by flow calculations at subsequent time-steps; the quantity of sediment available at the bed will still be updated, regardless of the state of this flag.

Include effect of sediment on fluid density You can include or neglect the effect of sediment on the fluid density by setting this option accordingly.

For coarser non-cohesive material you can specify that, at all open inflow boundaries, the flow should enter carrying all “sand” sediment fractions at their equilibrium concentration profiles. This means that the sediment load entering through the boundaries will be near-perfectly adapted to the local flow conditions, and very little accretion or erosion should be experienced near the model boundaries.

Equilibrium sand concentration profile at inflow boundary Tick off to use this option. When not activated the inflow concentrations specified in Data Group *Boundaries* → *Transport conditions* will be used.

One of the complications inherent in carrying out morphological projections on the basis of hydrodynamic flows is that morphological developments take place on a time scale several times longer than typical flow changes (for example, tidal flows change significantly in a period of hours, whereas the morphology of a coastline will usually take weeks, months, or years to change significantly). One technique for approaching this problem is to use a *Morphological time scale factor* whereby the speed of the changes in the morphology is scaled up to a rate that it begins to have a significant impact on the hydrodynamic flows.

Morphological scale factor The above can be achieved by specifying a non-unity value.



Remark:

◊ The *Morphological scale factor* can also be time-varying, see [subsection B.9.8](#). This feature is not yet supported by the GUI. You have to edit the <*.mor> file manually.

The implementation of the *Morphological time scale factor* is achieved by simply multiplying the erosion and deposition fluxes from the bed to the flow and vice-versa by this scale factor, at each computational time-step. This allows accelerated bed-level changes to be incorporated dynamically into the hydrodynamic flow calculations.

Frequently, a hydrodynamic simulation will take some time to stabilise after transitioning from

The screenshot shows the 'Morphology' tab selected in a top navigation bar. Below it is a 'Morphological data' section with 'Open' and 'Save' buttons. A note says 'File: Filename unknown'. Under 'General', three checkboxes are checked: 'Update bathymetry during FLOW simulation', 'Include effect of sediment on fluid density', and 'Equilibrium sand concentration profile at inflow boundaries'. There are three input fields: 'Morphological scale factor' (1), 'Spin-up interval before morphological changes' (720 min), and 'Minimum depth for sediment calculation' (0.1 m). In the 'Sediment transport parameters' section, there are three input fields: 'van Rijn's reference height factor' (1), 'Threshold sediment thickness' (0.05000000 m), and 'Estimated ripple height factor' (2). The 'Multiplication (calibration) factors' section contains five input fields: 'Factor for erosion of adjacent dry cells' (0), 'Current-related reference concentration factor' (1), 'Current-related transport vector magnitude factor' (1), 'Wave-related suspended transport factor' (1), and 'Wave-related bed-load transport factor' (1).

Morphological data	
<input type="button" value="Open"/>	<input type="button" value="Save"/>
File: Filename unknown	
General	
<input checked="" type="checkbox"/> Update bathymetry during FLOW simulation	
<input type="checkbox"/> Include effect of sediment on fluid density	
<input checked="" type="checkbox"/> Equilibrium sand concentration profile at inflow boundaries	
Morphological scale factor	<input type="text" value="1"/> [H]
Spin-up interval before morphological changes	<input type="text" value="720"/> [min]
Minimum depth for sediment calculation	<input type="text" value="0.1"/> [m]
Sediment transport parameters	
van Rijn's reference height factor	<input type="text" value="1"/> [H]
Threshold sediment thickness	<input type="text" value="0.05000000"/> [m]
Estimated ripple height factor	<input type="text" value="2"/> [H]
Multiplication (calibration) factors	
Factor for erosion of adjacent dry cells	<input type="text" value="0"/> [H]
Current-related reference concentration factor	<input type="text" value="1"/> [H]
Current-related transport vector magnitude factor	<input type="text" value="1"/> [H]
Wave-related suspended transport factor	<input type="text" value="1"/> [H]
Wave-related bed-load transport factor	<input type="text" value="1"/> [H]

Figure 4.40: Sub-data group Morphology

the initial conditions to the (dynamic) boundary conditions. It is likely that during this stabilisation period the patterns of erosion and accretion that take place do not accurately reflect the true morphological development and should be ignored.

<i>Spin-up interval before morphological changes</i>	Specify a time interval (in minutes after the start time) after which the morphological bottom updating will begin. During this time interval all other calculations will proceed as normal (sediment will be available for suspension for example) however the effect of the sediment fluxes on the available bottom sediments will not be taken into account.
<i>Minimum depth for sediment calculation</i>	In the case of erosion near dry points a threshold depth for computing sediment transport can be used, see subsection 11.6.2 .

Sediment transport parameters

The following parameters are only relevant for non-cohesive sediments.

<i>van Rijn's reference height factor</i>	For non-cohesive sediment (e.g. sand), we follow the method of Van Rijn (1993) for the combined effect of waves and currents, see section 11.3 . The reference height formulation contains a proportionality factor called <i>van Rijn's reference height factor</i> .
---	--

Because of the more complex, partly explicit — partly implicit, erosive flux terms used for “sand” type sediments it is not possible to use the simple source limitation technique used for cohesive sediments. Instead, you must specify a *Threshold sediment thickness*.

<i>Threshold sediment thickness</i>	At each time-step the thickness of the bottom sediments is calculated. If the remaining sediment thickness is less than the user-specified threshold <i>and erosive conditions are expected</i> then the source and sink sediment flux terms, see Eq. (??), are reduced in the following manner:
<i>Estimated ripple height factor</i>	In case of waves, the wave related roughness is related to the estimated ripple height, see subsection 11.3.3 .

Multiplication (calibration) factors

The following parameters are only relevant for non-cohesive sediments.

In the case of erosion near a dry beach or bank, the standard scheme will not allow erosion of the adjacent cells, even when a steep scour hole would develop right next to the beach. Therefore a scheme has been implemented, where for each wet cell, if there are dry points adjacent to it; the erosion volume is distributed over the wet cell and the adjacent dry cells.

<i>Factor for erosion of adjacent dry cells</i>	The distribution is governed by a user-specified <i>factor for erosion of adjacent dry cells</i> , which determines the fraction of the erosion to assign (evenly) to the adjacent cells; if this factor equals zero, the standard scheme is used; if this factor equals 1, all erosion that would occur in the wet cell is assigned to the adjacent dry cells.
<i>Current-related reference concentration factor</i>	The reference concentration is calculated in accordance with Van Rijn et al. (2000) , see subsection 11.3.3 .

Only [Van Rijn \(1993\)](#) and [Van Rijn et al. \(2004\)](#) compute explicitly a wave-related transport component.

$$S_{\text{total}} = \text{BED} \cdot S_{\text{bedload current}} + \text{BEDW} \cdot S_{\text{bedload waves}} + \text{SUSW} \cdot S_{\text{suspended waves}} \quad (4.11)$$

$$\frac{dDPS}{dt} = S_{\text{total out}} - S_{\text{total in}} + \text{SUS} \cdot (\text{Entrainment} - \text{Deposition}) \quad (4.12)$$

where $S_{\text{bedload current}}$, $S_{\text{bedload waves}}$, $S_{\text{suspended waves}}$, Entrainment, and Deposition depend on the sediment transport formula used. In most cases it holds $S_{\text{bedload waves}} = 0$ and $S_{\text{suspended waves}} = 0$.

Current-related transport vector magnitude factor: —

Wave-related suspended transport factor: The wave-related suspended sediment transport is modelled using an approximation method proposed by [Van Rijn \(2001\)](#), see [subsection 11.5.1](#).

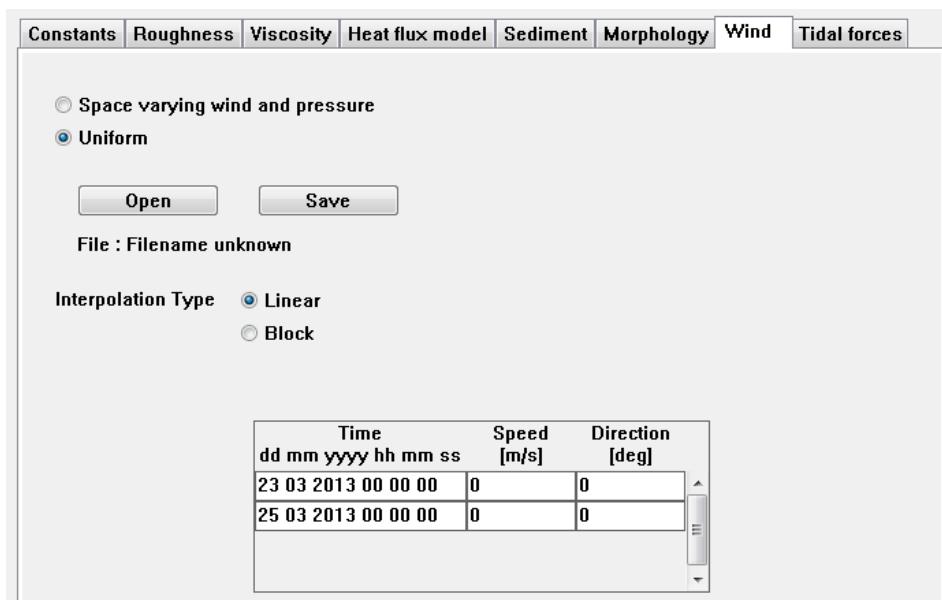
Wave-related bedload transport factor: —

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Update bathymetry during FLOW simulation	yes or no		yes	none
Include effect of sediment on fluid density	yes or no		no	none
Equilibrium sand concentration profile at inflow boundary	yes or no		yes	none
Morphological scale factor	0.0	10000.0	1.0	-
Spin-up interval	0.0		720.0	min
Minimum depth for sediment calculation	0.1	10.0	0.1	m
Van Rijn's reference height	0.4	2.0	1.0	-
Threshold sediment thickness	0.005	10.0	0.05	m
Estimated ripple height factor	1.0	5.0	2.0	-
Factor for erosion of adjacent dry cells	0.0	1.0	0.0	-
Current-related reference concentration factor	0.0	100.0	1.0	-
Current-related transport vector magnitude factor	0.0	100.0	1.0	-
Wave-related suspended transport factor	0.0	100.0	1.0	-
Wave-related bedload transport factor	0.0	100.0	1.0	-

4.5.7.6 Wind

When the *Wind* process is activated in Data Group *Processes*, see [Figure 4.16](#), the tab **Wind** is available in the Data Group *Physical parameters*. Selecting this tab displays [Figure 4.41](#).

**Figure 4.41:** Wind definition window

Wind fields can be specified as uniform but time dependent or as a space and time varying wind (and pressure) field. In the **Wind** tab you can select:

- | | |
|--|--|
| <i>Space varying wind and pressure</i> | To define a space and time varying wind and pressure field. Open the file with wind and pressure data. |
| <i>Uniform</i> | To define a wind field that is time dependent but uniform in space. |

Upon selecting *Uniform* you can either specify the wind data or read an attribute file.

Uniform wind

For a uniform wind you can specify:

- | | |
|---------------------------|--|
| <i>Open</i> | To read an attribute file with a uniform but (potentially) time dependent wind field; file extension <wnd>. |
| <i>Save</i> | To save the manually defined or modified wind field to a file. If you want to combine wind data read from a file and defined manually you must first import the file and then add, delete or modify the imported data; reversing this order will overwrite your manually defined data. |
| <i>Interpolation Type</i> | Time dependent data is specified at time instances called breakpoints; to determine the value at intermediate times you can apply a linear or a block type of interpolation. |
| <i>Time-Series</i> | Specify the time breakpoints in [dd mm yyyy hh mm ss], the wind speed in [m/s] and the wind direction in degrees (nautical definition). |

The wind direction is defined according to the nautical definition, i.e. relative to true North and positive measured clockwise. In [Figure 4.42](#) the wind direction is about +60 degrees, i.e. an East-North-East wind.



Remark:

- ◊ The wind direction is measured (clockwise) starting from true North. East-North-East wind thus has a direction of +67.5 degrees.

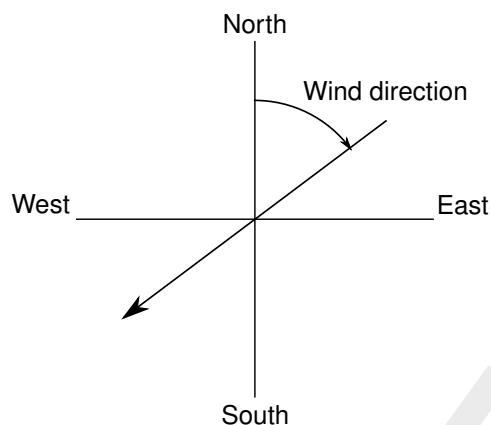


Figure 4.42: Nautical definition of the wind direction

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Wind speed	0.0	100.0	0.0	m/s
Wind direction	0.0	360.0	0.0	degrees North

Restrictions:

- ◊ The time breakpoints must differ from the simulation start time by an integer multiple of the time-step (see Data Group *Time frame*).
- ◊ The first time breakpoint must be before or equal to the simulation start time, and the last time breakpoint must be equal or later than the simulation stop time (see Data Group *Time frame*).
- ◊ Time breakpoints must be given in ascending order.



Space varying wind and pressure

For a space varying wind and pressure you can specify:

Select file

To select an attribute file with space and time varying wind and pressure data. The file is not read.

Interpolation Type

Time dependent data is specified at time instances called breakpoints; to determine the value at intermediate times you can apply a linear or a block type of interpolation.



Remarks:

- ◊ The space varying wind and pressure selected in this sub-data group must be defined on the hydrodynamic grid.
- ◊ It is also possible to use space varying wind and pressure defined on an independent wind grid. See [subsubsection A.2.8.2](#) and [section B.7](#) for details.
- ◊ When using space varying wind and pressure, the input is applied at the interior of the domain. On open boundaries this can cause a pressure gradient, which in turn causes a water level gradient. To reduce this effect, it is possible to apply a pressure correction on open boundaries. This functionality is described in [subsection B.7.1](#).

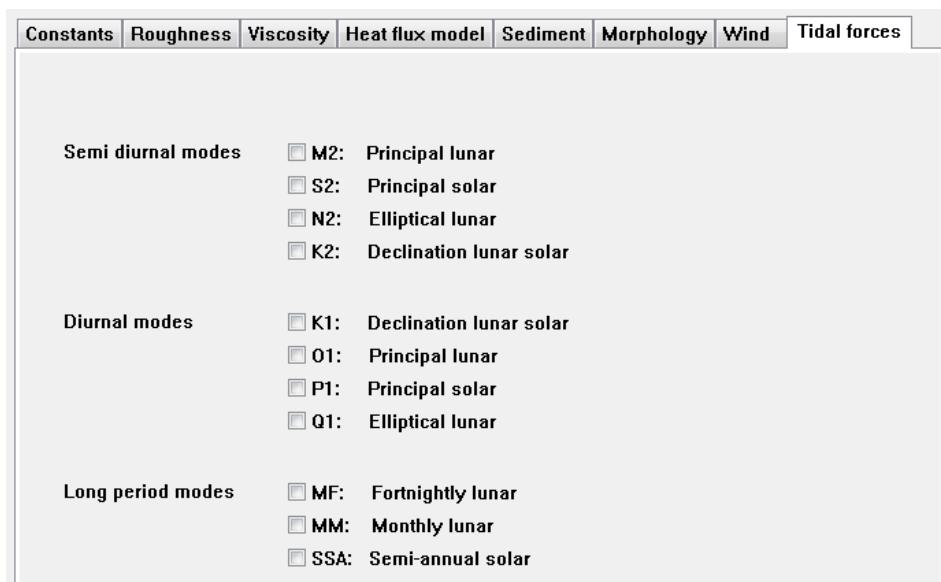


Figure 4.43: Sub-window **Tidal forces**

4.5.7.7 Tidal forces

When the process **Tidal forces** is selected (see [Figure 4.16](#)), the tab **Tidal forces** is available in Data Group *Physical parameters*. Selecting this tab [Figure 4.43](#) is displayed.

You can select the components that are taken into account by the tidal force generating routine by ticking off the check box. See [section 9.9](#) for details of tide generating forces.

4.5.8 Numerical parameters

In the Data Group *Numerical parameters* you can specify parameters related to drying and flooding and some other advanced options for numerical approximations.

Several *drying/flooding procedures* are available in Delft3D-FLOW, see [chapter 10](#). By default the water depth in each of the grid cell faces (velocity points) of a computational grid cell is determined to decide if the flow should be blocked by a (temporary) thin dam. If the flow at the four faces of a computational grid cell is blocked this grid cell is set dry. Note that this does not guarantee the water depth at the grid cell centre (water level point) to be positive definite. An additional check can be done on the water depth at the centre (water level point). The bottom depth in the cell centre (water level point) is not uniquely defined, several options are available. In combination with the water level in the cell centre it is determined if the cell is being dried or flooded; see [section 10.8](#) for more details on drying and flooding.



Click the Data Group *Numerical parameters*. This results in [Figure 4.44](#). **Note:** if your model is not 3D, or you have not modelled salinity or another constituent, then you will see less input parameters.

You must select or define one or more of the following parameters:

Drying and flooding check Determine if you want to apply an additional check on the water depth at the *cell centres* (water level points); or only a check at the *cell faces* (velocity points).

Depth at grid cell faces You can either select (see also [subsection 10.8.2](#)):

- ◊ *Mean* The depth at the cell face is the mean of the depths at the

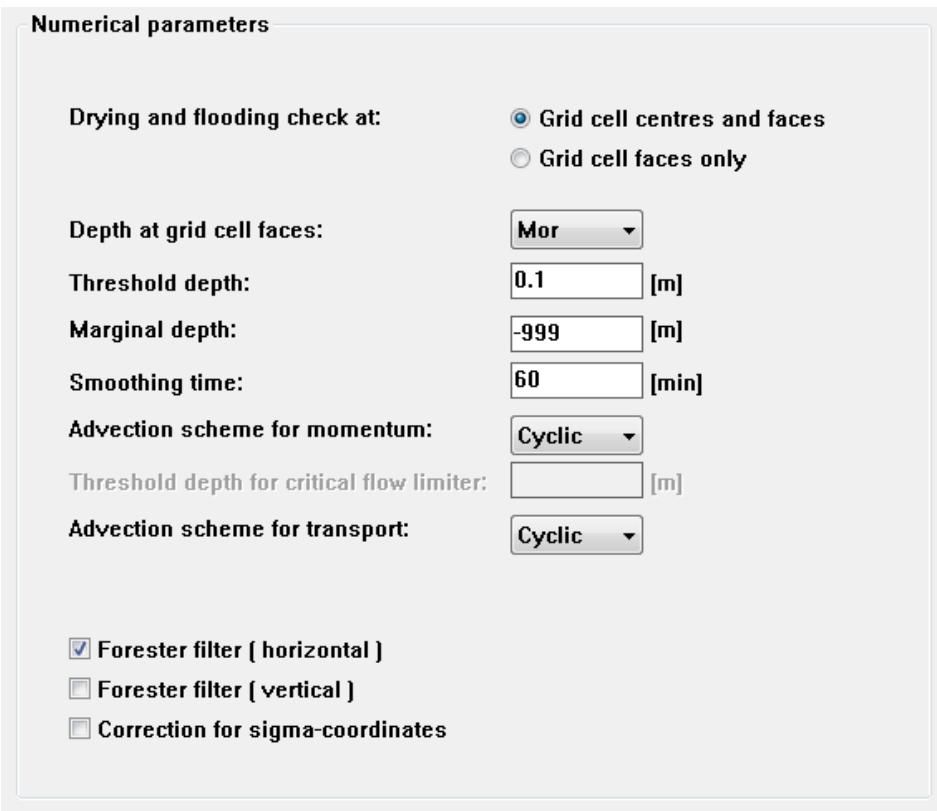


Figure 4.44: Data Group **Numerical parameters**

cell corners (m,n) and (m,n+1).

- ◊ *Min* The depth at the cell face is the minimum of the depths at the water level points (m,n) and (m+1,n).
- ◊ *Upwind* The depth at the *u*-velocity point is the depth at water level point (m,n) if the *u*-velocity is larger than 0, the depth at water level point (m+1,n) if the *u*-velocity is less than 0, and the average if the *u*-velocity is 0.
- ◊ *Mor* The *Mor* option is the same as *Min*.

Remark:

- ◊ *Depth specified at grid cell centres* ([subsubsection 4.5.2.2](#)) and *Depth at grid cell faces* is *Mean*, can not be used together.



Threshold depth

The threshold depth above which a grid cell is considered to be wet. The threshold depth must be defined in relation to the change of the water depth per time step in order to prevent the water depth to become negative in just one time step. This would result in iterations in the computation and thus a larger computational time.

As a rule of thumb you can use:

$$\delta \geq \left| \frac{\partial \zeta}{\partial t} \right| \Delta t \approx \frac{2\pi|a|}{T} \Delta t = \frac{2\pi|a|}{N} \quad (4.13)$$

where δ is the threshold depth, $\left| \frac{\partial \zeta}{\partial t} \right|$ is a measure for the change of the water level as a function of time with a as characteristic amplitude, T is the tidal period, Δt is the time step of the computation and N is the number of time steps per tidal period.

Example:

With a tidal amplitude of 2 m and time step of 1 minute ($N = 720$) we find $\delta \geq 0.017$ m.

Marginal depth

For the water level at the velocity points as default the mean value is used. It is also possible to determine the water levels at the cell faces with a so-called upwind approach: initially the water depth is the bottom depth at the velocity point plus the water level at the velocity point.

If this water depth is less than the marginal depth then:

- ◊ The water depth equals the bottom depth plus the water level at (m,n) if the u -velocity is larger than 0.
- ◊ The water depth equals the bottom depth plus the water level at (m+1,n) if the u -velocity is less than 0.
- ◊ The water depth equals the bottom depth plus the maximum of the water levels at (m,n) and (m+1,n) if the u -velocity is 0.

Smoothing time

The time interval used at the start of a simulation for a smooth transition between initial and boundary conditions.

In some cases, the prescribed initial condition and the boundary values at the start time of the simulation do not match. This can introduce large spurious waves that enter the model area. Subsequently, the wave will be reflected at the internal boundaries and along the open boundaries of the model until the wave energy is dissipated completely by bottom or internal (eddy viscosity) friction forces. These reflections can be observed as spurious oscillations in the solutions and they will enhance the spin-up (warming-up) time of the model. To reduce this time, a smooth transition period can be specified, during which the boundary condition will be adapted gradually starting from the prescribed initial condition value.

For the spatial discretization of the horizontal advection terms, three options are available in Delft3D-FLOW. The first and second option use higher-order dissipative approximations of the advection terms. The time integration is based on the ADI-method. The first scheme is denoted as the *Cyclic* method; see [Stelling and Leendertse \(1992\)](#). The second scheme is denoted as the *WAQUA*-scheme, see [Stelling \(1984\)](#) and [Stelling and Leendertse \(1992\)](#). The *WAQUA*-scheme and the *Cyclic* method do not impose a time step restriction. The third scheme can be applied for problems that include rapidly varying flows for instance in hydraulic jumps and bores ([Stelling and Duinmeijer, 2003](#)). The scheme is denoted as the *Flooding* scheme and was developed for 2D simulations with a rectilinear grid of the inundation of dry land with obstacles such as road banks and dikes. The integration of the advection term is explicit and the time step is restricted by the Courant number for advection. See [subsection 10.4.1](#) for more details.

Advection scheme for momentum

Select the type of numerical method for the advective terms in the momentum equation.

Currently, the following numerical schemes are supported:

- ◊ the *Cyclic* scheme
- ◊ the *Waqua* scheme
- ◊ the *Flood* scheme

Default the standard *Cyclic* method is applied; see [chapter 10](#).

For the *Flooding* scheme the accuracy in the numerical approximation of the critical discharge rate for flow with steep bed slopes, can be increased by the use of a special approximation (slope limiter) of the total water depth at a velocity point downstream, see [section 10.8](#) for

details.

Threshold depth for critical flow limiter

Advection scheme for transport

For the *Flood* scheme you have to specify a threshold depth which determines how the water depth is calculated.

Select the type of numerical method for the advective terms in the advection-diffusion equation for constituents.

Currently, two numerical schemes are supported:

- ◊ the *Cyclic* method (accurate, but positive definite results are not guaranteed).
- ◊ *Van Leer-2* method (less accurate, but positive definite results guaranteed).

Default the standard *Cyclic* method is applied; see [chapter 10](#). However, when a monotonous solution (no over- and undershoots) in the horizontal is required, then the slightly less accurate but monotonous *Van Leer-2* method can be applied (see [chapter 10](#)). It might still be necessary to remove wiggles generated by the central difference scheme in the vertical.

Remark:

- ◊ The continuity equation is solved iteratively, see [subsection 10.4.4](#). !
- The default number of iterations is 2. With the keyword and value `Iter = #`, where `#` is the required number of iterations, you can specify the number of iterations. You can add this keyword and value manually to the MDF-file or specify them under *Additional parameter*, see [subsection 4.5.11](#).

Relaxation factor QH-forcing

This factor only applies if QH-forcing is used at open boundaries. In some cases, when discharge fluctuates quickly, under-relaxation is necessary to stabilise the forcing by a QH-relation. This parameter in the range between 0 and 1 specifies the amount of under-relaxation. If no relaxation is required, this parameter should be set to 0.

Forester filter

Activate or de-activate a Forester filter in the horizontal or in the vertical direction.

In the horizontal direction numerical diffusion is added in points where the quantity (concentration) is negative. The diffusion is two-dimensional along σ -planes. The horizontal Forester filter acts on the salinity, the temperature and the concentrations of dissolved substances, including suspended sediments. The horizontal Forester filter is not applied to the spiral motion intensity, as this quantity can be negative by its nature.

In the vertical direction the solution is made monotone for salinity and temperature only; i.e. the vertical filter is not applied to constituents or suspended sediments. Wiggles, generated by the central differences in the vertical, are smoothed.

When the Forester filter is applied in both directions, the horizontal filter is applied first, followed by the filter in the vertical direction.

Remark:

- ◊ In 3D sediment transport computations, see [section B.9](#), small negative sediment concentrations (smaller than 1.10^{-3} kg/m³) can be found. These negative concentrations can be suppressed by applying a horizontal Forester filter. However, the penalty can be a substantially larger computational time.

Correction for sigma co-ordinates

Activate or de-activate an anti-creep correction. σ co-ordinates have a slight disadvantage in case large bottom gradients exist in the model. Owing to truncation errors, artificial vertical diffusion and ar-

tificial flow may occur in this case. A counter measure (a so-called anti-creep approach) can be activated to suppress this phenomenon.

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Drying and flooding check	At cell faces always; optional at cell centres		at both	-
Depth specified at	grid cell corners or centres		cell corners	-
Depth at grid cell centres	Max, Mean, Min		Max	-
Depth at grid cell faces	Mean, Min, Upwind, Mor		Mean	-
Threshold depth	0.0	10	0.1	m
Marginal depth	-	-	-999.999	m
Smoothing time	0.0	n.a.	0.0	min
Advection scheme momentum	Cyclic, Waqua, Flood		Cyclic	none
Threshold depth flow limiter	0.0		0.0	m
Advection scheme transport	Cyclic, Van Leer-2		Cyclic	none
Relaxation factor QH	0.0	1.0	0.0	-
Forester filters	on or off		off	none
Correction for sigma co-ordinates	on or off		off	



Remark:

◊ A warning will be issued if the *Threshold depth* exceeds 1.0 m.

4.5.9 Operations

Upon selecting the Data Group *Operations* the sub-window of Figure 4.45 is displayed. The tab **Dredging and dumping** is only available if the *Sediments* process has been selected and the man-made process *Dredging and dumping* has been switched on.

4.5.9.1 Discharge

In Delft3D-FLOW intake stations and waste-water outfalls can be considered as localised discharges of water and dissolved substances. A discharge (or source) in the model domain is characterised by a name which serves as identification, and its position in the grid (respectively m, n and k). If the discharged effluent is to take place at one point in the vertical, then the layer number k has to be specified. However, if the discharged flow is to be distributed uniformly over all layers (taking into account the layer thickness), then the layer number K must be set to 0.

In addition to the discharge rate the concentration of the constituents released and the temperature and/or the salinity, depending on the selections made in the Data Group *Processes* must be specified.

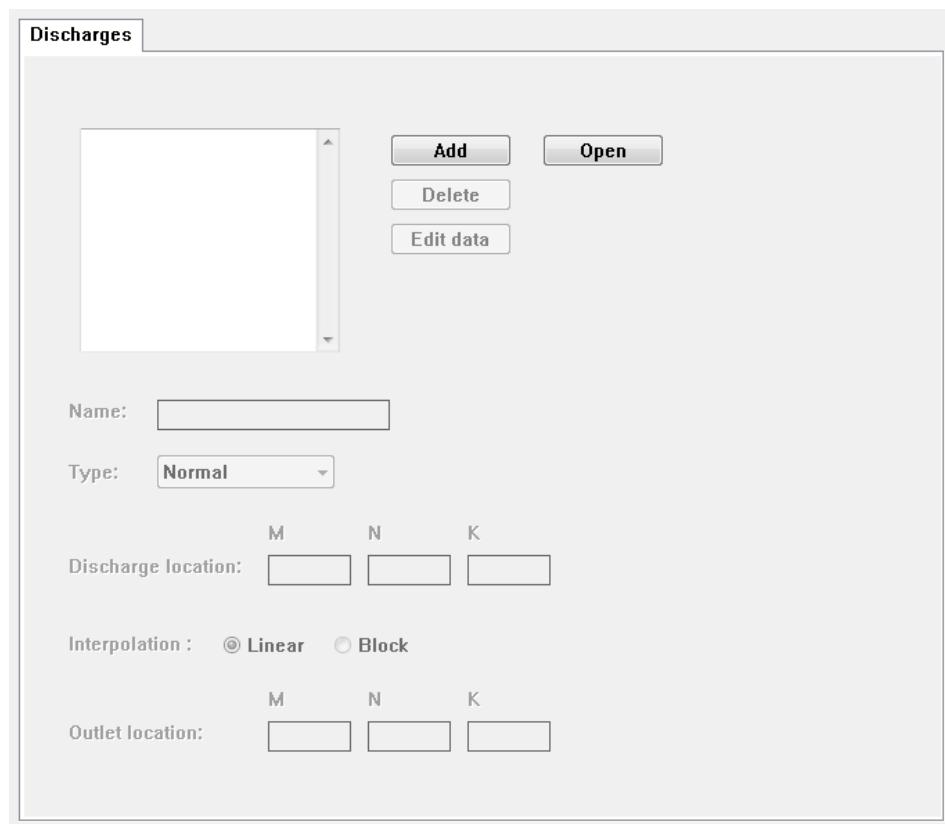


Figure 4.45: Data Group Discharges

Upon selecting the tab **Discharges** the sub-window of [Figure 4.45](#) is displayed.

Remarks:



- ◊ You can model a river as an open boundary or as a discharge.
- ◊ The discharge rate is communicated to the water quality and particle tracking modules WAQ and PART, respectively, but not the concentrations, temperature or salinity. These quantities are computed in the WAQ and PART module in combination with the other water quality processes.

The various input items you can select are discussed below.

Add

To Add a discharge either:

- Click **Add**.
 - Specify the grid indices and the layer of the release point, i.e. (M, N, K). K only if you have a 3D model.
- or
- Use the **Visualisation Area** window.
and

- Specify the name of the discharge in the *Name* input field.

Delete

To Delete a discharge either:

- Select the discharge to be deleted in the list box.
 - Click **Delete**.
- or

➤ Use the **Visualisation Area** window.

Open/Save

Open or save the discharge data to attribute files; the locations to a file with extension <src>, the discharge rates to a <dis> file.

**Remark:**

- ◊ If you want to combine discharges read from an attribute file and inserted manually you must read the file first and then add the manually defined discharges. In reverse order the manually defined discharges are overwritten by those of the file.

Type

The discharge can be of type:

- ◊ *Normal*: the discharge rate is released without taking specific aspects into account.
- ◊ *Momentum*: the momentum of the discharge is taken into account. When prescribing the data the momentum and its direction relative to the North must be specified; for more details see below under specify *v*-direction.
- ◊ *Walking*: the discharge moves to the nearest wet grid cell if the cell in which the discharge is released becomes dry. It moves back to its original position when the grid cell is flooded again.
- ◊ *In-out*: intake and outlet are on different locations. The intake and outlet discharge rates are the same; the concentration of a constituent at the outlet equals the concentrations at the inlet plus the user-defined concentration (when applicable).

Interpolation

The interpolation of time dependent input data can be either

- ◊ *Linear*: intermediate values are determined by linear interpolation.
- ◊ *Block*: the last value is repeated until the next value is defined.

Outlet location

You define the grid indices (M, N, K) of the outlet related to the just defined inlet. These indices are only required and accessible for the discharge *Type = In-out*.

Edit data

To specify the discharge rate and the concentration of the discharged substances. Upon selecting *Edit data* a sub-window is displayed, see [Figure 4.46](#).

The time dependent data are defined at time breakpoints and their intermediate value is determined by either linear or block wise interpolation as defined under *Interpolation*. Depending on the selections made in the Data Group *Processes* you must specify more or less quantities.

In the most complete case you must specify:

Time

Date and time of the time breakpoints.

Flow

The discharge rate in [m³/s].

Remark:

- ◊ A positive rate indicates discharging *into* the model, except for an *In-out* discharge: a positive rate means withdrawal at the intake. The same amount of water is discharged into the model at the outlet.

v-magnitude

The speed with which the discharge is released in [m/s]. Only for discharge *Type = Momentum*.

v-direction

The direction in which the momentum is delivered to the flow. The direction is defined similar to the wind direction, i.e. measured (clockwise) from the North. Only for discharge of *Type = Momentum*.



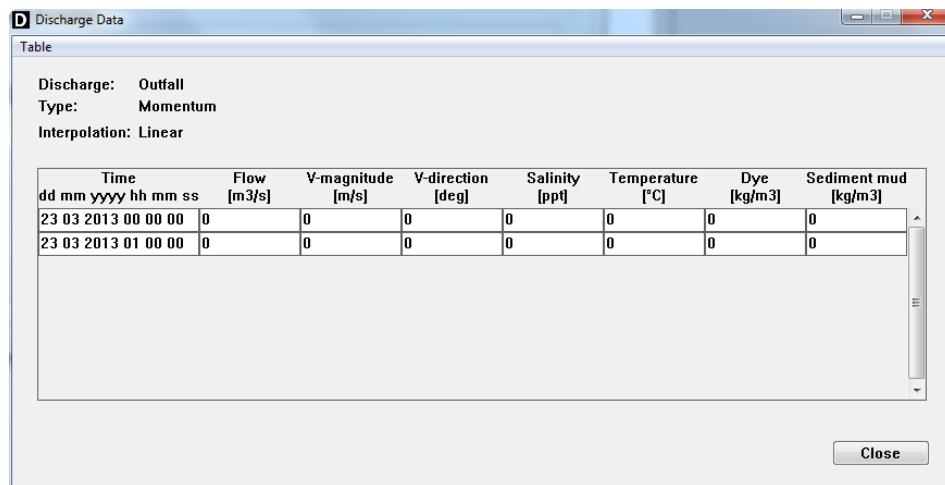


Figure 4.46: Sub-window to define the discharge rate and substance concentrations

The momentum released to the flow due to a discharge is distributed over the velocity points that bound the quadrant in which the discharge is released.

Example: the momentum of a discharge released in the first quadrant as seen from the discharge location (water level point), is decomposed over the u - and v -velocity point with grid indices (m, n) . The momentum of a discharge released in the third quadrant is decomposed over the u - and v -velocity points with grid indices of respectively, $(m - 1, n)$ and $(m, n - 1)$, see [Figure 4.47](#). The mass (of water and constituents) is released in the (water level) point with grid indices (m, n) .

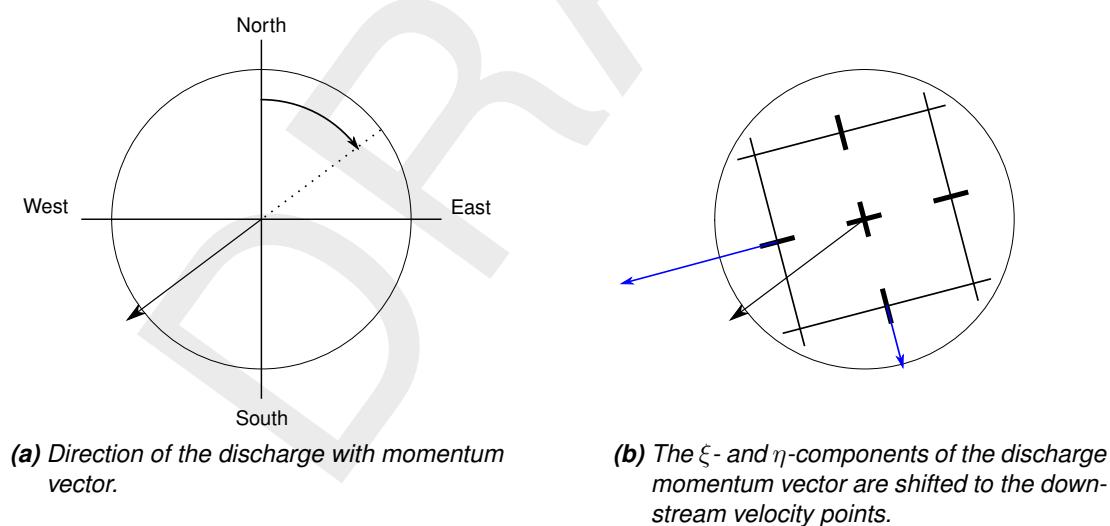


Figure 4.47: Decomposition of momentum released by a discharge station in (m, n)

<i>Salinity</i>	The salinity concentration in [ppt]. Only if <i>Salinity</i> is selected in the Data Group Processes.
<i>Temperature</i>	The temperature of the discharge in $^{\circ}\text{C}$. Only if <i>Temperature</i> is selected in the Data Group Processes.
<i>Pollutants and sediments</i>	The concentration of the constituents released in kg/m^3 . Only if <i>Pollutants</i> and/or <i>Sediments</i> are defined in the Data Group Processes.



Remarks:

- ◊ Layer 1 corresponds to the surface layer.
- ◊ If the discharge is to be distributed uniformly over the layers, you must enter $K = 0$.
- ◊ A positive discharge represents an inflow into the model area; a negative discharge represents an outflow from the area; for *Type* = *Normal*, *Walking* and *Momentum*.
- ◊ For *Type* = *In-out*, a positive discharge represents withdrawal at the intake. The same amount is an inflow at the outlet. Furthermore, the specified concentration is added to the concentration of the water withdrawn.
- ◊ A discharge can be located in an area that becomes temporarily dry during the simulation. In such a case there will be no intake, unless you have specified the discharge to be of *Type* = *Walking*.
- ◊ Walking discharges move to the nearest wet grid cell by looking at the steepest descent of the surrounding bathymetry.

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Location	Anywhere in the grid, within the computational domain and not at the model boundaries.			
Flow	-10,000	+10,000	0.0	m^3/s
<i>v</i> -magnitude	0.0	100.0	0.0	m/s
<i>v</i> -direction	-360.0	360.0	0.0	degree
Salinity	0.0	100.0	0.0	ppt
Temperature	0.0	60.0	0.0	$^\circ\text{C}$
Concentration	0.0	10,000	0.0	kg/m^3

**Restrictions:**

- ◊ The discharge name must be non-blank, and contains up to 20 characters.
- ◊ The discharge must be located within the grid enclosure.
- ◊ The number of discharges is limited to 5000, including the automatically generated discharges when using bubble screens, see [subsubsection B.22.4.2](#).
- ◊ If discharges are specified both through an attribute file and interactively, then the attribute file must be read first. Reversing this order will overwrite the interactively defined discharges.
- ◊ The time breakpoints specified for time-series must differ from the simulation start time by an integer multiple of the time-step (Data Group *Time frame*).
- ◊ The first time breakpoint must be before or equal to the simulation start time, and the last time breakpoint must be equal or later than the simulation stop time (see Data Group *Time frame*).
- ◊ Time breakpoints must be given in ascending order.



Figure 4.48: Sub-data group Dredging and dumping



Figure 4.49: Sub-window for Monitoring locations

4.5.9.2 Dredging and dumping

When sediments are modelled and the man-made process **Dredging and dumping** is switched on (see Data Group Processes), the tab **Dredging and dumping** is available, see [Figure 4.48](#).

- Click *Select file* and specify the file <*.dad> with dredging and dumping data.

Remark:

- ◊ See the QUICKIN User Manual on how to define dredging and dumping areas.



4.5.10 Monitoring

Computational results can be monitored as a function of time by using observation points, drogues or cross-sections. Monitoring points are characterised by a name and the grid indices of its location in the model area.

4.5.10.1 Observations

Observation points are used to monitor the time-dependent behaviour of one or all computed quantities as a function of time at a specific location, i.e. water elevations, velocities, fluxes, salinity, temperature and concentration of the constituents. Observation points represent an Eulerian viewpoint at the results. Observation points are located at cell centres, i.e. at water level points. Upon selecting the data and sub-data group *Monitoring* → *Observations* the sub-window of [Figure 4.50](#) is displayed.

You can apply one or more of the following options:

- | | |
|---------------|--|
| <i>Add</i> | To <i>Add</i> an observation point either: |
| | <ul style="list-style-type: none"> ➤ Click <i>Add</i>. ➤ Replace the name “-Unnamed-” by a useful name in the <i>Name</i> input field. ➤ Specify the grid indices of the observation point, i.e. (m, n).
or ➤ Use the Visualisation Area window. |
| <i>Delete</i> | To <i>Delete</i> an observation point either: |

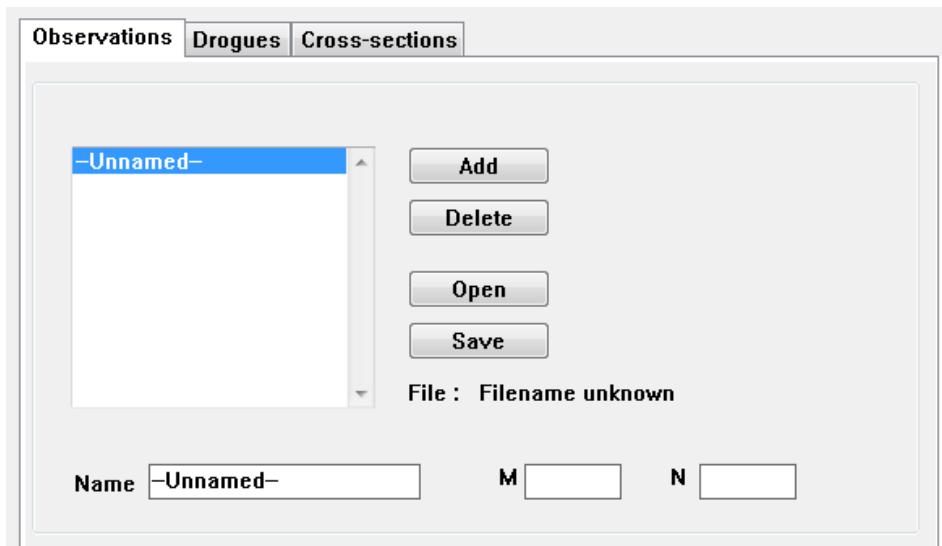


Figure 4.50: Sub-window for Observation points

- Select the observation point to be deleted in the list box.
- Click *Delete*.

or

- Use the **Visualisation Area** window.

Open and read observation points defined in an attribute file.

Remark:

- ◊ If you want to combine observation points read from an attribute file and inserted manually you must read the file first and then add the manually defined observation points. In reverse order the manually defined observation points are overwritten by those of the file.

Save

Save the observation points defined in an attribute file with extension <obs>.



The domain and restrictions for *Observation points*, *Drogues* and *Cross-sections* are very similar and are discussed at the end of this section.

4.5.10.2 Drogues

Drogues are used to monitor the path of a particle moving with the flow. In the horizontal: drogues can be released anywhere in the grid, this means you can release them at fractional grid cell positions. In the vertical: drogues are released in the surface layer. Drogues are only transported due to the velocities in the surface layer. There is no diffusion or random process involved. When wind applies, the wind affects the velocities in the surface and thus the path of drogues. You can release and recover them at any time during the simulation (provided the time complies with the restrictions for time instances).

Upon selecting the data and sub-data group *Monitoring* → *Drogues* the sub-window of [Figure 4.51](#) is displayed.

You can apply one or more of the following options:

Add

To Add a drogue either:

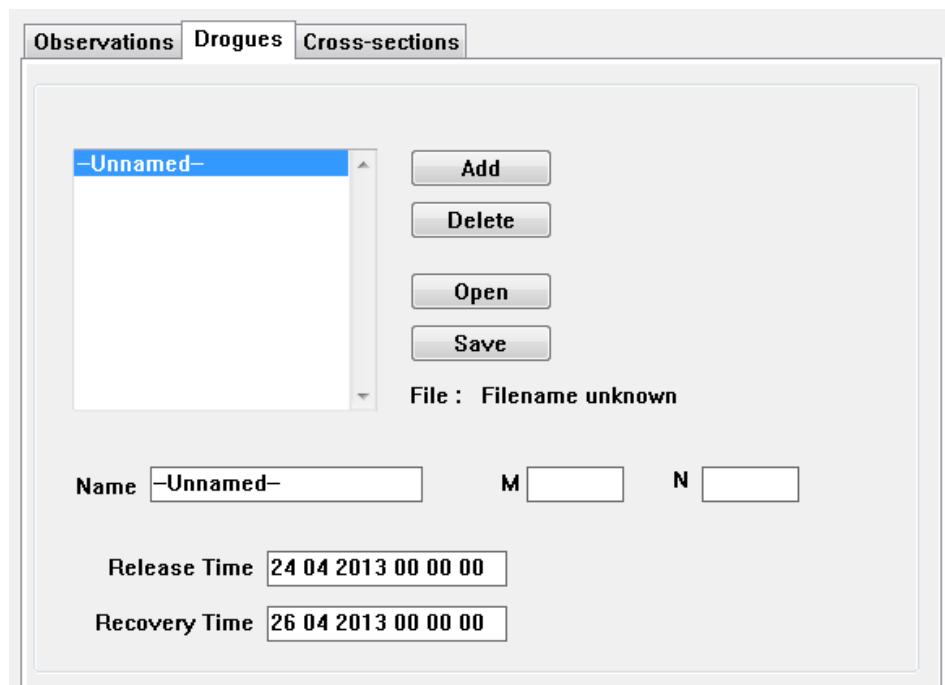


Figure 4.51: Sub-data group Monitoring → Drogues

- Click **Add**.
- Replace the name “-Unnamed-” by a useful name in the **Name** input field.
- Specify the grid indices of the drogue release point, i.e. (m, n) .
or
- Use the **Visualisation Area** window.

Remark:

- ◊ Drogues can be released inside grid cells, i.e. their grid indices M and N can have fractional values. When using the visualisation area you can only specify release points at the grid centres; so if other fractional values are required you have to specify them manually.



Delete

To **Delete** a drogue point either:

- Select the drogue to be deleted in the list box.
- Click **Delete**.
or

- Use the **Visualisation Area** window.

Open

Open and read drogues defined in an attribute file.

Remark:

- ◊ If you want to combine drogues read from an attribute file and inserted manually you must read the file first and then add the manually defined drogues. In reverse order the manually defined drogues are overwritten by those of the file.



Save

Save the drogues defined in an attribute file with extension <par>.

The domain and restrictions for *Drogues*, *Observations* points and *Cross-sections* are very similar and are discussed at the end of this section.

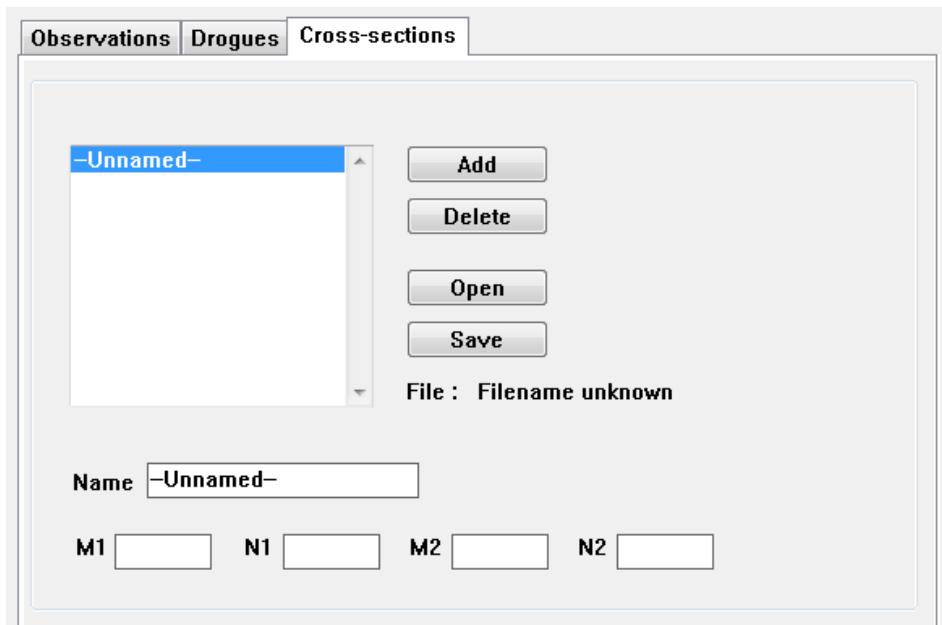


Figure 4.52: Sub-data group Monitoring → Cross-Sections

4.5.10.3 Cross-sections

Cross-sections are used to store the sum of computed fluxes (hydrodynamic), flux rates (hydrodynamic), fluxes of matter (if existing) and transport rates of matter (if existing) sequentially in time at a prescribed interval. A cross-section is defined along a constant m- or n- grid index and it must include at least two grid points. To define a cross-section, you must specify an arbitrary section name, the begin indices (M1, N1) and end (M2, N2) indices of the section.

Upon selecting the data and sub-data group *Monitoring* → *Cross-sections* the sub-window of Figure 4.52 is displayed.

You can apply one or more of the following options:

Add

To *Add* a cross-section either:

- Click *Add*.
 - Replace the name “-Unnamed-” by a useful name in the *Name* input field.
 - Specify the grid indices (M1, N1) and (M2, N2).
- or
- Use the **Visualisation Area** window.

Remark:

- ◊ Cross-sections typically extent over more than just one grid cell, so you have to specify their start and end point.

Delete

To *Delete* a cross-section either:

- Select the cross-section to be deleted in the list box.
 - Click *Delete*.
- or

- Use the **Visualisation Area** window.

Open
Save

Open and read cross-sections defined in an attribute file.

Save the cross-sections in an attribute file with extension <crs>.



If you want to combine cross-sections read from an attribute file and inserted manually you must read the file first and then add the manually defined cross-sections. In reverse order the manually defined cross-sections are overwritten by those of the file.

All cross-sections *must* be saved in an attribute file upon leaving the sub-data group.

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Location	Anywhere in the grid, within the computational domain and not at the model boundaries, i.e. $2 \leq M \leq M_{max}-1$; $2 \leq N \leq N_{max}-1$.			

Restrictions:

- ◊ The release time and the recovery time of drogues must comply with the definition of time parameters.
- ◊ Drogues must be released in an active grid point; this point may become dry during the computation.
- ◊ A cross-section must extend over at least two grid points.
- ◊ If observation points are read from an attribute file and defined interactively, the file must be read first; reversing the order will overwrite the manually defined observation points.
- ◊ When saving the results as time-series, you have to define at least 2 obsservation points.
- ◊ When using cross-sections, you should define at least 2 sections.



4.5.11 Additional parameters

Through the use of an *Additional parameter* you have access to additional functionalities that are not yet fully supported by the FLOW-GUI. The option of *Additional parameters* has been introduced to be more flexible in providing special functionality in the computational code without changing the FLOW-GUI. After thorough testing it will be released to all users. If you have access to an option, you can specify the required parameters in a separate input file.

Upon selecting this data group [Figure 4.53](#) is displayed.

Each *Additional parameter* is characterised by a keyword and a value:



Remarks:

- ◊ Keywords are CASE-sensitive!
- ◊ The value is often the name of an attribute file and must be enclosed between #'s.

See [Appendix B](#) for full details of the currently available Add-ons, their keywords, values, file format and file contents.

Example:

Keyword	Value
Fillwl	#ex-001.lwl#
Iter	4

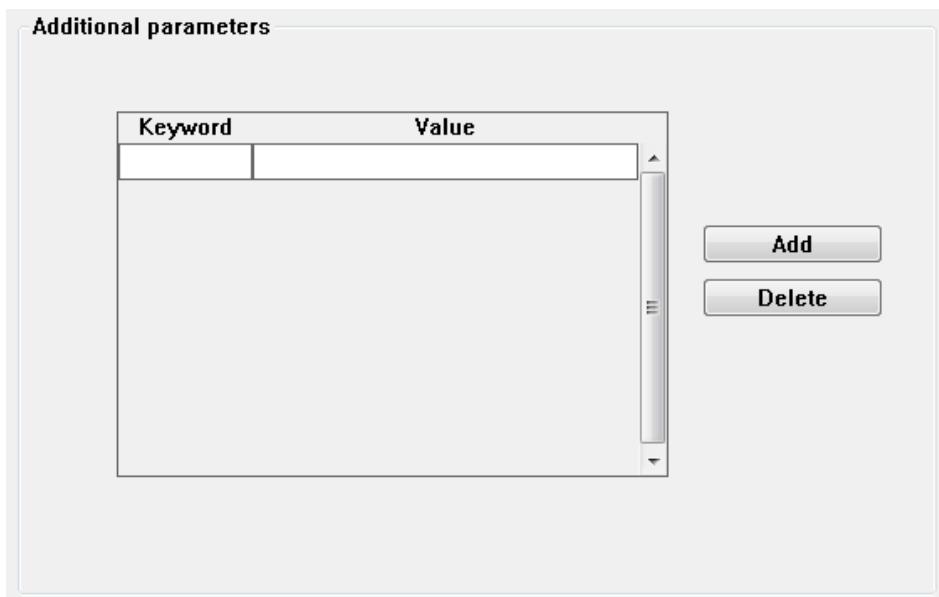


Figure 4.53: Data Group Additional parameters

The first keyword refers to local weirs and the value is the filename in which the location and other properties of the local weir(s) are defined; for details see [section B.3](#).

The second keyword redefines the number of iterations used when computing the continuity equation to be 4 (instead of the default value 2); for details see [section A.1](#).

4.5.12 Output

In the Data Group *Output* you can specify which computational results will be stored for further analysis or for other computations and which output shall be printed. Though the printing option is hardly used it is a useful option when numerical output values are required.

Before defining the various input fields we want to explain in short the use of the various data files used to store the results.

Delft3D modules use one or more Map, History, Communication and Restart files to store the simulation results and other information needed to understand and interpret what is on the files:

Maps are snap shots of the computed quantities of the entire area. As you (can) save all results in all grid points a typical Map file can be many hundreds of MB large. So typically Map results are only stored at a small number of instances during the simulation.

In a History file you store all results as function of time, but only in the specified monitoring points. The amount of data is usually much smaller than for a Map file, and you typically store history data at a small time interval to have a smooth time function when plotting the results.

In the Communication file you store data required for other modules of Delft3D, such as the hydrodynamic results for a water quality simulation or the wave forces for a wave-current interaction. As the results must be stored in all grid points a Communication file can typically be as large as or even larger than a Map file. So, you only store the results in the Communication file as far as needed for the other simulations. For water quality simulations you typically store the results of the last day or a (couple of) tidal cycles and use this data cyclic to make a water

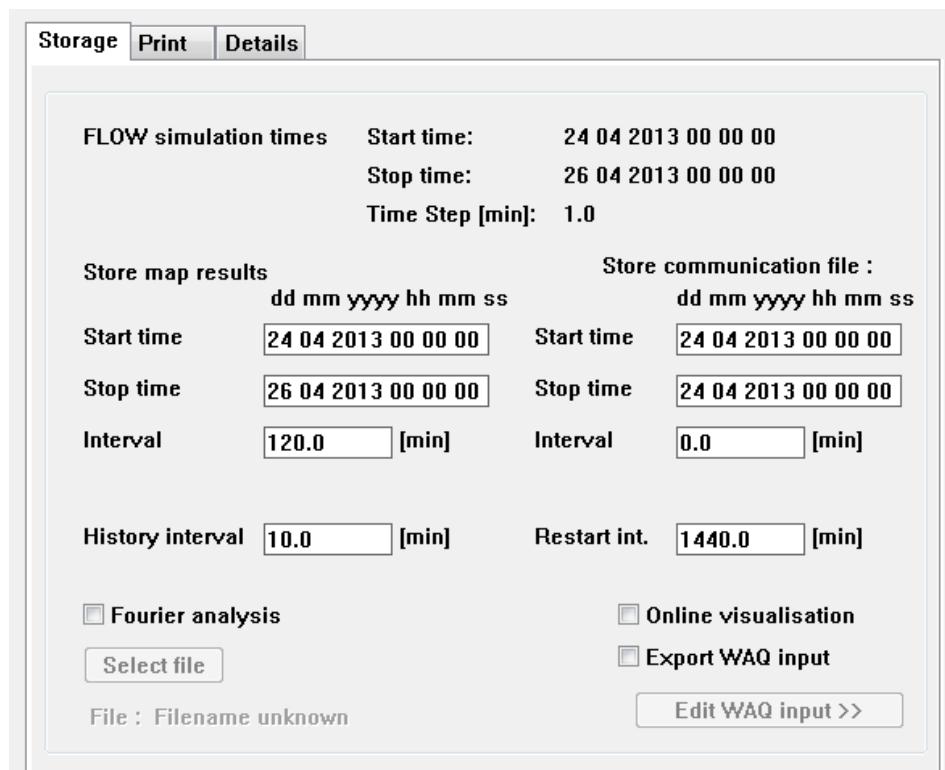


Figure 4.54: Sub-data group Output storage

quality simulation over many days or weeks.

In order not to waste computing time you can restart a simulation at a predefined time of a previously executed simulation. For this you specify the restart interval at which all information required to restart the simulation, are stored in a so-called Restart file. Typically, you select a large restart interval in order not to waste disk space.

Remark:



- ◊ The restart file is *not* platform independent. This means you can not use directly, i.e. without some kind of conversion, a restart file from a simulation executed on another hardware platform. You can also restart from a map-file which is platform independent.

➤ Click the Data Group *Output*.

The data group *Output* is organised in three sub-data groups:

- ◊ Storage, to define the time intervals to various file used.
- ◊ Print, to define if, when and which data will be printed.
- ◊ Details, to make a selection of all possible data to be stored in the files or to be printed.

4.5.12.1 Storage

The sub-data group *Output storage* is displayed in [Figure 4.55](#).

In the top of the **Output storage** sub-window the simulation start and stop time and the time-step used in the simulation are displayed for your convenience, but they cannot be changed in this data group (see the Data Group *Time frame*).

You can apply one or more of the following options:

Store map results

<i>Start time</i>	Date and time to start storing the results in the Map file.
<i>Stop time</i>	Date and time to stop storing the results in the Map file.
<i>Interval</i>	Time interval, in minutes, to write the Map file.

Store history results

<i>History interval</i>	Time interval, in minutes, to write the History file for the whole simulation.
-------------------------	--

Store communication file

<i>Start time</i>	Date and time to start storing the results in the Communication file.
<i>Stop time</i>	Date and time to stop storing the results in the Communication file.
<i>Interval</i>	Time interval, in minutes, to write the Communication file.

Write restart files

<i>Restart interval</i>	Time interval, in minutes, to write the Restart file.
-------------------------	---

Perform Fourier analysis

<i>Fourier Analysis</i>	To activate the Fourier analysis on computational results. Upon ticking off this option you can select a file with the time period and the frequencies you want to apply, see below for a short description and subsection A.2.30 for details of the file format.
-------------------------	---

For large scale tidal flow models, Delft3D-FLOW is equipped with a facility to perform online Fourier analysis on computational results. This enables the generation of computed co-tidal maps. These maps can be compared with co-tidal maps from literature, a very convenient way to check if the tidal wave propagation is simulated correctly by the model. Upon selecting *Fourier analysis* you specify the time period to use, the variable(s) to analyse, the layer number, the frequencies to use and some other more special items (see [Appendix A](#)). When you specify a zero frequency the minimum, mean and maximum level of the variable is computed in the model area over the specified time period. You can select one or more of the following parameters:

- ◊ Water level.
- ◊ Velocity components.
- ◊ Mass fluxes of water.
- ◊ Temperature.
- ◊ Salinity.
- ◊ Constituent(s).

This option creates by default output in ascii format. The output format can be switched to NetCDF by means of the `F1Ncdf` keyword under *Additional parameters*. For spherical models we recommend the NetCDF format since the latitude and longitude are written to the ascii file with only three digits behind the decimal point.

View results during the simulation

Online visualisation When selecting this option you can inspect the results during the computation.

The option of *Online visualisation* lets you monitor the results during the simulation as an animation. You can select the type of parameter displayed, the type of view used, such as iso-lines and vector plots in plain view per layer, or as cross-section, and per layer. You can store the results to file and run this file afterwards as an animation. During the simulation you can change your selection of parameter(s) displayed. Online animation is a very powerful tool to let you inspect the behaviour of your model; this will help you to see anomalies in the results much better and in a much earlier stage than when using plots and figures.

Built-in coupling with D-Water Quality

Export WAQ input When continuing with the Water Quality modules, you can either couple the com-file after the FLOW simulation has ended, or do it via the built-in option.

The D-Water Quality input files can either be generated using a coupling program that runs concurrently with the Delft3D-FLOW simulation or they can be written by Delft3D-FLOW directly. The latter approach is the preferred approach (i.e. the built-in option). See [section B.18](#) for details of the file format.

Export WAQ input window

In the top of the **Output storage** sub-window, frame *WAQ simulation times* ([Figure 4.55](#)), the start and stop time and the time-step used for hydro-morphodynamic simulation are displayed for your convenience, but they cannot be changed in this window (see the Data Group *Time frame*).

Store results for WAQ

To store results for use in D-Water Quality you can edit the following options:

<i>Start time</i>	Date and time to start storing the results in files suitable for D-Water Quality.
<i>Stop time</i>	Date and time to stop storing the results in files suitable for D-Water Quality.
<i>Time step [min]</i>	Time interval, in minutes, to write files suitable for D-Water Quality.

In the frame *WAQ aggregation* you can specify the horizontal aggregation and the vertical aggregation.

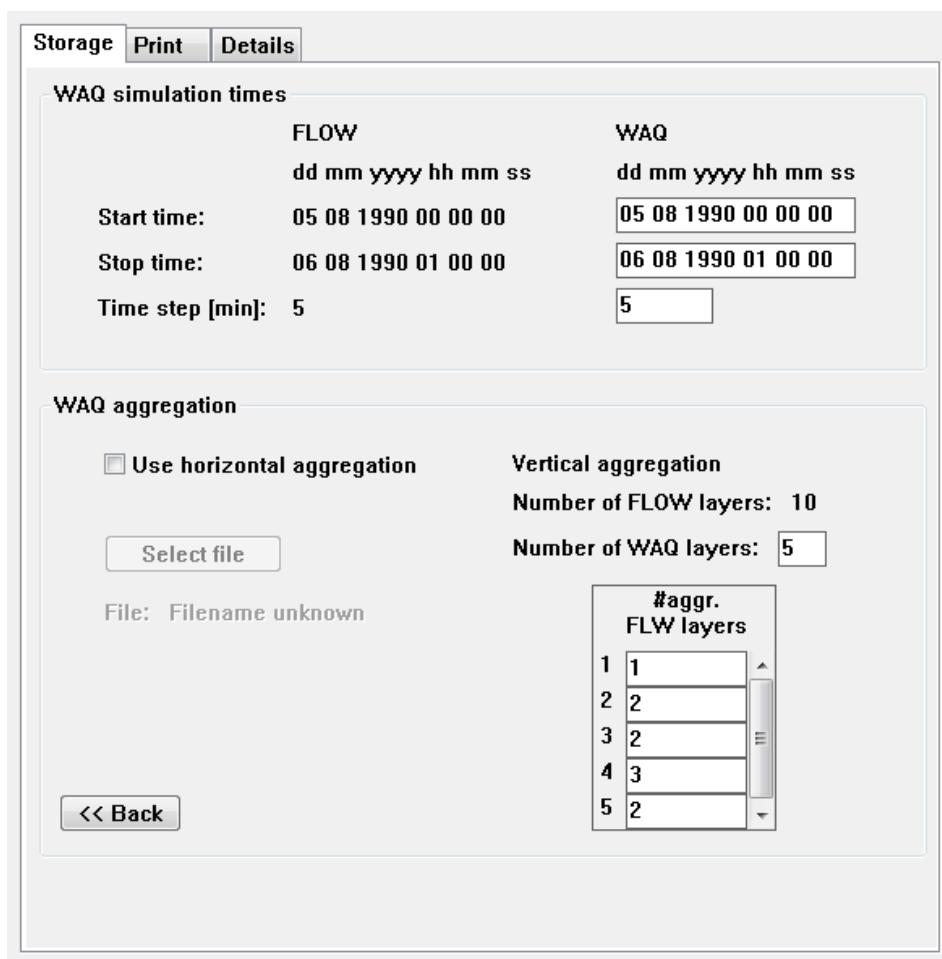


Figure 4.55: Sub-data group Output → Storage → Edit WAQ input

Horizontal aggregation

Horizontal aggregation When selecting this option you can select a file defining the horizontal aggregation. This file can be made with the program D-Waq DIDO.

To make a horizontal aggregation with the program D-Waq DIDO, save first the MDF-file without a check on *Use horizontal aggregation*. Now a hydro-morphodynamic simulation without horizontal aggregation could be made. Start the program D-Waq DIDO and load the just saved MDF-file, define your horizontal aggregation (see [DIDO UM \(2013\)](#)). Save the aggregation file in D-Waq DIDO and select this file in the FLOW-GUI. Go to data group *Output* → *Storage* → *Edit WAQ input*, check mark *Use horizontal aggregation* and select the just saved aggregation file.

Vertical aggregation

For convenience the number of layers in the hydrodynamic simulation is given. You are able to specify the number of layers needed for the D-Water Quality simulation. If this number is less than the number of hydrodynamic layers you have to specify which layers should be aggregated to one WAQ layer.

Example: In the example 5 WAQ layers with aggregation “1 2 2 3 2” in table *# aggr. FLW layers* are specified, which means that hydrodynamic layers 2/3, 4/5, 6/7/8 and 9/10 are aggregated (see [Figure 4.55](#)).

Remarks:

- ◊ No Map, History, Communication or Restart file is written if you set the interval to write the data equal to zero. When verifying the MDF-file a warning will be issued that the time definition is not correct and that the corresponding file will not be created. If you did not want to create this file, you can ignore this warning.
- ◊ By default the start time to store the hydrodynamic results is used as the simulation start time of the far- and mid-field water quality scenarios using these hydrodynamic results.
- ◊ Drogues are written to a separate drogue-file for each time step between the release and recovery time. You can not specify the interval to write the drogue-file.



Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Date and time	Reference date		Simulation date and time	Valid date and time
Interval	0.0		0.00	minutes

Restriction:

- ◊ The start and stop time must differ an integer multiple from the *Reference date* and the time interval for storing results must be an integer multiple of the *Time step*, both specified in the Data Group *Time frame*.



4.5.12.2 Print

In the sub-data group *Output* → *Print* you can specify which results will be printed. The selected results are printed to a file; the name of the file is composed using the prefix of the name of the MDF-file: <tri-prt.runid>.

Upon selecting the sub-data group *Output* → *Print* a sub-window is displayed, see [Figure 4.56](#).

You can apply one or more of the following options:

Print history results

- | | |
|-------------------|---|
| <i>Start time</i> | Date and time to start writing the history data to a print file. |
| <i>Stop time</i> | Date and time to stop writing the history data to a print file. |
| <i>Interval</i> | Time interval, in minutes, to write the history data to a print file. |

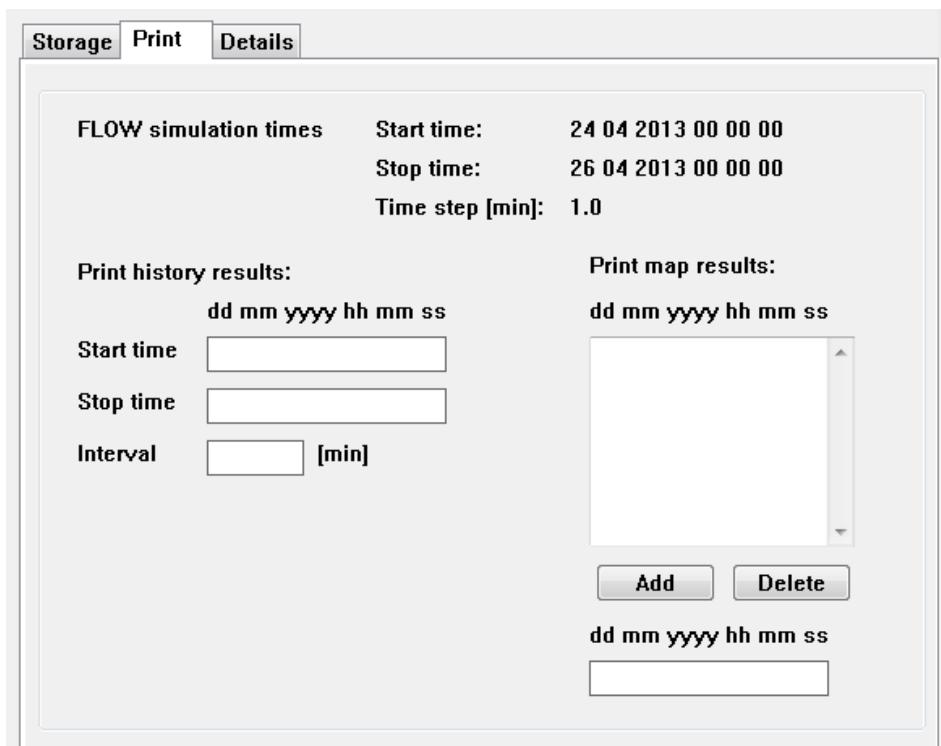


Figure 4.56: Sub-data group Output → Print

Print Map Results

Add

Add the map time, specified in the input field below the list box, to the list of time instances to be written to the print file.

Delete

Select a time instance from the list box and press *Delete* to remove a time instance from the list box.

A warning is issued if the maximum number of time instances is reached. The print times may be specified in ascending order; the computed water elevations, flows and transported quantities at all points are stored in an ASCII-formatted file for printing.

4.5.12.3 Details

In the sub-data group *Output → Details* you can make a selection of all possible data to be stored in the files or selected for printing. Upon selecting the sub-data group [Figure 4.57](#) is displayed.

By ticking the check boxes you can specify which items you want the program to store on file or to print and which not.

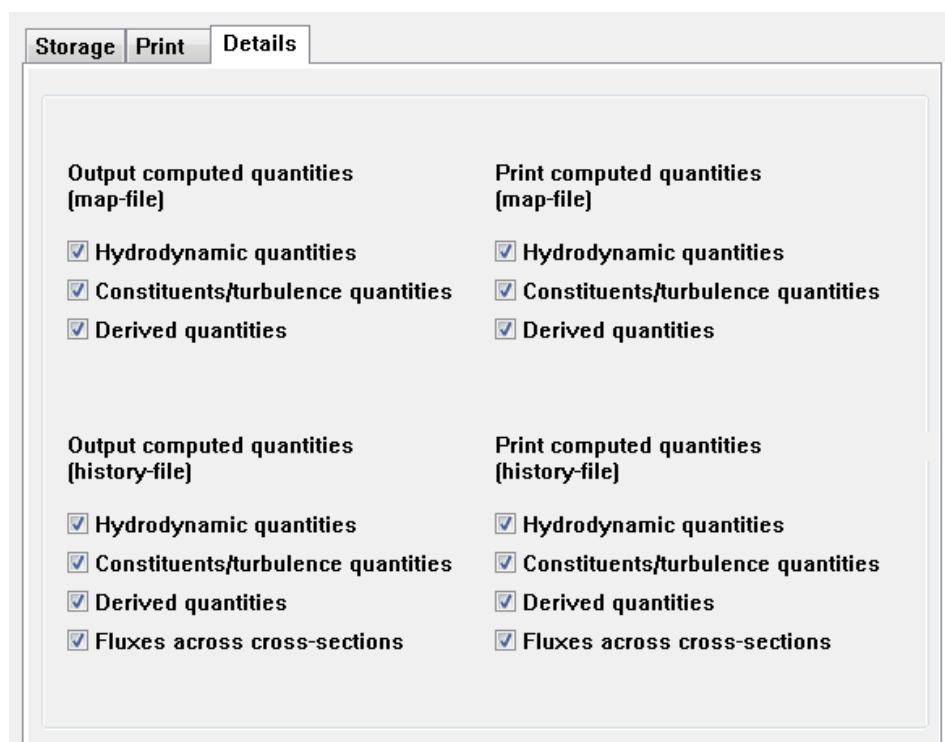


Figure 4.57: *Output Specifications* window

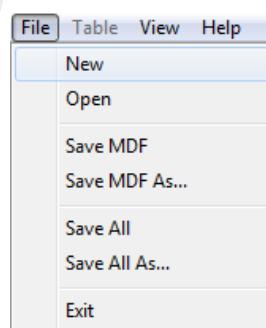


Figure 4.58: File drop down menu

4.6 Save the MDF and attribute files and exit

To save the MDF-file or any of the attribute files:

- Click *File* in the menu bar of the **FLOW-GUI** window and select any of the *Save* or *Save As* options, see [Figure 4.58](#).

For saving the files and exiting the FLOW-GUI only the last five options in the file drop down menu are relevant.

Select:

- *Save MDF* to save the MDF-data under its current name.
- *Save MDF As* to save the MDF-data under a new name.
- *Save All* to save all attribute data in the current attribute files.
- *Save All As* to save all attribute data under a new name.
- *Exit* to exit the FLOW-GUI and return to the **Hydrodynamics** window.

Upon selecting *Save MDF* or *Save MDF As* the MDF-file is saved with a reference to the attribute files as they existed at the time you started the edit session or after the last save command. Changes to attributes are ignored if you have not explicitly saved them. These *Save MDF* and *Save MDF As* options allows you to ignore some or all of the changes you have made to some or all attribute data and restore the last saved situation. The non-saved changes are not lost, as they are still available in the input fields of the FLOW-GUI.

If you want to save the changes to the attribute files as well, you must select *Save All* or *Save All As*. Existing files are overwritten when you select *Save All*; you are requested to specify a filename if the attribute file does not yet exist. With *Save All As* you must specify a (new) filename for all the files, including the MDF-file, which must be saved.

To exit the FLOW-GUI:

- Click *File* → *Exit*.

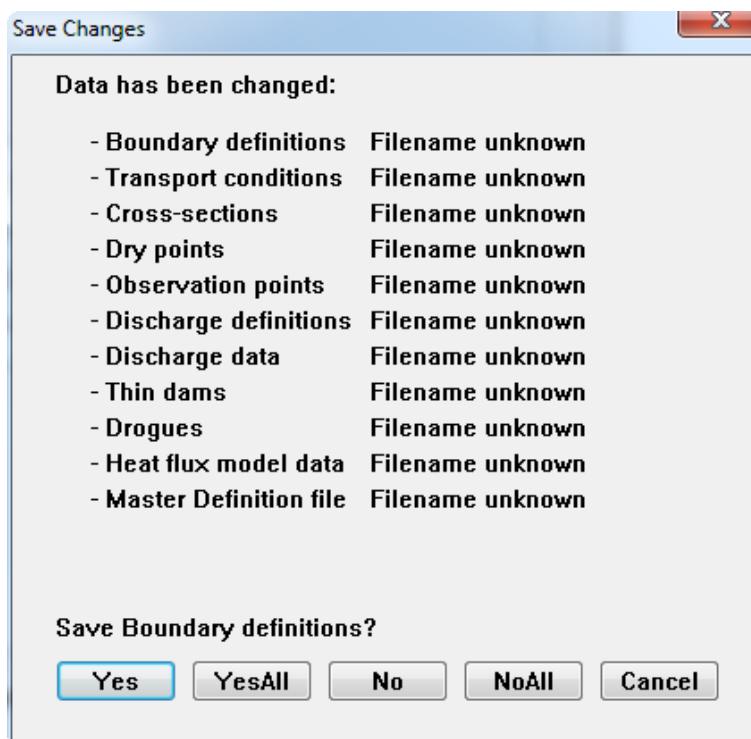
If all data items, including the MDF-file, have been saved, the FLOW-GUI is closed and the **Hydrodynamics (including morphology)** window of Delft3D-MENU is redisplayed.

However, if not all data items have been saved yet a list of all non-saved (attribute) data will be displayed and you can select for each non-saved data item the required action, see [Figure 4.59](#). In this case none of the data items nor the MDF-file itself has been saved yet.

Select one of the options displayed:

<i>Yes</i>	Save the first unsaved data item, request for a filename if not yet defined and proceed to the next unsaved data item.
<i>YesAll</i>	Save all unsaved data items and request a filename if not yet defined.
<i>No</i>	Don't save the first unsaved data item; proceed to the next unsaved data item.
<i>NoAll</i>	Exit without saving any unsaved data.
<i>Cancel</i>	Abort this <i>Exit</i> action and return to the FLOW-GUI.

After you have worked through all unsaved data items the FLOW-GUI will be closed and the **Hydrodynamics (including morphology)** window of Delft3D-MENU will be displayed.

**Figure 4.59: Save changes window**

4.7 Importing, removing and exporting of data

When starting a scenario definition session from an existing MDF-file, all attribute files are imported automatically. If the reference to an attribute file does not exist, or if you wish to apply a different attribute file you can apply the *Open* or *Select file* option.

Warnings:

- ◊ If you want to combine attribute data from an existing file with manually defined data, you must open the existing data before you add new data items.
- ◊ If you want to concatenate data items from two files you must do so offline with your favourable editor.



The reference to an empty attribute file (because you have deleted all data items) is automatically removed from the MDF-file upon saving the MDF-file.

Most attribute files are ASCII-files and free formatted; you are referred to [Appendix A](#) for details on their format and contents.

DRAFT

5 Tutorial

5.1 Introduction – MDF-file and attribute files

In this Tutorial we will guide you through the process of creating a simple example of a flow simulation. All information for a flow simulation, also called scenario, is stored in an input file, also known as Master Definition Flow file (MDF-file). However, before starting this input definition process we want to explain in short the basics of a model definition, the structure of an MDF-file and the basic steps you are supposed to execute.

To execute a flow simulation for a specific area we need various kinds of information, such as the extent of the model area, (i.e. both the boundary between wet and dry areas and the location of the open boundaries where water level or flow conditions are prescribed), the bathymetry, geometrical details of the area such as breakwaters, structures, discharges and the definition of which and where results of the simulation need to be stored for later inspection or for other simulations. Finally, a numerical grid must be defined on which all location related parameters are being defined. So, the basic steps that precede the definition of an input file can be summarised as:

- ◊ Selection of the extent of the area to be modelled.
- ◊ Definition of location and extent of open boundaries and the type of boundary conditions to be prescribed, i.e. water level, velocities or discharges.
- ◊ Definition of the land(-water) boundary.
- ◊ Generation of a numerical grid.
- ◊ Generation of the bathymetry defined on the numerical grid.
- ◊ Definition of many different grid related quantities, such as open boundaries, monitoring points, discharge locations, release points of drogues.
- ◊ Definition of the time frame of your scenario, i.e. start and stop time and various time functions, such as the open boundary conditions, wind speed and direction, discharges and salinity concentrations or other substances transported by the flow.

Most activities, but the last two, must be executed before starting the FLOW Graphical User Interface (GUI). They result in most cases in one or more files that are to be located in the project directory to be defined when starting the project (see [chapter 8](#)). The project directory is also referred to as the working directory. The first two steps are based on experience in solving similar problems and on engineering judgement, no tools are available to support these steps others than (GIS-based) maps and (digitised) charts.

Delft3D supports the use of a rectilinear, a curvilinear and a spherical grid. Rectilinear and spherical grids are generated automatically given the grid dimensions and sizes, and no tool is required. To generate a curvilinear grid you can use the grid generator program RGFGRID, provided with Delft3D. To generate a curvilinear body fitted grid you need a file with the land boundary and preferably the bathymetry. The land boundary is obtained from digitising a map or from GIS and the bathymetry is obtained upon using the tool QUICKIN. For details you are referred to the User Manuals of RGFGRID and QUICKIN. For the file formats of the land boundary, the bathymetry and other files, see [Appendix A](#). For this tutorial these files are provided.

The data of the land boundary, the bathymetry and the numerical grid, but also information on boundary conditions, discharges and grid related quantities such as monitoring points, discharges etc. are stored in separate, so-called attribute files. In the MDF-file only a reference is made to these files instead of including all data in the MDF-file itself. The big advantage of using attribute files is that the data can be used in many scenarios but is stored only once on the system disks.

In this tutorial we provide an existing MDF-file with attribute files for a specific example called ‘fti’. The area modelled concerns the entrance between two islands in the northern part of The Netherlands and is called “Friesian Tidal Inlet”. On the outer side of the area the flow is driven by a semi-diurnal tide, while the inner side the area enclosed by land. We use this basic example to guide you through most of the input definition part of a flow simulation.

To create or modify an MDF-file you use the FLOW-GUI you became acquainted with in [chapter 3](#). Furthermore, two attribute files are provided for use in this tutorial, viz. the curvilinear grid file called <fti_02.grd> and the grid enclosure file <fti_02.enc>. The grid enclosure describes the extent of the numerical model; points outside the grid enclosure will not take part in the simulation (are outside the open boundaries or are always on-land). While using the FLOW-GUI you will create new attribute files which contain data like locations of monitoring points or time-series needed for the computation. The complete MDF-file and all attribute files created in this tutorial are also provided in the Tutorial directory, but instead of using them you will build your own attribute files and MDF-file.



Remarks:

- ◊ The choices you are going to make are not based on physical relevance or other practical considerations, but merely to show a number of features of the FLOW-GUI and to make you familiar with it.
- ◊ You will start with a new (empty) input file, so all input fields when displayed are empty or only have default values. But, the screens displayed in this tutorial display the situation *after* you have typed in the required data.

5.2 Filenames and conventions

The names of the MDF-file and of attribute files have a specific structure, some aspects are obliged while others are only advised or preferred.

MDF-file

The name of an MDF-file must have the following structure: <*name.mdf*>. The *name* may consist of up to 252 alpha-numeric characters and may contain (sub-)directories: <*path\runid*>. The *runid* of the filename is used in the names of the result files to safeguard the link between an MDF-file and the result files. We suggest to use a combination of one alpha- and two numeric-characters followed by a useful name of your project.

Example: <c:\projects\panama\r01-calibration.mdf>

This file could indicate the flow-input file of the first calibration run of a project named Panama in the directory <c:\projects>.

The extension <mdf> is mandatory.

Table 5.1: Overview of attribute files.

Quantity	Filename and mandatory extension
Astronomic correction factors	<name.cor>
Bathymetry or depth	<name.dep>
Bottom roughness	<name.rgh>
Constituents boundary conditions	<name.bcc>
Cross-sections	<name.crs>
Curvilinear grid	<name.grd>
Discharge locations	<name.src>
Discharges rates	<name.dis>
Dredge and dump characteristics	<name.dad>
Drogues or floating particles	<name.par>
Dry points	<name.dry>
Flow boundary conditions (astronomic)	<name.bca>
Flow boundary conditions (harmonic)	<name.bch>
Flow boundary conditions (QH-relation)	<name.bcq>
Flow boundary conditions (time-series)	<name.bct>
Fourier analysis input file	<name.fou>
Grid enclosure	<name.enc>
Horizontal eddy viscosity and diffusivity	<name.edy>
Initial conditions	<name.ini>
Morphology characteristics	<name.mor>
Observation points	<name.obs>
Open boundaries	<name.bnd>
Sediment characteristics	<name.sed>
Temperature model parameters	<name.tem>
Thin dams	<name.thd>
Wind	<name wnd>

Result files

The results of a Delft3D-FLOW computation are stored in four types of files:

- ◊ History file: <trih-runid.def> and <trih-runid.dat>.
- ◊ Map file: <trim-runid.def> and <trim-runid.dat>.
- ◊ Drogues file: <trid-runid.def> and <trid-runid.dat>.
- ◊ Communication file: <com-runid.def> and <com-runid.dat>.

The result files are stored in the working or project directory.

The history file contains results of all computed quantities in a number of user-defined grid points at a user-defined time interval. The map-file contains results of all quantities in all grid points at a user-specified time interval. The drogues file contains the positions of all drogues released as a function of time. The communication file contains results that are required by other modules, such as the water quality module.

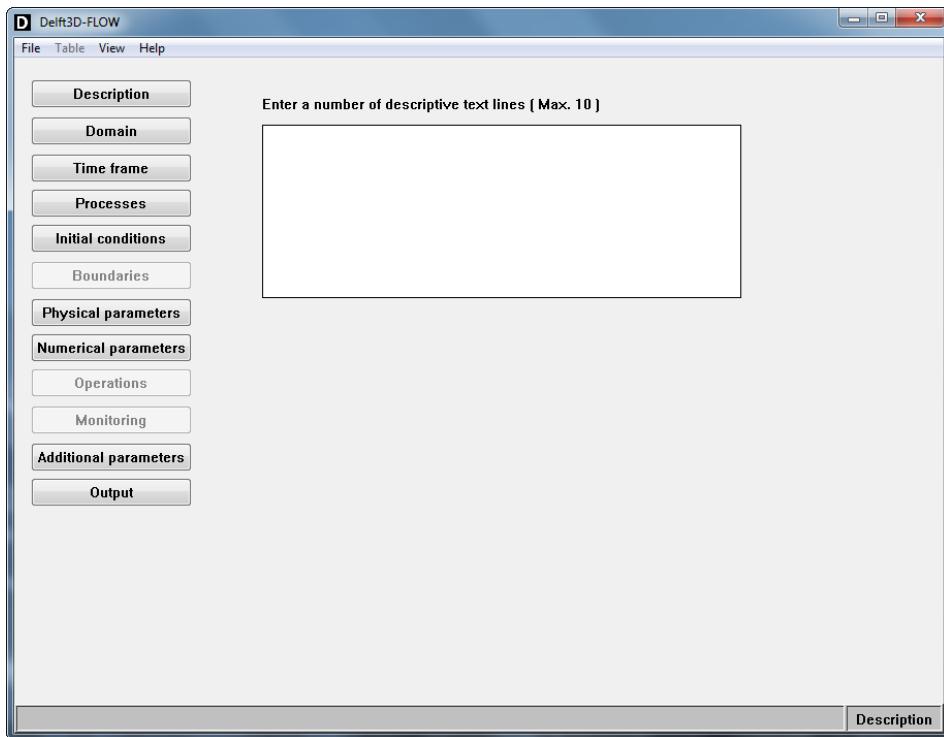


Figure 5.1: Starting window of the FLOW Graphical User Interface

Attribute files

Attribute files contain certain input quantities, such as monitoring points or time dependent input data such as wind. The names of attribute files are basically free, but their extension is mandatory as indicated below.

5.3 FLOW Graphical User Interface

5.3.1 Introduction

To start the FLOW Graphical User Interface (GUI) you must execute the following commands, see [chapter 3](#) for details:

- Click the Delft3D-MENU icon on the desk-top (Windows) or execute the `delft3d-menu` command on the command line (Linux).
- Select the menu item *FLOW*, see [Figure 3.2](#).
- Change, i.e. go to the working directory `<..\\tutorial\\flow\\friesian_tidal_inlet>`.
- Select the menu item *Flow input*, [Figure 3.3](#).

Next the main window of the FLOW-GUI is displayed, see [Figure 5.1](#).

To open an existing input file:

- Select *File → Open*.

By selecting an existing MDF-file all attribute files and their related data are loaded into memory and displayed in the input fields.

Instead, you are going to define all input data from scratch and save your data into an input file called `<tut_fti.mdf>`.

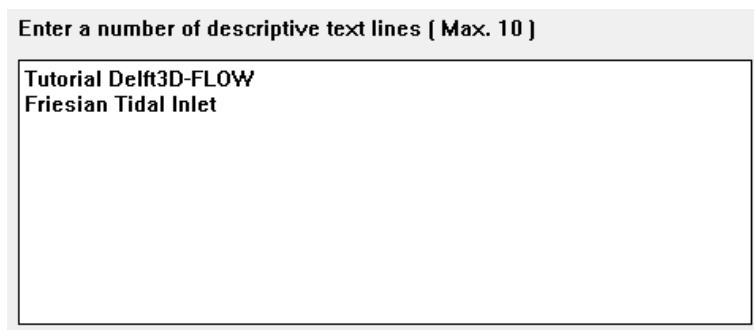


Figure 5.2: Data Group Description sub-window

If you have selected *File* → *Open* you can return to the initial situation right after starting the FLOW-GUI by selecting *File* → *New*. This will reset the internal data structure.

You are now ready to start defining your own scenario, but before doing so you will learn how to save your input data, even if the input data is incomplete.

5.3.2 Saving the input data

Initially, this tutorial may be somewhat tedious to work with. Rather than going on until the end you may want to stop somewhere during the exercise. In order to save you from re-entering again all the data another time up to the moment where you had stopped, you should save the just entered data, so that later on you can pick up the thread where you left it.

To save your data:

- Select *File* → *Save All*. This option is enabled when necessary.
- For each Data Group with changed data for which you have not yet defined an attribute filename and for the MDF-data (if you have not yet defined an MDF-filename) a file selection window will be opened, enabling you to save the unsaved data in an existing or in a new file.
- Next select *Exit* to close the FLOW-GUI or continue with your data definition.

Neglect the other options of the *File* drop down menu; they will be discussed later on.

5.4 Description

In the Data Group *Description* you can identify the scenario you are going to define by giving a comprehensive description of the project, the application domain and the specific selections to be made in this scenario. The description is only used for identification and has no influence on the simulation itself. Type the description as displayed in, [Figure 5.2](#).

5.5 Domain

The Data Group *Domain* supports four sub-data groups:

- ◊ *Grid*, to define the numerical grid.
- ◊ *Bathymetry*, to define the bottom topography.
- ◊ *Dry points*, to define grid points that are excluded from being flooded during a simulation.
- ◊ *Thin dams*, to define dams along grid lines that block the flow but have no influence on the storage.

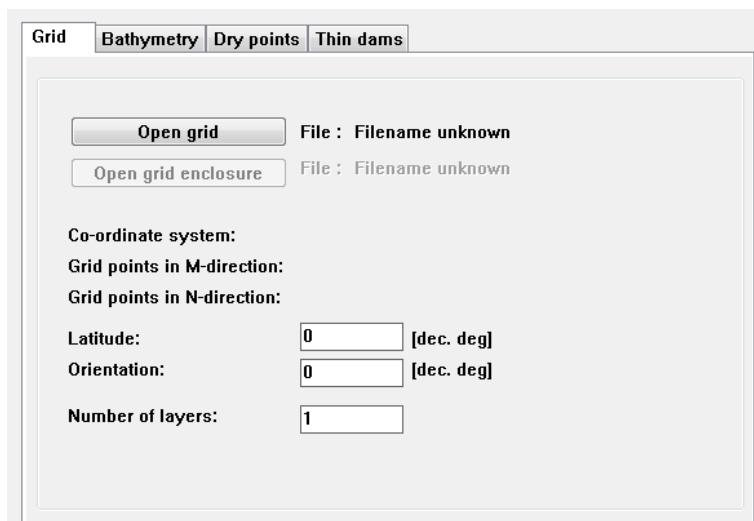


Figure 5.3: Sub-data group **Grid**

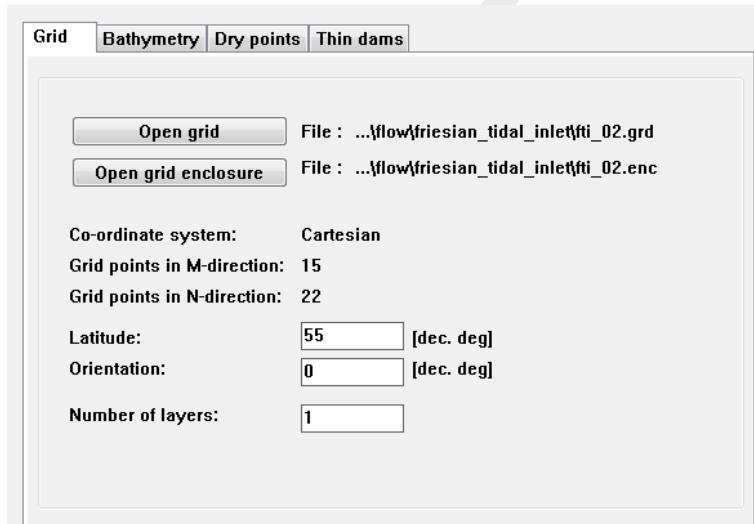


Figure 5.4: Sub-data group **Grid**; filenames, type of co-ordinate system and grid dimensions

5.5.1 Grid parameters

In the sub-data group *Grid* you can open the grid to be used, its location and the number and thickness of the vertical layers.

- Click on *Domain* and then on *Grid*.

The resulting window is given in [Figure 5.3](#).

There are two *Open* buttons, one for the grid and one for the grid enclosure.

- Click *Open grid*. A file selection window is displayed.
- Select the folder <..\\tutorial\\flow\\friesian_tidal_inlet> if not already selected at the start of this tutorial, see [subsection 5.3.1](#).
- Select the file <fti_02.grd> and open it.

The file selection window is closed and the filename <fti_02.grd> appears to the right of the *Open grid* button, see [Figure 5.4](#).

Furthermore, the type of co-ordinate system (Cartesian or Spherical) is displayed together with the grid dimensions. This information is contained in the grid file.

- Repeat the same actions for the grid enclosure file <fti_02.enc>.
- Set the *Latitude* to: "55.00" [degrees N].

Orientation is the angle (clockwise is positive) between true North and the *y*-direction; its default value is zero and can be left unchanged. For the current purpose the grid is ready for use.

Remarks:



- ◊ You **cannot** modify the Grid Dimensions after loading a grid file. The dimensions of the grid Mmax (number of grid points in ξ -direction) and Nmax (the number of grid points in η -direction) have been set (in this case to 15 and 22) in the grid generation program and cannot be changed. However, you can still select the number of vertical layers. Maintain the currently inserted value 1 (for a depth averaged computation).
- ◊ Many of the data items to be specified in the coming windows are related to and defined at a numerical grid. It is therefore obvious *first* to select or define the grid you are going to use.
- ◊ A warning will be displayed when you try to define a grid related quantity before you have defined your grid.

Before you continue with the other sub-data groups, you should visualise and inspect the area to be modelled and the numerical grid used.

- Click the item *View* in the menu bar and select the item *Visualisation Area* in the drop down list. This will result in the window as shown in [Figure 5.5](#).
- Move the **Visualisation Area** window to the right of the screen, to keep it (at least partly) uncovered.

In [Figure 5.5](#) the numerical grid is shown with the active grid cells enclosed by the grid enclosure. In the **Visualisation Area** window all grid related quantities can be viewed, defined, moved and deleted, but as no quantities are yet defined only the grid is displayed. Select the menu items in the menu bar of the **Visualisation Area** window to inspect the various options. For the tutorial we leave the **Visualisation Area** window, this window is described in more details in [chapter 4](#).

Before continuing with the sub-data group *Bathymetry* you should make yourself acquainted with the concept of the staggered grid applied in Delft3D.

In Delft3D-FLOW we apply a staggered grid, i.e. not all quantities are defined at the same location in the numerical grid. When defining grid related quantities it is important you have familiarised yourself with the concept of staggered grids.

The staggered grid applied in Delft3D is given in [Figure 5.6](#), with the following legend:

full lines	the numerical grid
grey area	items with the same grid indices (m, n)
+	water level, concentration of constituents, salinity, temperature
-	horizontal velocity component in x - or ξ -direction (also called u - and m -direction)
	horizontal velocity component in y - or η -direction (also called v - and n -direction)
•	depth below mean (still) water level (reference level). Depth is defined positive downward.

Closed boundaries are defined through u - or v -points; open boundaries through u -, v - or

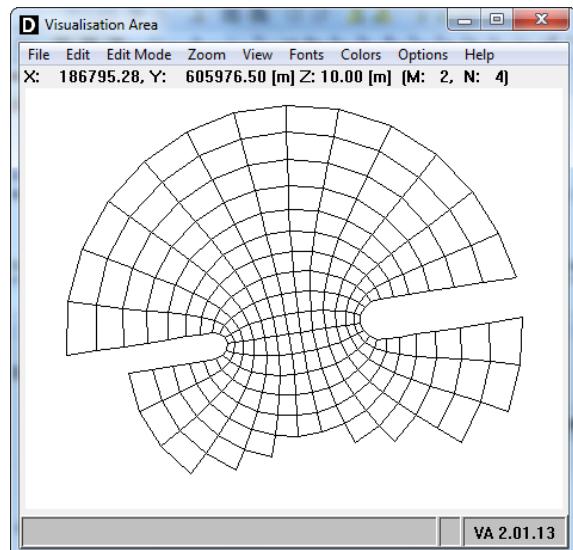


Figure 5.5: *Visualisation Area* window

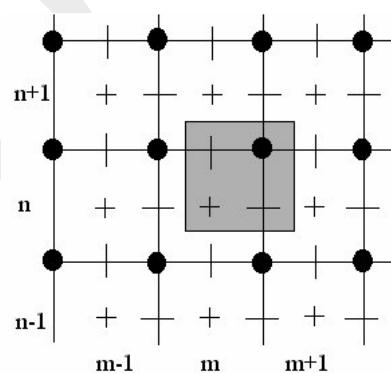


Figure 5.6: *Staggered grid* used in Delft3D-FLOW

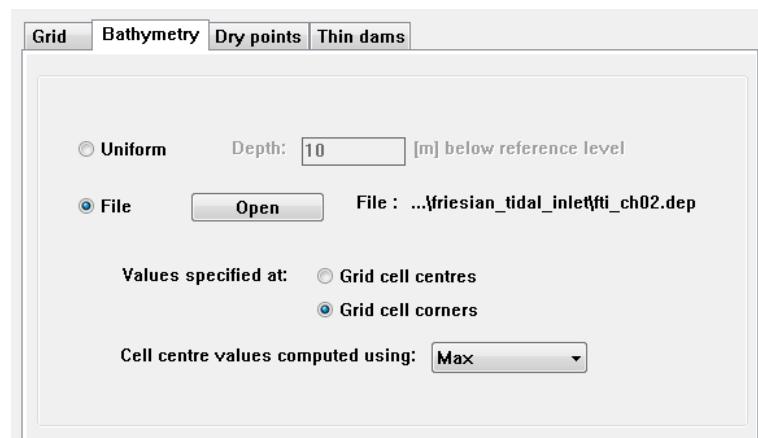


Figure 5.7: Sub-data group **Bathymetry**

water level (zeta-)points depending on the type of boundary condition such as velocity or water level. The location of other grid related quantities are given when appropriate.

5.5.2 Bathymetry

The next step in preparing the MDF-file is to set up the bathymetry of the area for which you just have loaded the grid:

- Click the sub-data group *Bathymetry*.
- Activate the *File* option.
- Select the file <fti_ch02.dep> after pressing the *Open* button.
- Behind *Values specified at*, select radio button *Grid cell corners*.
- The *Cell centre values computed using* is set to *MAX* by default.

The sub-window should look like [Figure 5.7](#).

Another option would have been to *Open* a (free or unformatted) bathymetry file.

Remark:

- ◊ The bathymetry is defined relative to the reference level and is counted positive downward to the bottom. The tool QUICKIN provides an option to compose a bathymetry from measured (sampled) data and to reverse the sign of grid related data in case the data is provided with the opposite sign.



5.5.3 Dry points

Dry points are grid cells that are permanently dry during a computation, irrespective of the local water depth. Dry points are located at the water level point, [Figure 5.8](#).

- Click the sub-data group *Dry points*.

Next the **Dry points** sub-window is displayed, [Figure 5.9](#).

You will see the layout of [Figure 5.9](#) in several other sub-data groups, so it is worthwhile to discuss this window in some more details. In this window you can see the following items:

- ◊ A list box with the names of the quantities defined so far. The name of a dry point is composed of its grid indices, but for other quantities (such as discharges) you can specify a name. You can select an item by clicking on its name in the list box.

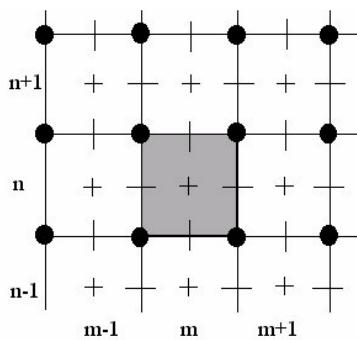


Figure 5.8: Location of a dry point at grid indices (m, n)

- ◊ Buttons *Add* and *Delete* for adding or deleting items from the list box.
- ◊ Buttons *Open* and *Save* for opening the items from an existing attribute file or saving the items to an attribute file.
- ◊ A couple of input fields to specify the location (and for some quantities a name field). For some quantities you can specify only one item at a time, so only one pair of (m, n) indices needs to be specified. For other quantities you can specify a line of quantities and then the indices of the start and end point must be specified.

To add a dry point:

- Click *Add*.

In the list box a dry point with default name **-Unnamed-** appears.

As dry points can be defined as a line of dry points you must specify the indices of the start and end point of the line, or set the indices the same for a single dry point.

- Select and enter one after the other the following values: M1 = "2", N1 = "13", M2 = "7", and N2 = "13".
- Click in the list box to confirm the input.

Now, you have defined a line of dry points along the grid line N=13 and extending over several M-grid points, i.e. M=2, ..., M=7. In the list box the name of the line of dry points is changed to **(2,13)..(7,13)**; note the name is composed of the grid indices of start and end point.

To see the just defined dry points:

- Select in the GUI menu *View* → *Visualisation Area*.

In the **Visualisation Area** window you can see the indicated fields light up in red; this implies that dry points have been defined at these locations and are selected.

- Remain in the **Visualisation Area** window and select *View* → *Legend*.

A window is displayed with a legend of the symbols that can be displayed in the **Visualisation Area** window. Minimise this window after inspection to keep it near by.

You can also define dry points by means of point-and-click in the **Visualisation Area** window. Please note that while moving the cursor over the visualisation area the position of the cursor is displayed just below the menu bar in both world coordinates and grid indices.

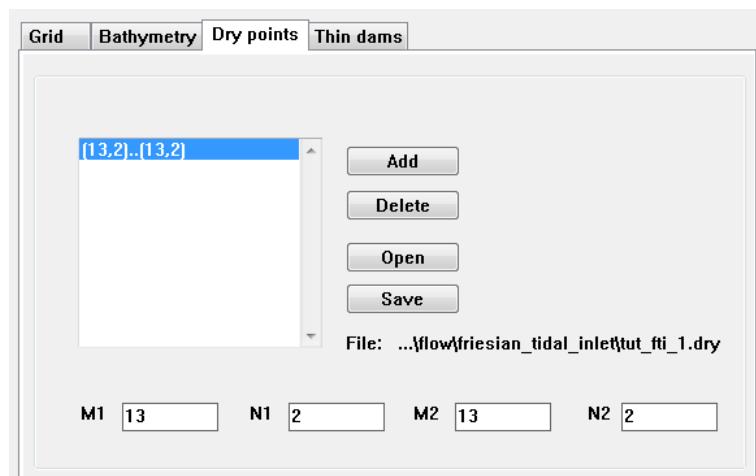


Figure 5.9: Sub-data group Dry Points

To add a dry point by point-and-click in the **Visualisation Area** window:

- Select *Edit Mode* → *Add* and select *Edit* → *Dry Point* in the menu bar.
- Move the mouse pointer to the grid cell with indices M=13 and N=2.
- Select this point.

Remark:

- ◊ Selecting a menu item or a grid related quantity actually consists of two actions: the first action is to move the cursor to the required item or position and next confirm the selection with clicking the left mouse. In short this process will be called selecting.



As a result the earlier defined six dry points in the grid have turned yellow, the newly defined dry point is displayed in red and in the *Dry points* list box the indices of the newly defined dry point is given. Now, before you move on to the next item you will delete the six defined dry points and single dry point, hence:

- Select the dry points **(2,13)..(7,13)** and click *Delete*; the dry points are removed from the list (and from the **Visualisation Area**).
- Click *Save* next to the list box.
- Specify the name <tut_fti_1.dry> in the working directory of the opened file selection window and save the file. If the file already exists the program will ask you whether you wish to overwrite the old file <tut_fti_1.dry>. Please confirm.

Below Save the filename <tut_fti_1.dry> appears, see [Figure 5.9](#).

Remark:

- ◊ You don't need to save the data in each sub-data group. The FLOW-GUI keeps track of all changed data fields. When selecting *File* → *New*, or when exiting the FLOW-GUI you will be notified that not all data has not been saved and you can either continue and discard this data or save the data to an existing or new file. However, a preferred way of working is to save data items when defined.



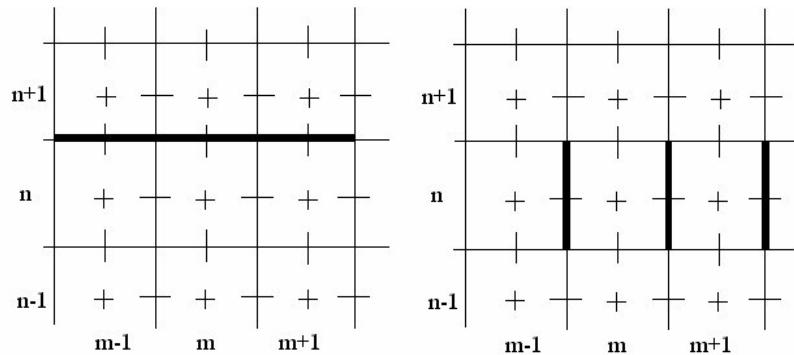


Figure 5.10: Sets of thin dams blocking v -velocities (left) and blocking u -velocities (right)

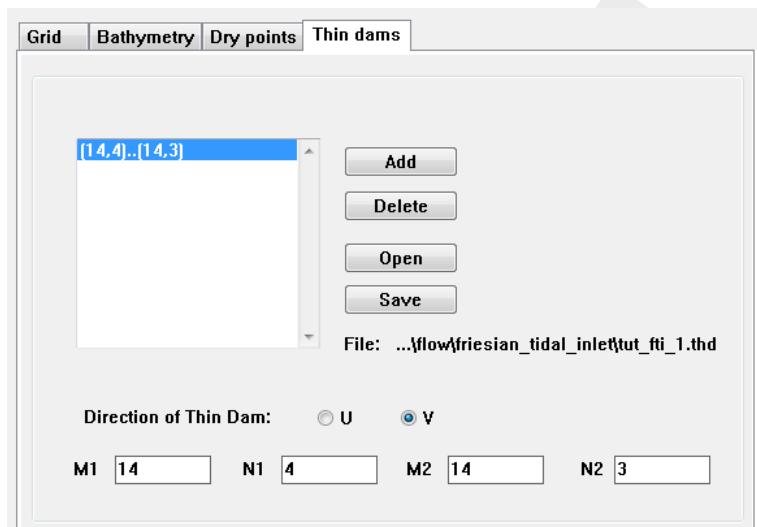


Figure 5.11: Sub-data group **Thin dams**

5.5.4 Thin dams

Next you are going to define a *Thin dam*. Thin dams are infinitely thin objects defined along the grid lines, which prohibit flow exchange between two computational cells at the two sides of the dam without reducing the total wet surface and the volume of the model. The purpose of a thin dam is to represent small obstacles (e.g. breakwaters or dams). See [Figure 5.10](#) for the location of u - and v -thin dams.

The **Thin dams** sub-window is displayed in [Figure 5.11](#).

- Click the sub-data group **Thin dams** in the **Domain** window. Similar to what happened in the previous section, there appears a sub-window for **Thin dams**.
- Define a line of thin dam points who block the flow in v -direction by entering the indices $M1 = "14"$, $N1 = "4"$, $M2 = "14"$ and $N2 = "3"$.
- The selected grid section turns red in the visualisation area to indicate thin dams have been defined and are currently selected.
- Save the thin dams in the file `<tut_fti_1.thd>`.

You have just defined two thin dams with a length of one grid cell each. In the list box a name composed of the indices of start and end point is displayed. The selected direction of the thin dam is indicated in the check box of u - or v -direction.

You can also define a single thin dam or a line of thin dams by point-and-click in the **Visuali-**

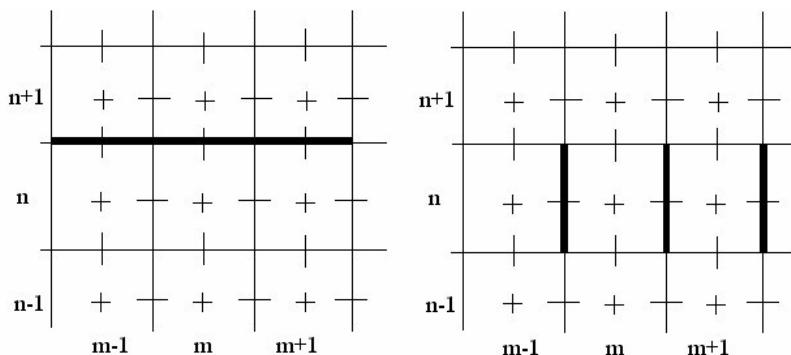


Figure 5.12: Equivalence of *v*-type thin dams (left) and *u*-type thin dams (right) with the same grid indices, ($m - 1$ to $m + 1, n$)

sation Area window:

- Select *Edit Mode* → *Add* and *Edit* → *Thin Dams* in the **Visualisation Area** menu bar.
- Click a grid location, but keep the mouse button pressed.
- Move the mouse pointer to the required length and release the mouse button.
- The selected grid line section turns yellow, thus indicating a thin dam has been defined.

Remark:

- ◊ You can define lines or sets of thin dams along the grid lines but also at angles of a multiple of 45 degrees with the grid.



Add a couple of more thin dams, change the direction in which the thin dam is defined by clicking the *u*- or *v*-direction check box and see how the change is visualised in the **Visualisation Area** window. Familiarise yourself with this, understand why a line of thin dams along one grid line changes to a set of single thin dams along the other grid direction when changing from *u*- to *v*-direction and vice versa. See [Figure 5.12](#) for *u*- and *v*-types of thin dams with the same grid indices.

Before you go on to the next data group you should undo the random changes; the easiest way to do this is to neglect the additional thin dams and go on to the next data group, as you already have saved the firstly defined thin dams. However, this would stimulate a wrong working habit and so you are advised to delete all the additional thin dams before you leave the data group, or to overwrite the currently defined thin dams by the saved thin dams in `<tut_fti_1.thd>`. So:

- Open the thin dam file `<tut_fti_1.thd>`.

You have defined all the required items in the **Domain** window, so you can move on to the next data group.

Time frame

Reference date	05 08 1990 [dd mm yyyy]
Simulation start time	05 08 1990 00 00 00 [dd mm yyyy hh mm ss]
Simulation stop time	06 08 1990 01 00 00 [dd mm yyyy hh mm ss]
Time step	5 [min]
Local time zone (LTZ)	0 +GMT

GMT = Local time - LTZ

Figure 5.13: Window Time frame

5.6 Time frame

In the Data Group *Time frame* you can specify the time frame of your computation composed of the reference date, the simulation start time, the simulation stop time and the time step used in the numerical simulation.

- Click the Data Group *Time frame*. In the canvas area the following sub-window is displayed, [Figure 5.13](#):



Remark:

- ◊ When this window is first activated it always shows the current calendar date of your computer.

The *Reference date* prescribes the (arbitrary) $t = 0$ point for all time-series as used in the simulations. However, in the FLOW-GUI all time related input is specified in a real time format, i.e. [dd mm yyyy] for the date and [hh mm ss] for the time.

- Click the *Reference date* input field and type the date “05 08 1990”.
- Set the *Simulation start time* to “05 08 1990 00 00 00”.
- Set the *Simulation stop time* to “06 08 1990 01 00 00”.
- Set the time step to “5” minutes.

The simulation time thus covers exactly 25 hours, or about two semi-diurnal tidal cycles.

Neglect the other input field.



Remarks:

- ◊ All data items of the Data Group *Time frame* are stored in the MDF-file and not in an attribute file. In fact all single data items are stored in the MDF-file.
- ◊ The maximum Courant number given the grid, depth and time step is about 11.

5.7 Processes

In the Data Group *Processes* you specify which processes or quantities that might influence the hydrodynamic simulation are taken into account. You can select constituents:

- ◊ Salinity
- ◊ Temperature
- ◊ Pollutants and tracers
- ◊ Sediments

or physical processes:

- ◊ Wind
- ◊ Wave
- ◊ Secondary flow (2D mode only and no Z-model)
- ◊ Online Delft3D-WAVE
- ◊ Tidal Forces (spherical co-ordinates only)

or man-made processes:

- ◊ Dredging and dumping (only if sediments are selected)

To select the required processes:

- Click the Data Group *Processes*.

The *Processes* sub-window is displayed with the list of possible constituents and processes, see [Figure 5.14](#). The buttons *Edit* are only displayed after you have selected the constituents *Pollutants and tracers* or *Sediments*.

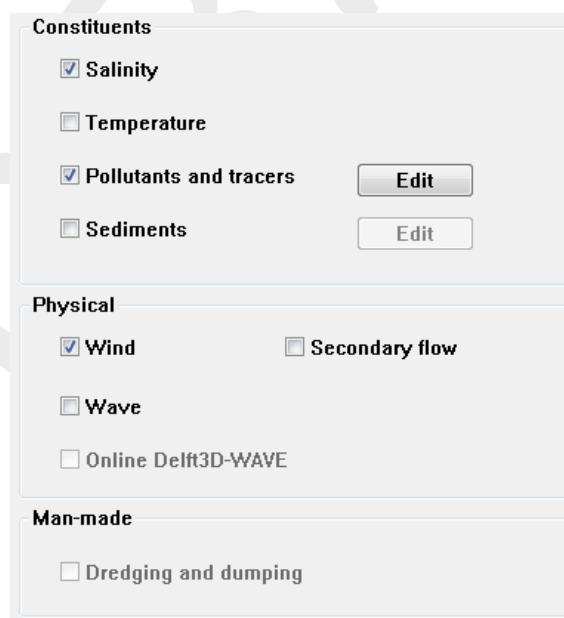


Figure 5.14: Processes window

- Activate the required processes by ticking off their check box; in this tutorial we select the constituents:

Salinity and
Pollutants and tracers,



Figure 5.15: Processes: Pollutants and tracers sub-window

Immediately after ticking off the *Pollutants and tracers* checkbox or when the *Edit* is pressed (if enabled), the **Processes: Pollutants and tracers** window is displayed, see [Figure 5.15](#).

And tick off the physical process

Wind.

- Now type into the *Name* input field the words “Conservative Spill” and click *Add*.
- Close the sub-window.

You have just defined one conservative constituent that is transported by the flow.



Remarks:

- ◊ The first time you select *Processes - Pollutants and tracers*, [Figure 5.14](#) is automatically displayed. For changing the constituent names later on you must click the aforementioned *Edit* button.
- ◊ You can specify a simple decay rate for each of the substances. It is currently implemented as a special feature, see [section B.2](#) for details. For interacting constituents you have to apply the water quality module D-Water Quality.
- ◊ See [chapter 11](#) for the use of cohesive and non-cohesive sediments.

5.8 Initial conditions

In the Data Group *Initial conditions* you can specify the initial values used when starting the computation. The initial conditions can be very simple, such as a uniform value in the whole area or taken from a previous computation.

- Click the Data Group *Initial conditions*.

The sub-window *Initial conditions* is displayed, [Figure 5.16](#).

Shown are some option boxes and input fields. In the dropdown list *Uniform values* has already been set. If not:

- Select *Uniform values*.

Input fields are shown for the processes selected in the Data Group *Processes*, so in this case for the processes water level (by default), salinity and one constituent.

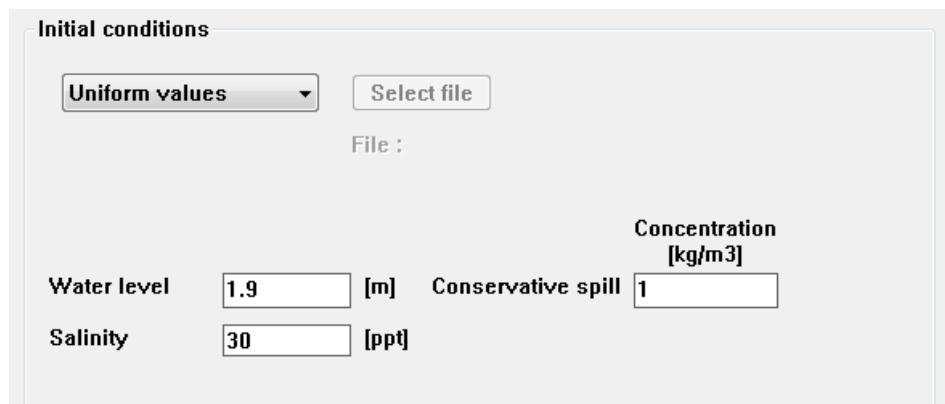


Figure 5.16: Initial conditions sub-window

- Enter the values: *Water Level*: “1.9” m; *Salinity*: “30.0” ppt and *Conservative Spill*: “1.0” kg/m3.

The initial conditions are stored in the MDF-file when saving the input.

5.9 Boundaries

In the Data Group *Boundaries* you can define the open boundaries, their location, type and all input data related to forcing the simulation. Open boundaries are defined in sections of one or more grid cells. Boundary conditions are prescribed at both ends of a section, called begin and end, or, A and B. Boundary conditions for intermediate points along a boundary section are determined by linear interpolation. Boundary sections may not be overlapping, so if you define two boundary sections along an open boundary, end-point B of section one and start point A of section two are located in consecutive grid cells.

Firstly, you are going to define the location and extent of the open boundaries:

- Click the Data Group *Boundaries*.
- Click *Add* to the right of the list box.

In the input field *Section name* appears the word **–Unnamed–**.

- Enter instead: “Sea Boundary”.

Remark:

- ◊ Values are read from the input field upon losing the focus, i.e. when you select *another* input field or select a boundary section in the list box.



Now define the begin and end point of the open boundary:

- Type the following values of the grid indices in the appropriate input fields: M1 = “2”, N1 = “22”, M2 = “14” and N2 = “22”.

Retain the value 0.0 in the *Reflection parameter alpha* input field.

In the **Visualisation Area** window the open boundary just defined is displayed as a thick red line.

To save these settings:

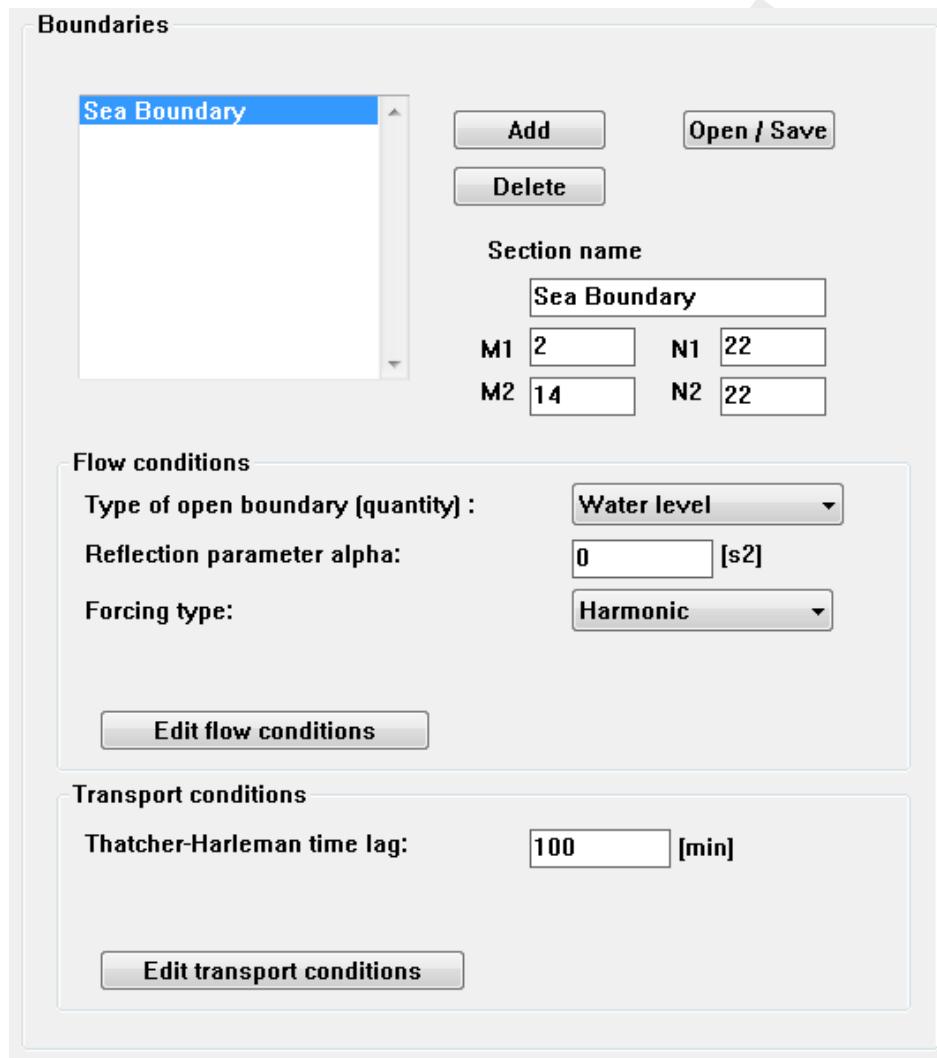


Figure 5.17: Boundaries sub-window

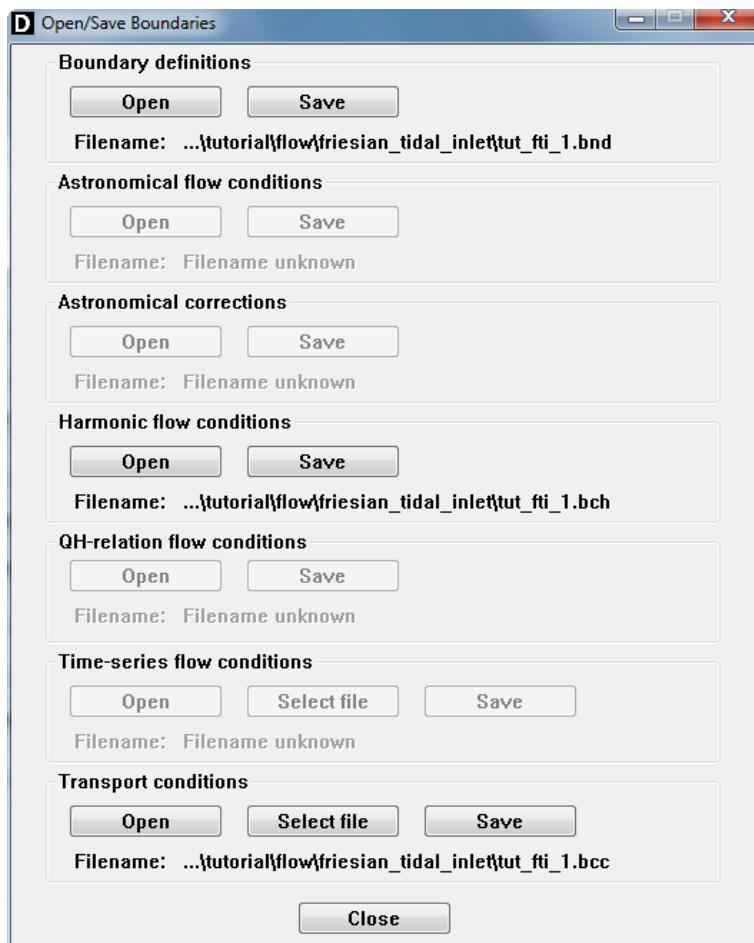


Figure 5.18: Open/Save Boundaries sub-window

- Click *Open/Save* in the **Boundaries** sub-window.
- Click *Save* (Boundary definitions) in the pop-up sub-window, [Figure 5.18](#).
- Enter the filename <tut_fti_1.bnd> in the working directory and close the **Open/Save Boundaries** menu window.

Remarks:



- ◊ After pressing *Open/Save* a sub-window is displayed with references to all type of attribute files related to open boundaries. Concentrate for the moment on the Boundary Definitions attribute file and discard the others.
 - ◊ You can define an open boundary in the **Visualisation Area** window in the same way as you define a line of thin dams or a line of dry points, i.e. select the begin point, keep the mouse button pressed, drag the mouse pointer to the end point of the boundary section along the grid line, and release the mouse button.
- For open boundaries there is one additional condition, i.e. the open boundary *must* be defined at the edge of the grid as shown in the **Visualisation Area**. To define an open boundary by point-and-click: select the *Edit Mode* → *Add* and *Edit* → *Open Boundaries* in the menu bar of the **Visualisation Area** window and continue as prescribed. Make sure you undo all changes; the easiest way is to import the previously saved open boundary locations.
- ◊ In the visualisation area all open boundaries are visualised at grid lines, i.e. at the edge of computational grid cells. The (M, N) indices displayed in the **Visualisation Area** refers to the computational grid cell. This grid cell number differs by one in M or N depending on the location of the boundary (left, lower, right or upper) from the grid line.

This rule also applies to open boundaries inside the computational domain. For a detailed discussion on the computational grid and the location of open and closed boundaries see [Appendix E](#).

- To define an open boundary by point and click you should select and drag inside the first (or last) grid cell so as to select the first (or last) water level or velocity point.

In this tutorial you will use the default type of open boundary i.e. *Water level* and the *Harmonic* type of forcing (the default is *Astronomic*).

- Set the forcing type to *Harmonic*.
- Click *Edit flow conditions*.

In this tutorial we are going to specify harmonic boundary conditions for a depth averaged computation (only one layer in the vertical, see Data Group *Domain* → *Grid*), so several options are either invisible or insensitive.

The harmonic components must be specified for both ends of an open boundary section. The begin and end point are referred to as ‘Begin’ or ‘point A’ and ‘End’ or ‘point B’. The Begin point has the indices (M1, N1) and the End point has the indices (M2, N2); see [Figure 5.17](#).

You are going to prescribe a semi-diurnal tide with amplitude of 1.1 m and phase 0 degrees at the Begin point and amplitude 1.1 m and phase -34.26 degrees in the End point.

After filling the input fields the table should look like [Figure 5.19](#).



Remark:

- ◊ The frequency is specified in degrees per hour. Hence, $360/12.5 = 28.8$ for the semi-diurnal tide. A mean sea level is defined with a frequency of 0 degrees/hr.

Now save the data to the attribute file:

- Close the **Boundaries - Flow Conditions** sub-window.
- Select *Open/Save* in the **Boundaries** window.
- Save the harmonic flow conditions into file <tut_fti_1.bch>.
- Close the **Open/Save Boundaries** window.

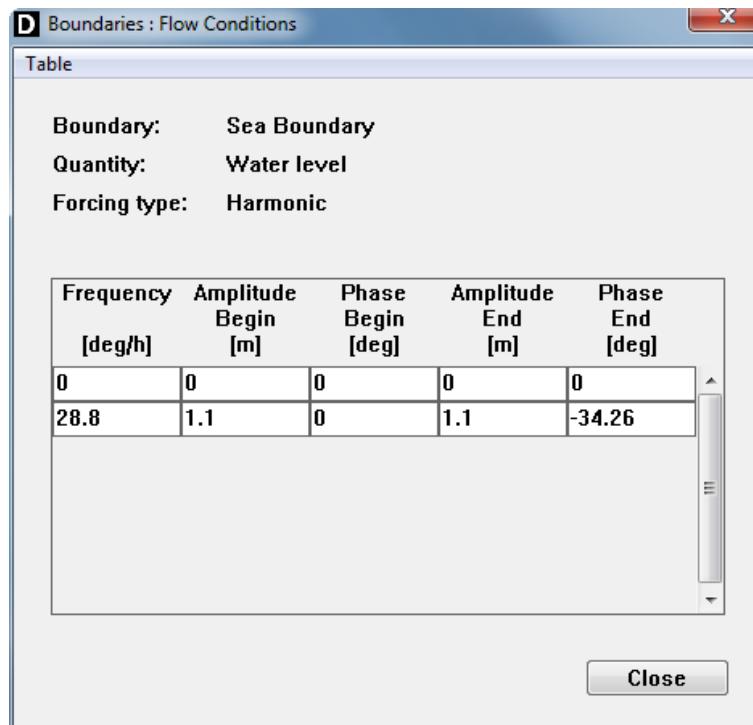
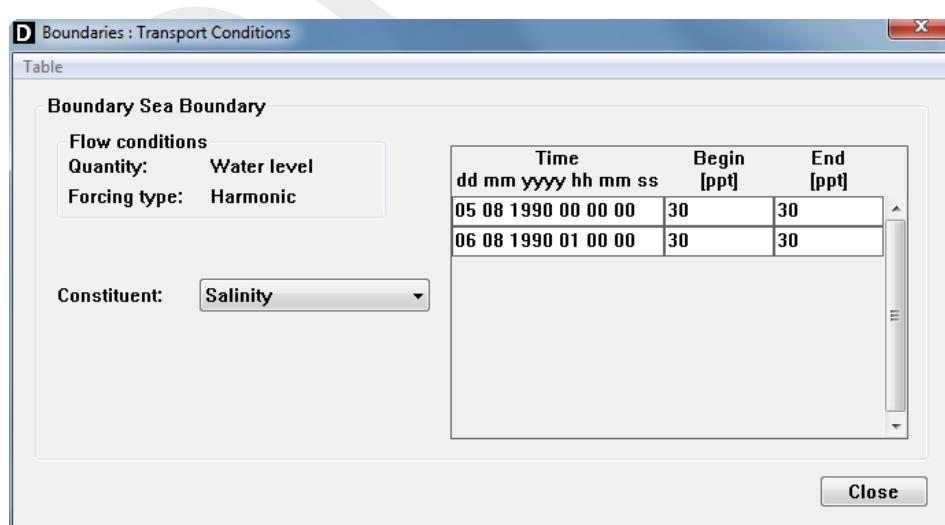
You are back in the **Boundaries** window. Next you will provide time-series for the concentrations of the constituent and salinity at the open boundary.

- Set the *Thatcher-Harleman time lag* to “100” minutes, see [Figure 5.17](#).
- Click *Edit transport conditions*.

To define the boundary conditions for the process salinity:

- Select from the *Constituents* dropdown list: *Salinity*.

Next, fill out the table form with the following time breakpoints and parameter values at both ends (Begin and End):

**Figure 5.19:** Harmonic boundary conditions**Figure 5.20:** Boundaries: Transport Conditions window

Date and time	Begin	End
05 08 1990 00 00 00	30.0	30.0
06 08 1990 01 00 00	30.0	30.0

➤ Next select from the dropdown list the constituent *Conservative Spill*.

and fill out the table with the following time breakpoints and parameter values at both ends.

Date and time	Begin	End
05 08 1990 00 00 00	1.0	1.0
06 08 1990 01 00 00	1.0	1.0

For this constituent you set all concentrations to 1.0, representing a tracer that is present in the inflowing sea water. For a tracer to be washed out of the system you should enter the sea-background concentration, for instance zero or a small value. Recall you have specified the initial condition (Data Group *Initial conditions*) to 1.0 for this constituent.



Remarks:

- ◊ The time breakpoints for *Salinity* and the constituent *Conservative Spill* and for any other process selected can be different; they *do not* need to be the same.
- ◊ In [Figure 5.20](#) only the input data for *Salinity* is visible; you must use the *Constituents* drop down list to select the other selected constituents.



Remark:

- ◊ In a 3D model you can specify a profile. To inspect this you can go back to the Data Group *Domain → Grid* and increase the number of layers to two (with a distribution of 50 % for each layer) and return to the current window (Data Group *Boundaries - Edit transport conditions - Constituents - Salinity*). Ignore all warnings and error messages that might be displayed (remember you might not have specified all kind of data required for a real 3D simulation). After inspection make sure to return the number of layers to one.

You have now finished the boundary conditions for all processes and constituents and you are ready to save the data.

- Close this window to return to the **Boundaries** window.
- Select in the **Boundaries** window *Open/Save - Transport conditions - Save* and save the transport boundary conditions in the working directory in the file <tut_fti_1.bcc>.
- Close the **Open/Save Boundaries** window.



Figure 5.21: Physical parameters sub-data groups

5.10 Physical parameters

In this Data Group you can define a number of physical parameters, grouped in several sub-data groups.

- Click the Data Group *Physical parameters*.

Next the sub-data groups *Constants*, *Roughness*, *Viscosity* and *Wind* are displayed as tabs; [Figure 5.21](#). As temperature was not selected in the Data Group *Processes* the sub-data group *Heat flux model* is invisible.

In the next sections the other sub-data groups will be treated.

5.10.1 Constants

The sub-data group *Constants* concerns items such as the acceleration of gravity, the water and air density and constants that determine the wind drag force. To specify the constants:

- Click the tab **Constants**, see [Figure 5.22](#).

Set the values by entering into their respective input fields:

- *Gravity*: “9.81” m/s² (default value).
- *Water density*: “1024.0” kg/m³.
- *Air density*: “1.00” kg/m³.
- *Temperature*: “10.0” °C.

There are three *Wind drag coefficients*:

- Type into the input fields of the first line “0.0025” [-] and “0.00” [m/s].
- Type into the input fields of the second line “0.0025” [-] and “100.00” [m/s].
- Type into the input fields of the third line “0.0025” [-] and “100.00” [m/s].

The three wind drag coefficients determine breakpoints in the piece-wise linear function of wind drag and wind speed:

- ◊ The first two coefficients determine the wind drag value that is used from zero wind speed up to the wind speed specified by the second constant.
- ◊ The third two coefficients specify the wind drag used from the specified wind speed and higher.

As a result you can specify a constant wind drag, a linearly increasing wind drag or a piece-wise linear function of wind speed.

Hydrodynamic constants			
Gravity	9.81	[m/s ²]	
Water density	1024	[kg/m ³]	
Air density	1	[kg/m ³]	
Temperature	10	[°C]	

Wind drag coefficients			
Breakpoints	Coefficient	Wind speed	
A	0.0025	H: 0	[m/s]
B	0.0025	H: 100	[m/s]
C	0.0025	H: 100	[m/s]

Figure 5.22: Physical parameters - Constants sub-window

Bottom roughness			
Roughness formula:	White-Colebrook ▾		
<input checked="" type="radio"/> Uniform	U: 0.05	V: 0.05	
<input type="radio"/> File	Select file		
File: Filename unknown			

Wall roughness			
Slip condition:	Free ▾		
Roughness length:	0	[m]	

Figure 5.23: Roughness sub-window

5.10.2 Roughness

In the sub-data group *Roughness* you can specify the bed roughness and for specific situations the roughness of the side walls.

- Click the tab **Roughness**, next [Figure 5.23](#) is displayed.

Three different friction formulations are given, i.e. *Manning*, *White-Colebrook* or *Chézy*:

- Select *White-Colebrook*.

The bottom roughness can be specified as a constant value over the whole area or as a space varying value (read from an attribute file). The option *Uniform* has already been activated by default, you only have to specify the constant value that is to be used.

- Enter the value “0.05” for both **U-** and **V**-direction.

Ignore the *Slip condition* dropdown box (and the roughness value related to it).

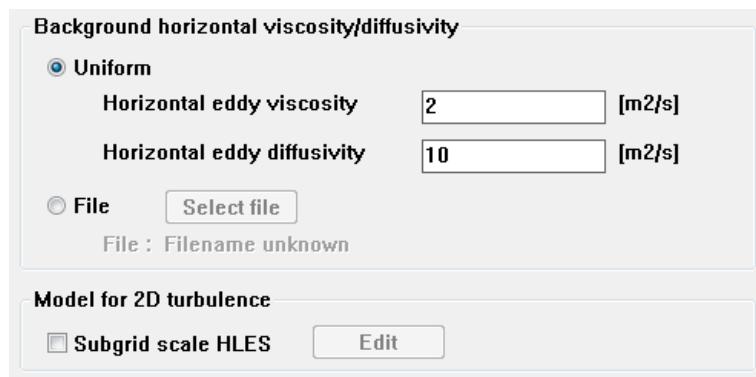


Figure 5.24: Viscosity sub-window

5.10.3 Viscosity

In the sub-data group *Viscosity* you can specify the eddy viscosity and the eddy diffusivity in the horizontal and vertical directions.

- Click the tab **Viscosity**, next Figure 5.24 is displayed.

As this is a depth averaged model, only the background *Horizontal eddy viscosity* and the background *Horizontal eddy diffusivity* are displayed and require a value:

- Set the horizontal eddy viscosity to “2.0” m²/s. and the horizontal eddy diffusivity to “10.0” m²/s, respectively.

Remark:

- ◊ You specify one value for both the Horizontal Eddy Viscosity and the Horizontal Eddy Diffusivity in U- and V-direction. Space varying values can be read from a file; these files must be prepared offline using some kind of editor or a program, such as the tool QUICKIN provided with Delft3D.



The model for 2D turbulence computes viscosities and diffusivities which will be added to the background values. In this tutorial we will not use this model.

5.10.4 Wind

Next you are going to specify the wind speed and direction:

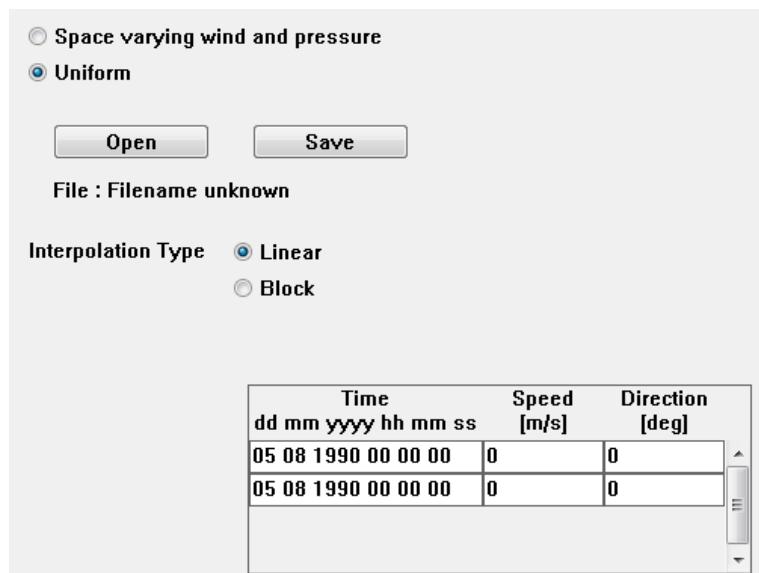
- Click the **Wind** tab.

Next Figure 5.25 is displayed. The check box *Uniform* is set, indicating you are specifying a uniform, but possibly time dependent wind field. Now for the sake of curiosity, select *Space varying wind and pressure*, and see what happens next. But then go back by activating again the *Uniform* check box.

You are going to define a linearly increasing wind speed from 0 m/s to 10 m/s during the first 12:30 hour of the simulation, the direction changing from 0 to 90 degrees in the same time interval, and keeping that value for the rest of the computation.

For operations on a table use the menu item *Table* in the menu bar. To add or insert (before the current row) a row:

- Select *Table* → *Copy row above*.

**Figure 5.25:** Wind sub-window

Time dd mm yyyy hh mm ss	Speed [m/s]	Direction [deg]
05 08 1990 00 00 00	0	0
05 08 1990 12 30 00	10	90
06 08 1990 01 00 00	10	90

Figure 5.26: Filled time table in the Wind sub-window

A new row has been inserted and you can modify the data.

In the time table, fill in the following data:

- “05 08 1990 00 00 00”, “0”, “0”
- “05 08 1990 12 30 00”, “10”, “90”
- “06 08 1990 01 00 00”, “10”, “90”

After editing, the table should look like [Figure 5.26](#).

Now save the wind data:

- Save the wind data in the working directory in the file <tut_fti_1.wnd>.

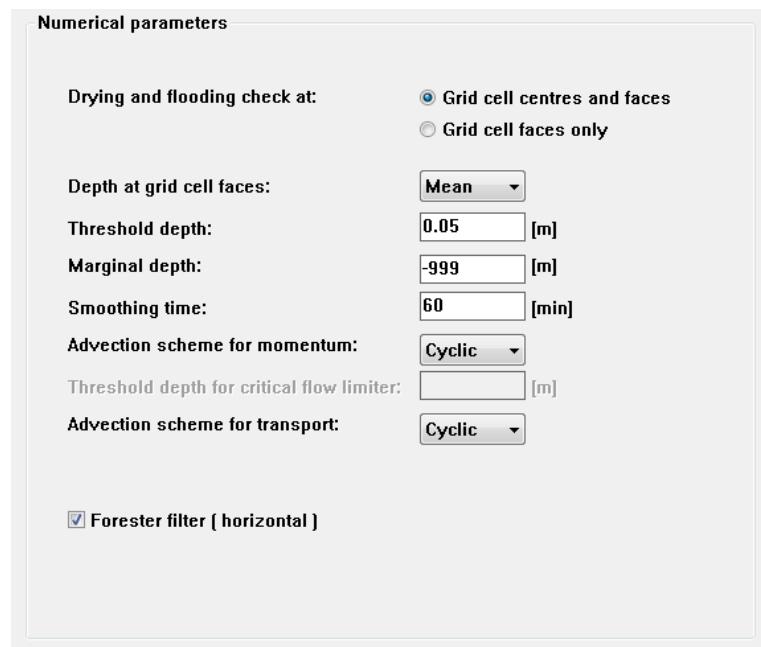


Figure 5.27: Numerical parameters sub-window

5.11 Numerical parameters

In the Data Group *Numerical parameters* you can specify parameters related to drying and flooding and some other advanced options for numerical approximations.

Although drying and flooding will not take place in this model, for the sake of getting a good overview of what is going on, you will include a drying and flooding procedure.

➤ Click the Data Group *Numerical parameters*, next Figure 5.27 is displayed.

The first selection to be made concerns drying and flooding. The options for *Drying and flooding check at* determine where the check should be performed:

- ◊ at grid cell centres (water level points) and grid cell faces (velocity points),
- ◊ or at only grid cell faces (velocity points).

A grid cell is flooded if the water depth is larger than the specified *Threshold depth* and dried if the water depth is smaller than half this value. See chapter 10 for details of this drying and flooding procedure. The *Marginal depth* is used to change the approximation of the convective terms in the momentum equations to allow for local super-critical flows. Skip for this tutorial the *Marginal depth* input field.

➤ Set the *Threshold depth* to “0.05” m.

The *Smoothing time* determines the time interval in which the open boundary conditions are gradually applied, starting at the specified initial condition to the specified open boundary conditions. This smoothing of the boundary conditions prevents the introduction of short wave disturbances into the model.

➤ Set the *Smoothing time* to “60” minutes.

Ignore the other options or input fields.

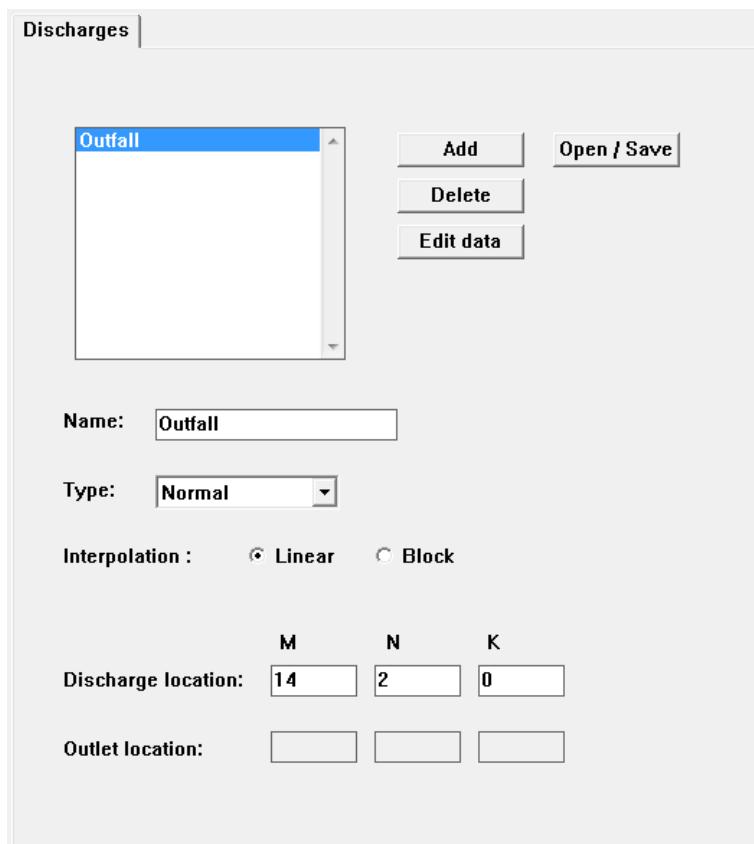


Figure 5.28: Data Group *Operations*; **Discharges** sub-window

5.12 Operations

In the Data Group *Operations* you can specify the location and all related properties of discharges. For this tutorial we have introduced one constituent (Conservative Spill); its release rate must be defined.

- Click the Data Group *Operations*.

The **Discharges** tab, [Figure 5.28](#), has a similar layout as the **Dry points** or **Thin dams** sub-window, but in addition you must specify the rate and concentrations of the discharged substances.

To add a discharge by point-and-click in the **Visualisation Area** window:

- Select *Edit Mode* → *Add* and select *Edit* → *Discharges* in the menu bar.
- Define the discharge location by point-and-click in the **Visualisation Area** window at grid cell M = "14" and N = "2".
- Replace the text (14,2) in the *Name* input field with "Outfall".

In the **Visualisation Area** window discharges are represented by a lozenge at a water level point, see [Figure 5.29](#).

The discharge is located at the centre of the computational cell (m, n).

To save the discharge location:

- Click *Open/Save* to open the **Open/Save Discharges** window.
- Click *Save* in group *Discharge definitions* to save the discharge location in the file <tut_fti_1.src>

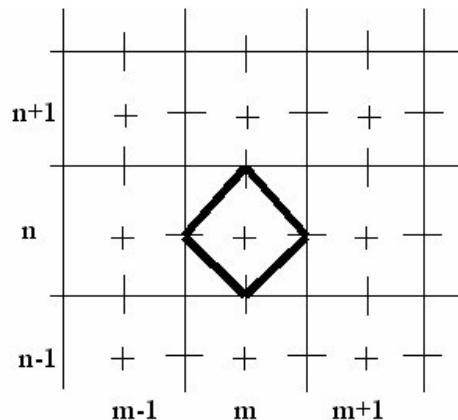


Figure 5.29: Representation of a discharge in the **Visualisation Area** window

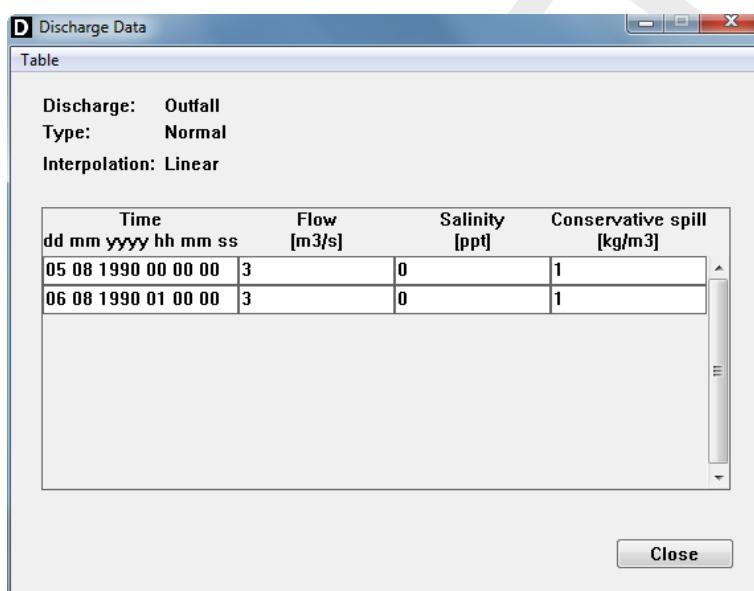


Figure 5.30: Discharge Data sub-window

in the working directory.

- Close the Open/Save Discharges window.

Next you must define the time dependent discharge data:

- Click *Edit data*.

The table to define the discharge rate and concentrations is displayed in [Figure 5.30](#).

Specify the following data in this table:

Date and time	Flow m³/s	Salinity ppt	Cons.Spill kg/m³
05 08 1990 00 00 00	3.0	0.0	1.0
06 08 1990 01 00 00	3.0	0.0	1.0



Remarks:

- ◊ The data specified represents a fresh water discharge (salinity = 0 ppt) of 3 m³/s.
- ◊ Obviously, the time breakpoints are the same for all time dependent quantities related to the discharge.
- ◊ Scroll the table horizontally, if needed, to display the Conservative Spill data input fields.

- Close the window if all input is specified.
- Click *Open/Save* and click on *Save* in datagroup *Discharge data* to save the discharge data in the attribute file <tut_fti_1.dis> in the working directory.
- Close the **Open/Save Discharges** window.

5.13 Monitoring

The Data Group *Monitoring* consists of three sub-data groups, i.e. *Observations*, *Drogues* and *Cross-sections*.

5.13.1 Observation points

Observation points are grid locations where all quantities computed during the simulations are stored at a user-defined time interval (Data Group *Output* → *Storage*). Observation points are located at the computational grid cell centres, i.e. at the water level points.

- Click the sub-data group *Observations*, [Figure 5.31](#) is displayed.

To add an observation point:

- Select from the menubar *Edit mode* → *Add* and *Edit* → *Observation Point*.
- Select the location (M = 4, N = 17) in the **Visualisation Area** window.
- Replace the text **(4,17)** with "Obs1".
- Repeat this for the observation points Obs2 to Obs5 as specified in the table below:

Name	M index	N index
Obs1	4	17
Obs2	9	17
Obs3	12	17
Obs4	9	11
Obs5	9	7

- Save the Observation Points just defined in the file <tut_fti_1.obs> in the working directory.

In the **Visualisation Area** window observation points are represented by a cross through the computational grid cell centre, [Figure 5.32](#).

The observation point is defined at (m, n).

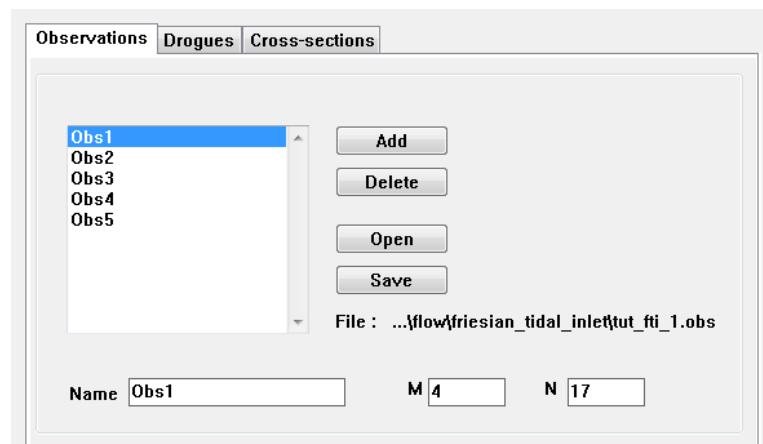


Figure 5.31: *Observation points* sub-window

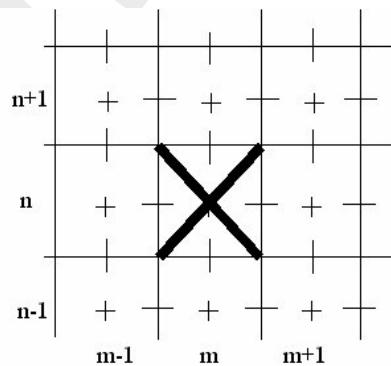


Figure 5.32: Representation of an observation point in the *Visualisation area* window

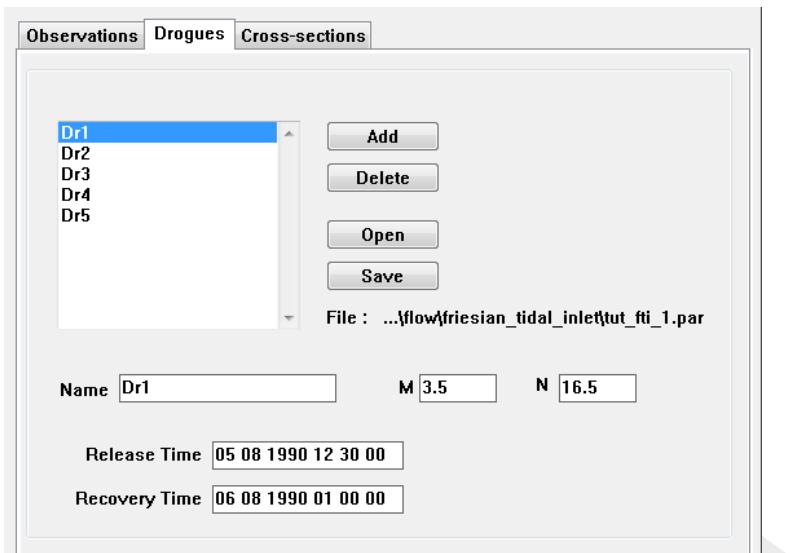


Figure 5.33: Drogues sub-window

5.13.2 Drogues

Drogues are floats that move with the flow. They are released and recovered at a user-specified time. Their location is stored at the same time interval as the computational time step (see Data Group *Time frame*). Drogues can be located anywhere in a grid cell, i.e. a drogue can be released at decimal indices. However, in the **Visualisation Area** window you can only select grid *centre* location, so if other decimal values are required you must enter these indices manually in the FLOW-GUI.

- Click the sub-data group *Drogues*, Figure 5.33 is displayed.

To add drogues:

- Click *Add* and enter the following drogues:

Name	M index	N index
Dr1	3.50	16.50
Dr2	8.50	16.50
Dr3	11.50	16.50
Dr4	8.50	10.50
Dr5	8.50	6.50

- The *Release Time* and *Recovery Time* are by default the same as the start and stop times of the simulation. For all drogues change them to:
 Release Time : "05 08 1990 12 30 00"
 Recovery Time : "06 08 1990 01 00 00"
- Save the drogue data in the file <tut_fti_1.par> in the working directory.

In the **Visualisation Area** window drogues are represented by a '+' through the computational grid centre, Figure 5.34.

The drogue is located at (m, n).

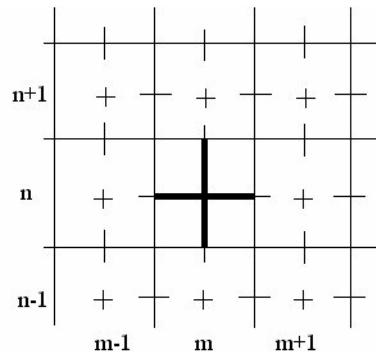


Figure 5.34: Representation of a drogue in the **Visualisation Area** window

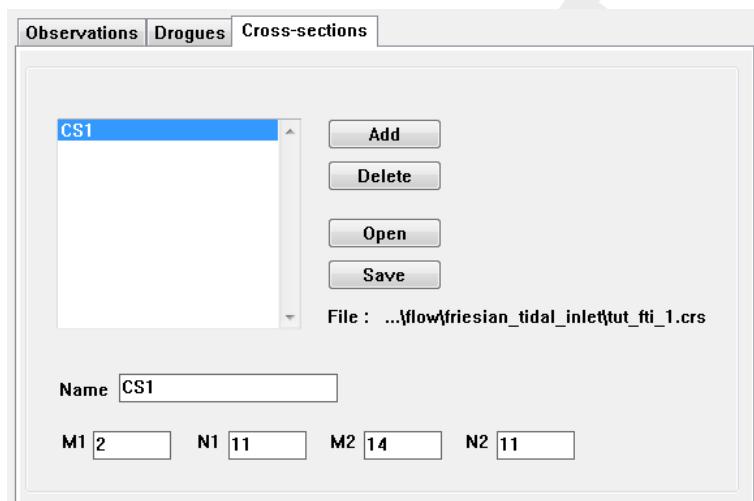


Figure 5.35: **Cross-sections** sub-window

5.13.3 Cross-sections

Cross-sections are sections along one of the grid directions through which the various kinds of transport fluxes are determined and stored as a function of time. For defining cross-sections you can apply the same procedures as for *Thin dams* or *Dry points* in the Data Group *Domain*. Cross-sections are defined along the grid lines, either as U- or as V-cross-section. To enter the sub-data group *Cross-sections*:

- Click *Cross-sections*, next Figure 5.35 is displayed.

To add a cross-section:

- Click *Add*.
- Type in the *Name* field the word “CS1” to replace **–Unnamed–**.
- Enter the indices M1 = “2”, N1 = “11”, M2 = “14” and N2 = “11”.

In the **Visualisation Area** window a cross-section is represented by a thick line along one of the grid lines, Figure 5.36.

In Figure 5.36 the V-cross-section runs from (m–1, n) to (m+1, n).

Remark:

- ◊ Cross-sections must have a minimum length of two grid cells.



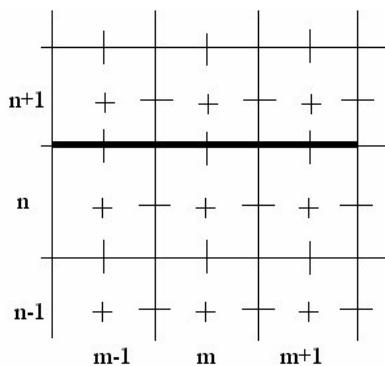


Figure 5.36: Representation of a v-cross-section in the **Visualisation Area** window

To save the just defined cross-section:

- Click **Save**.
- Save the data in file <tut_fti_1.crs> in the working directory.



Remark:

- ◊ You are encouraged to define a cross-section along either of the grid lines by point-drag-and-click. Make sure to undo the additional definitions (for instance by opening the just saved attribute file) before you move on to the next data group.

5.14 Additional parameters

In the Data Group *Additional parameters* you can specify keywords and their value, often the name of an attribute file, which are not yet supported by the FLOW-GUI. For details you are referred to [Appendix B](#), but for this tutorial we ignore this Data Group.

5.15 Output

In the Data Group *Output* you can specify which computational results will be stored for further analysis or other computations and which output shall be printed. Though the printing option is hardly used it is a useful option when numerical output values are required.

To enter the data group:

- Click the Data Group *Output*.

The Data Group *Output* supports three sub-data groups:

- *Storage*: to define the time interval to store results to disk.
- *Print*: to define the time interval to store results to a printable file.
- *Details*: to define which quantities will be stored or printed.

Upon selecting *Storage* the sub-window, [Figure 5.37](#), will be displayed.

Before defining the various input fields we want to explain in short the use of the various files used to store the results. Basically, Delft3D-FLOW uses four types of files to store results:

- ◊ *History file*

The history file contains all quantities as a function of time, but only in the specified Monitoring Points and Cross-Sections. The amount of data is usually much smaller than for a map file, and you typically store history data at a small interval to have a smooth time

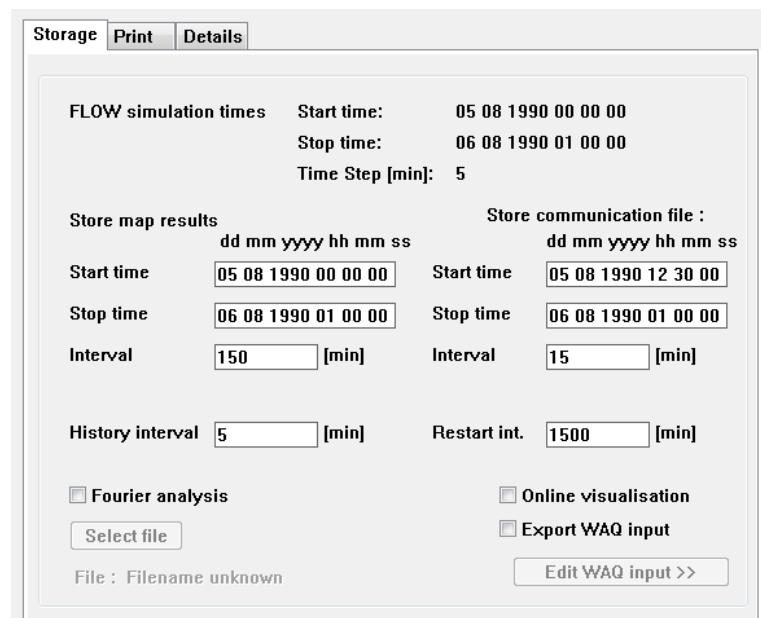


Figure 5.37: Output sub-window

function when plotting the results.

◊ *Map file*

The map file contains snap shots of the computed quantities of the entire area. As you save all results in all grid points a typical Map file can be many hundreds of Mbytes large. So typically map results are only stored at a small number of instances during the simulation.

◊ *Drogue file*

The drogue file contains the (x, y) -position of all drogues at each computational time step in the time interval between release and recovery time.

◊ *Communication file*

The communication file contains results required by other modules of Delft3D, such as the hydrodynamic results for a water quality simulation or the wave forces for a wave-current interaction. As the results must be stored in all grid points a communication file can be as large as or even larger than a Map file. So typically you only store the results in the Communication file as far as needed for the other simulations; for water quality simulations you typically store the results of the last day or a (couple of) cyclic tidal cycle and use this data repeatedly to make a water quality simulation over many days or weeks.

In order not to waste computing time you can restart a simulation at a predefined time of a previously executed simulation (Data Group *Initial conditions*). For this you specify the *Restart Interval* at which all information required to restart the simulation. Typically you select a large restart interval in order not to waste disk space.

Now continue to define the various times and time intervals:

◊ Set *Store map results* to:

Start time: "05 08 1990 00 00 00"

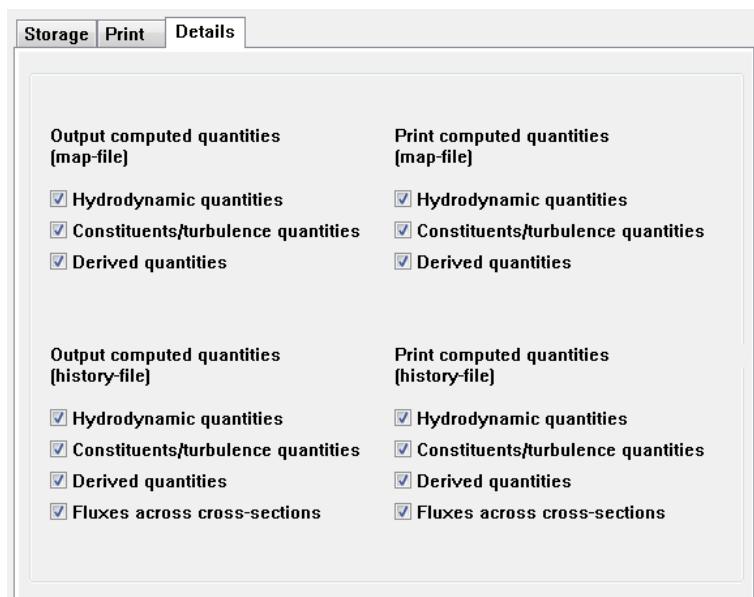
Stop time: "06 08 1990 01 00 00"

Interval time: "150" min.

◊ Set *Store communication file* to:

Start time: "05 08 1990 12 30 00"

Stop time: "06 08 1990 01 00 00"

**Figure 5.38: Output details sub-window**

Interval time: “15” min

- ◊ Set *History interval* to: “5” minutes.
- ◊ Set *Restart int.*: “1500” min



Remarks:

- ◊ The interval to write the various files *must* be an integral multiple of the computational time step, defined in the Data Group *Time frame*.
- ◊ The history, map, and drogue data is stored by default in binary NEFIS files. This file format is best suited for further post-processing using Deltires software. For interoperability with other software you may want to use the binary NetCDF file format instead. The file format can be changed by means of the `F1Ncdf` keyword; see [subsection A.1.4](#).

The restart interval as set results in a restart file at the end of the simulation.

For this tutorial we ignore the item *Fourier analysis*, it is discussed in detail in [chapter 4](#).

- Tick off *Online visualisation*, this results in viewing the results during the simulation.

To control the quantities that are actually stored on files or printed:

- Click the sub-data group *Details*, next [Figure 5.38](#) is displayed.

By ticking the check boxes you can specify which items you want the program to store on file or to print and which not; details of the quantities covered by each group are given in [chapter 4](#).

For this tutorial we ignore the sub-data group *Print*.

You have now gone through all Data Groups and you have defined in most of them one or more data items. Before continuing you should save this data to make sure you can always return to this status of the FLOW-GUI.

5.16 Save MDF-file

In the preceding sections you have entered many data items, some of which are already saved in attribute files.

To summarise:

- 1 Dry Points <tut_fti_1.dry>
- 2 Thin Dams <tut_fti_1.thd>
- 3 Open Boundaries definitions <tut_fti_1.bnd>
- 4 Wind <tut_fti_1.wnd>
- 5 Flow boundary conditions (harmonic) <tut_fti_1.bch>
- 6 Transport boundary conditions <tut_fti_1.bcc>
- 7 Discharge locations <tut_fti_1.src>
- 8 Discharge rates <tut_fti_1.dis>
- 9 Observation points <tut_fti_1.obs>
- 10 Drogues <tut_fti_1.par>
- 11 Cross sections <tut_fti_1.crs>

In addition you have used three predefined attribute file:

- 1 Curvilinear Grid <fti_02.grd>
- 2 Grid Enclosure <fti_02.enc>
- 3 Bathymetry <fti_ch02.dep>

To save the reference to all these attribute files and other (single) input data in the MDF-file:

- Click *File* → *Save MDF As* in the menu bar of the FLOW-GUI.
- Save the MDF-data in the file <tut_fti.mdf> in the working directory.

Remarks:



- ◊ Do not use blanks in the name of the MDF-file.
- ◊ All attribute data is saved in attribute files (if they exist) and the references to existing attribute files are stored in the new MDF-file.
- ◊ If you have defined *all* attribute files in their respective Data Group, the MDF-file is complete.
- ◊ All unsaved Data Groups are ignored.
- ◊ Upon selecting *File* → *New*, *File* → *Open* or *Exit* you will be notified if some data is not saved and you can choose to ignore the data, or to save the data.
- ◊ In previous versions of the FLOW-GUI you were obliged to save the attribute data in their respective Data Group. A warning was issued on leaving a Data Group without saving its related attribute file(s). Starting this version of the FLOW-GUI you can save attribute data in their respective Data Group, but you can postpone it to the moment when you save the MDF-data. You will be prompted for unsaved data.

To exit the program:

- Click *File* → *Exit*.

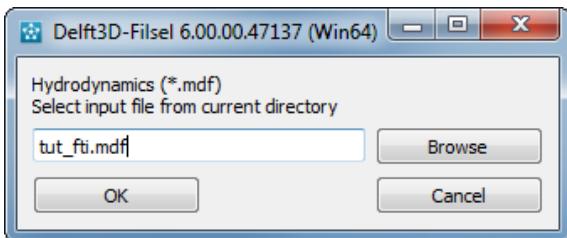


Figure 5.39: Select scenario to be executed

5.17 Additional exercises

You are encouraged to go back to the start of the tutorial and explore some of the options not used or discussed in this first example. A very worthwhile exercise is to define, delete or modify the definition of grid related quantities using the **Visualisation Area** window only. With the menu item *Edit Mode* → *Modify* you can drag-and-drop (i.e. move) any of the grid related quantities.

5.18 Execute the scenario

If you have an error free scenario you can start the simulation:

- Select *Start* in the Delft3D-MENU to start the computation in foreground. In [Figure 5.39](#) the selection window is displayed for the scenario to be executed (i.e. MDF-file).
- If <tut_fti.mdf> is not selected already, use *Select file* to navigate to the working directory and select <tut_fti.mdf>.
- Confirm your selection with *OK*.

In foreground the status of the simulation and possible messages are displayed in the active window.

After the simulation is finished you are strongly advised to inspect at least some of the report files generated during the simulation to check if all went according to plan. This concerns especially the <tri-diag> file (located in the working directory). To see this report:

- Select *Reports* in the **Hydrodynamics (including morphology)** window.
- Select *Flow* to inspect the report file.

Especially the end of this file is of importance as it summarises errors, warnings and information of the simulation, see [Figure 5.40](#).

The simulation is executed without errors or warnings.

```

76 Delft3D - [tut_fti] - running computational program
File
Standalone FLOW computation
Time to finish 0s, 95.7% completed, time steps left 13
Time to finish 0s, 96.0% completed, time steps left 12
Time to finish 0s, 96.3% completed, time steps left 11
Time to finish 0s, 96.7% completed, time steps left 10
Time to finish 0s, 97.0% completed, time steps left 9
Time to finish 0s, 97.3% completed, time steps left 8
Time to finish 0s, 97.7% completed, time steps left 7
Time to finish 0s, 98.0% completed, time steps left 6
Time to finish 0s, 98.3% completed, time steps left 5
Time to finish 0s, 98.7% completed, time steps left 4
Time to finish 0s, 99.0% completed, time steps left 3
Time to finish 0s, 99.3% completed, time steps left 2
Time to finish 0s, 99.7% completed, time steps left 1

-----
FINISHED      Delft3D-FLOW
runid        : tut_fti
date, time  : 2013-03-24, 12:50:46
0 errors and 0 warnings
returning to main program from domain tut_fti

D_Hydro [46.000808] <anonymous> >> d_hydro shutting down normally

```

Figure 5.40: Part of the <tri-diag> report file

5.19 Inspect the results

In [Figure 5.41](#) to [Figure 5.43](#) some results are shown of the simulation just finished. To reproduce these plots you should start the general post processing program GPP (see the GPP User Manual [GPP UM \(2013\)](#)).

- Select *GPP* in the **Hydrodynamics (including morphology)** window or *Utilities - GPP* in the main Delft3D-MENU.
- In the main window of GPP select *Session → Open*.
- Copy the file <all_input\tut_fti.ssn> to the working directory <.\tut_fti.ssn>.
- In the file selection menu select and open the file <tut_fti.ssn>.
- In the main window of GPP select *Plots* and select from the list of possible plots the one you would like to inspect.

In a <*.ssn> file the references are stored to data sets, in this case the result files of the tut_fti-scenario, and the definition of earlier defined plots and their layout. By calling this scenario file you can inspect the same plots after repeating the simulation with (other input data) of the FLOW scenario tut_fti. For details of using GPP you are referred to the User Manual of GPP.

To return to the main window of GPP while viewing a plot:

- Select *Plot → Close*.

You can select another plot as described above.

To close GPP and return to Delft3D-MENU:

- Select in the main window of GPP *Session → Exit*.

To close Delft3D-MENU:

- Select *Return*.

➤ Select *Exit*.

DRAFT

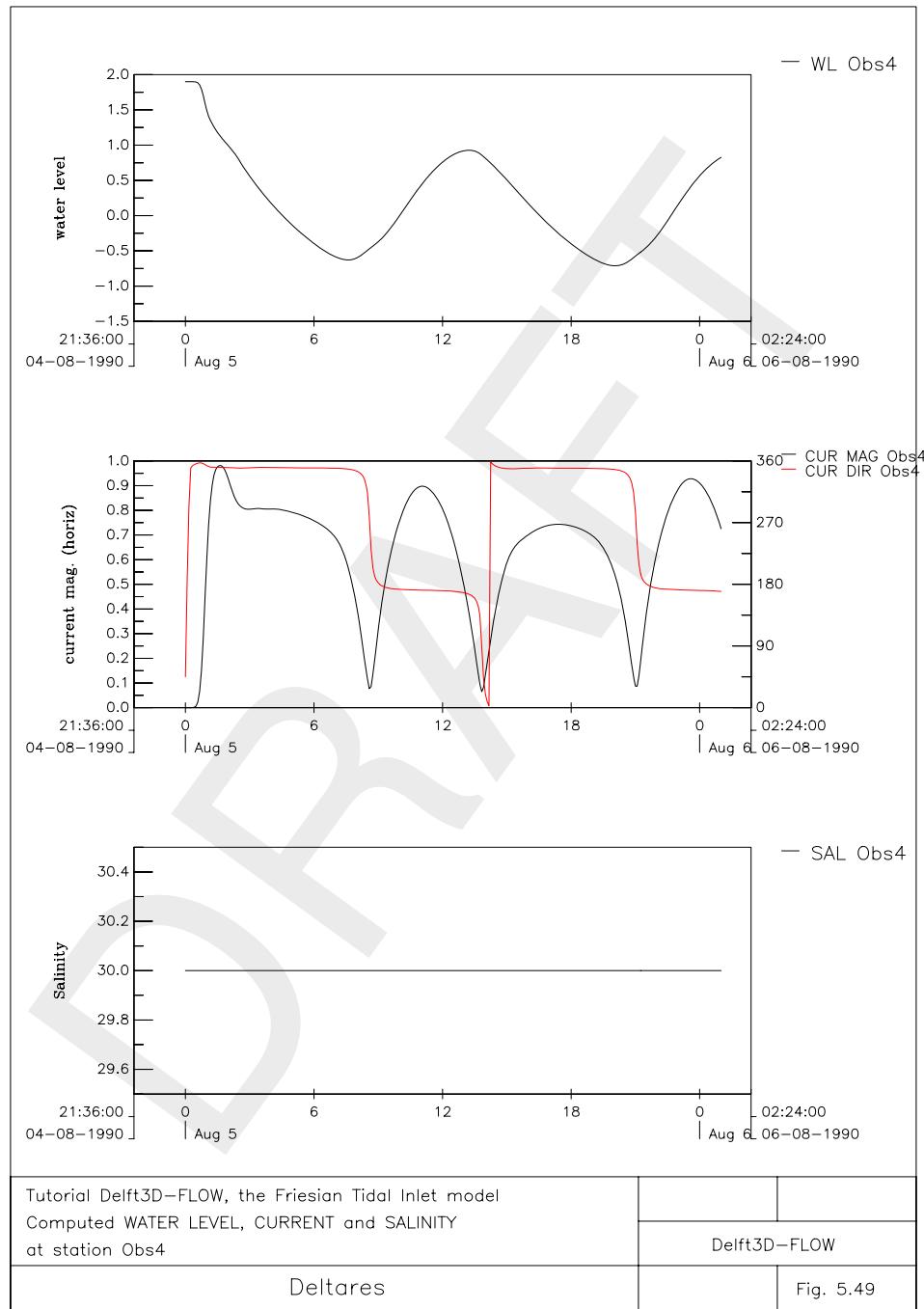


Figure 5.41: Computed time-series of the water level, current and salinity in observation point Obs4

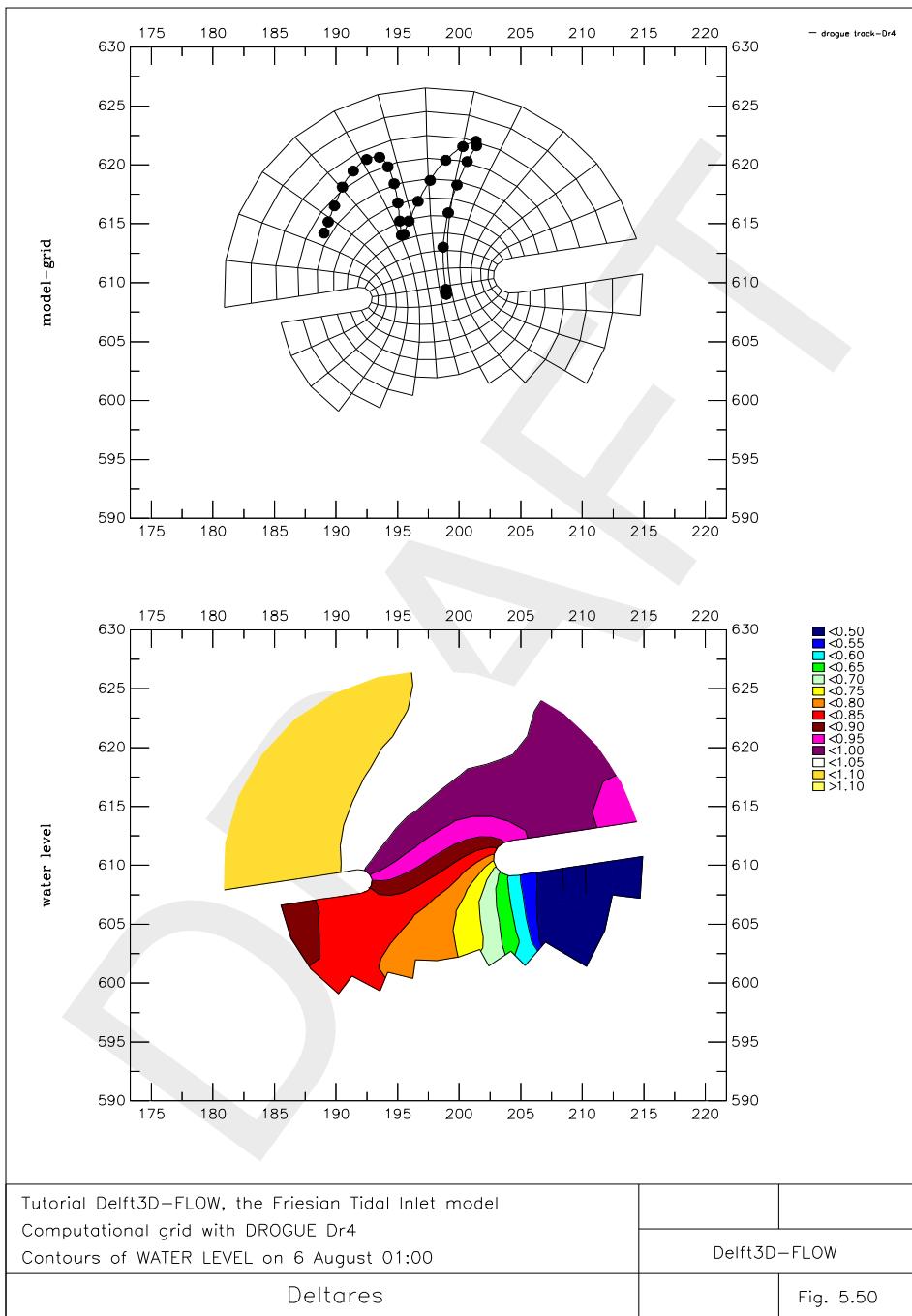


Figure 5.42: Computational grid with drogue Dr4, and contours of water level on 6 August 1990 01:00 hr

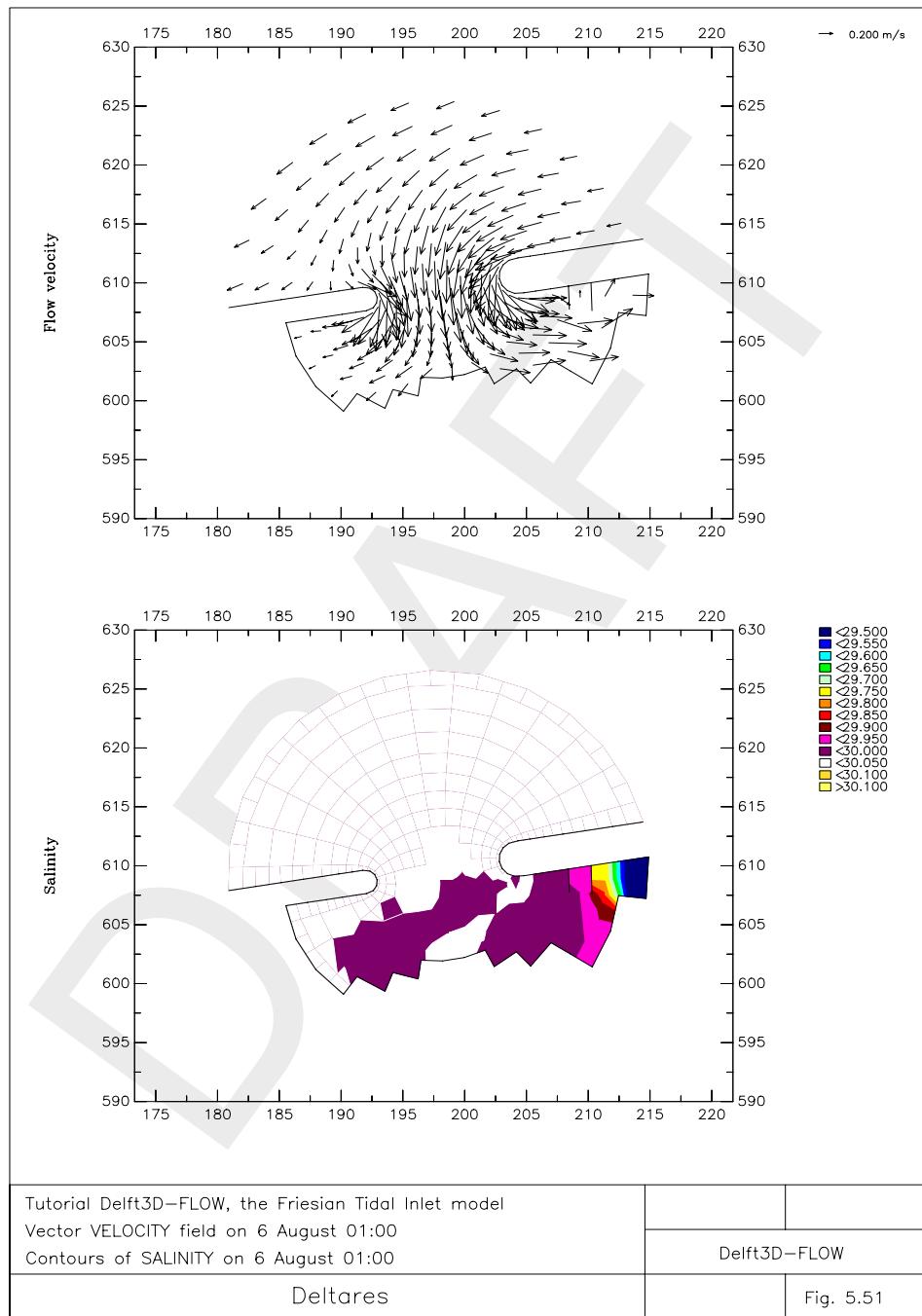


Figure 5.43: Vector velocities and contours of salinity on 6 August 1990 01:00 hr

DRAFT

6 Execute a scenario

6.1 Running a scenario

After defining the input for the Delft3D-FLOW hydrodynamic simulation, the computation can be executed either via Delft3D-MENU or using a batch script. Via Delft3D-MENU, the status of the computation and possible messages are displayed in a separate window. When using a batch script, all messages are written to a file and you can continue working in the current window.

Not all functionality is available when using Delft3D-MENU to start a calculation. Use a batch script (see [subsection 6.1.3](#)) in the following cases:

- ◊ Using MPI to run in parallel
- ◊ Using mormerge
- ◊ Using fluid mud
- ◊ Using some queueing mechanism on a cluster

6.1.1 Parallel calculations

This section is an overview of all techniques where parallel calculations are involved.

6.1.1.1 DomainDecomposition

See [section B.14](#). An example is at https://svn.oss.deltares.nl/repos/delft3d/trunk/examples/02_domaindecomposition. Each subdomain runs in a separate thread, inside one executable.

- ◊ Multiple cores will be used automatically when available on the executing machine.
- ◊ DomainDecomposition calculations can not be executed using multiple machines.
- ◊ Does not work in combination with MPI-based parallel calculations

6.1.1.2 MPI-based parallel

Example scripts are inside https://svn.oss.deltares.nl/repos/delft3d/trunk/examples/01_standard. The domain is split automatically in stripwise partitions.

- ◊ Can not be started via Delft3D-MENU
- ◊ Does not work in combination with:
 - DomainDecomposition
 - Fluid mud
 - Coup online
 - Drogues and moving observation points
 - Culverts
 - Power stations with inlet and outlet in different partitions
 - Non-hydrostatic solvers
 - Walking discharges
 - 2D skewed weirs
 - $\max(mmax,nmax)/npart \leq 4$
 - Roller model
 - Mmerge
 - Mass balance polygons

6.1.1.3 Fluid mud

See [section B.10](https://svn.oss.deltares.nl/repos/delft3d/trunk/examples/04_fluidmud). An example is at https://svn.oss.deltares.nl/repos/delft3d/trunk/examples/04_fluidmud. The two calculations run in separate executables.

- ◊ Multiple cores will be used automatically when available on the executing machine.
- ◊ Fluid mud calculations can not be executed using multiple machines.
- ◊ Does not work in combination with MPI-based parallel calculations

6.1.1.4 Mormerge

An example is at https://svn.oss.deltares.nl/repos/delft3d/trunk/examples/05_mormerge. A script, written in the language "Tcl" is used to start all executables:<mormerge.tcl>.

- ◊ Multiple cores will be used automatically when available on the executing machine.
- ◊ Does not work in combination with MPI-based parallel calculations

6.1.2 Running a scenario using Delft3D-MENU

To start Delft3D-FLOW:

- ◊ Select Start, see [Figure 6.1](#).

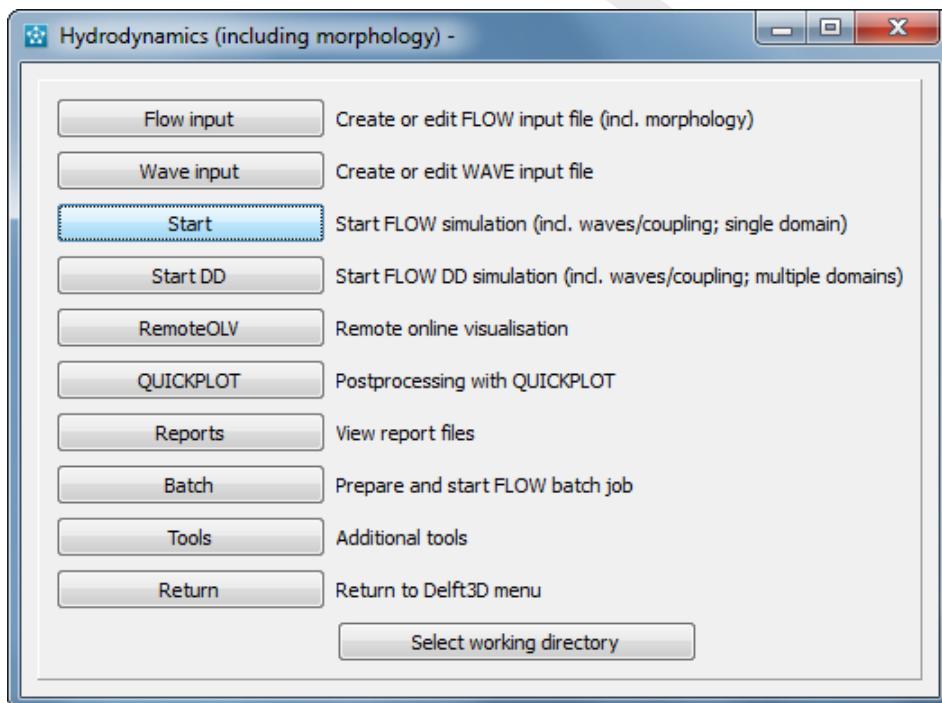


Figure 6.1: MENU-window for Hydrodynamics

A new window is displayed in which you can select the input file to be used, [Figure 6.2](#).

- ◊ Apply Select file to select the required MDF-file.

After pressing *OK* the simulation will start and a window will appear. In foreground the status of the simulation and possible messages are displayed in the active window, see [Figure 6.3](#)

After the simulation is finished you are strongly advised to inspect at least some of the report files generated during the simulation to check if all went according to plan. To see which

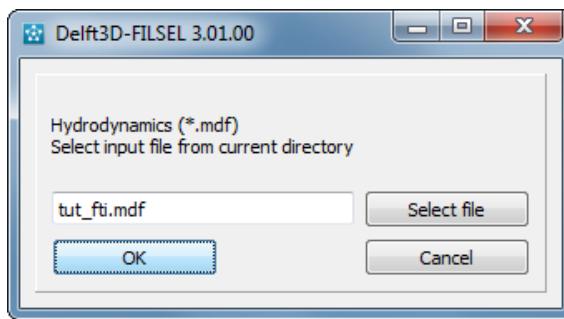


Figure 6.2: Select the MDF-file to be verified

```

Time to finish 0s, 95.7% completed, time steps left 13
Time to finish 0s, 96.0% completed, time steps left 12
Time to finish 0s, 96.3% completed, time steps left 11
Time to finish 0s, 96.7% completed, time steps left 10
Time to finish 0s, 97.0% completed, time steps left 9
Time to finish 0s, 97.3% completed, time steps left 8
Time to finish 0s, 97.7% completed, time steps left 7
Time to finish 0s, 98.0% completed, time steps left 6
Time to finish 0s, 98.3% completed, time steps left 5
Time to finish 0s, 98.7% completed, time steps left 4
Time to finish 0s, 99.0% completed, time steps left 3
Time to finish 0s, 99.3% completed, time steps left 2
Time to finish 0s, 99.7% completed, time steps left 1

-----
FINISHED      Delft3D-FLOW
runid         : tut_fti
date, time   : 2013-03-24, 13:04:43
0 errors and 0 warnings
returning to main program from domain tut_fti
-----
D_Hydro [43.000698] <anonymous> >> d_hydro shutting down normally

```

Figure 6.3: Part of the report to the output window

reports are generated:

- Select *Reports* in the **Hydrodynamics (including morphology)** window, upon which [Figure 6.4](#) is displayed.

Select either one to inspect:

- ◊ The report file of the simulation: <tri-diag.*>.
- ◊ The print file with printed output results: <tri-prt.*>.
- ◊ The report file of the WAVE simulation (if applicable): <swn-diag.*>.
- ◊ The report file of the Hydrodynamic coupling (if applicable): <couplnef.out>.
- ◊ A random file, upon selecting this option an editor window is opened that allows you to inspect or modify a file without having to leave the FLOW-GUI.

The report file <tri-diag.*> contains information on the total simulation time, the time spent in certain parts of the simulation and an indication of the performance (seconds processor time per grid point per time step). In the next section we shall give some guidelines to estimate the simulation time and the required disk space for the various result files.

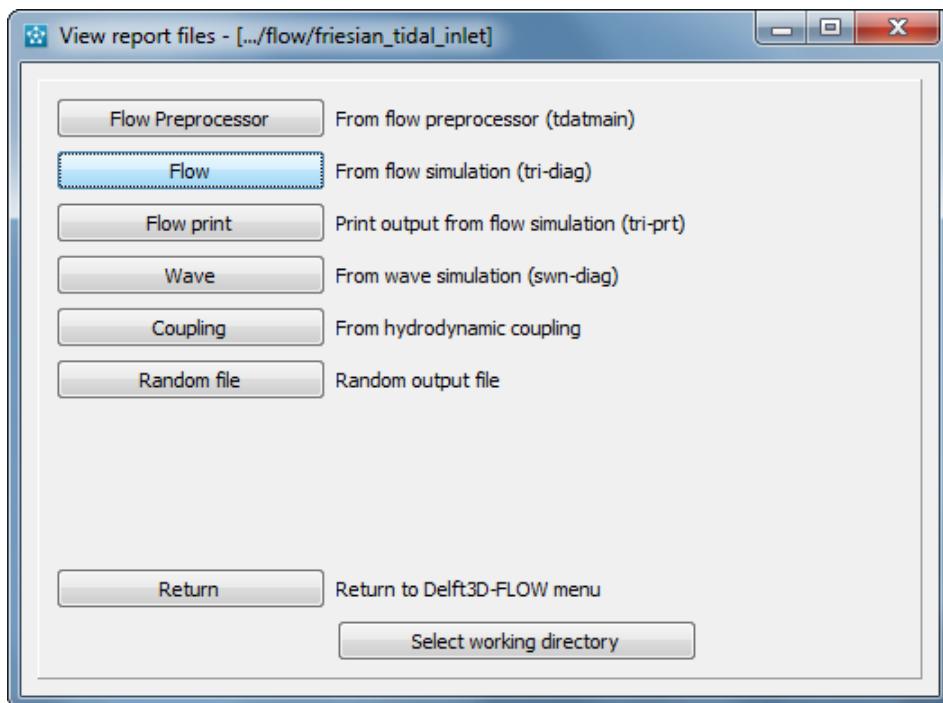


Figure 6.4: Select a report file for inspection

6.1.3 Running a scenario using a batch script

Batch scripts can be generated in two ways:

- ◊ Delft3D-MENU can generate batch scripts (not for all functionality)
- ◊ Copy an existing script and adapt it for your purpose. Examples are available at <https://svn.oss.deltares.nl/repos/delft3d/trunk/examples>

Separate scripts are needed for Windows (with extension "bat") and Linux (with extension "sh"). See [section 6.4](#) for the command-line arguments.

The easiest Windows script (assuming Delft3D is installed properly, assuming a correct <config_d_hydro.xml> is available, compare with example "01_standard"):

```
set PATH=%D3D_HOME%\%ARCH%\flow2d3d\bin;%PATH%
%D3D_HOME%\%ARCH%\flow2d3d\bin\d_hydro.exe config_d_hydro.xml
```

The easiest Linux script (assuming Delft3D is installed properly, assuming a correct <config_d_hydro.xml> is available, compare with example "01_standard"):

```
export LD_LIBRARY_PATH=$D3D_HOME/$ARCH/flow2d3d/bin:$LD_LIBRARY_PATH
$D3D_HOME/$ARCH/flow2d3d/bin/d_hydro.exe config_d_hydro.xml
```

6.2 Run time

The actual run time of a model can vary considerably depending on a variety of factors such as:

- ◊ The problem being solved, characterised by the number of active grid points, the number of layers in the vertical or the number of processes taken into account.

Table 6.1: Simulation performance on different operating systems

Computation	WIN	LINUX
Without Domain decomposition	0.5E-05	0.4E-05
With Domain decomposition	0.5E-04 to 0.1E-03	0.5E-04 to 0.1E-03

- ◊ The length of the simulation in time and the time step being used.
- ◊ The hardware configuration that is used and the work load of the processor.

For this reason, only some general considerations are given to determine the run time of a hydrodynamic simulation. On a PC or a workstation without separate I/O-processors the CPU time is the sum of the processor time and the I/O time.

The *processor time* required for a simulation is primarily determined by:

- ◊ The model definition, i.e. the number of active grid points and the number and type of the processes taken into account.
- ◊ The length of the simulated period in terms of the number of time steps executed.

The *I/O time* is determined by:

- ◊ The number of times the computed data are written to history, map, print and communication files.
- ◊ The number of observation points, cross-sections and the number of output parameters.

The simulation performance is defined as the CPU time per grid point per time step per constituent:

$$\text{simulation performance} = \frac{\text{CPU time}}{N \cdot M_{\max} \cdot N_{\max} \cdot K_{\max} \cdot L_{\max}} \text{ [system seconds]}$$

where:

- | | |
|------------------|---|
| N | is the number of time steps executed |
| M _{max} | is the number of grid points in <i>x</i> -direction |
| N _{max} | is the number of grid points in <i>y</i> -direction |
| K _{max} | is the number of layers in the vertical |
| L _{max} | is the number of constituents |

In **Table 6.1** the simulation performance is given for a PC with a Windows and a Linux operating system respectively. The Windows and Linux PC have similar processors (AMD Athlon 64 X2 Dual Core Processor 4200+, 2.19GHz).

Remarks:

- ◊ The simulation performance is almost linearly dependent on the processor speed.
- ◊ The simulation performance can be improved with about 20 % by minimizing the output written to the result files.
- ◊ The simulation performance is written to the diagnosis file at the end of a calculation.



- ◊ When using Domain decomposition, each domain produces its own simulation performance index. The difference between these indexes will become very large when the domain load is unbalanced.
- ◊ To improve the time estimates we suggest to keep record of the simulation performance on your hardware environment including all site specific aspects. In case the simulation performance is very bad on similar hardware and the workload is not excessively high, you are advised to contact your system manager to check the general settings of your hardware such as swap and stack space. For system requirements you are referred to the Installation Manual.

6.3 Files and file sizes

For estimating the required disk space the following files are important:

- ◊ history file
- ◊ map file
- ◊ print file
- ◊ communication file
- ◊ drogues file



Remark:

- ◊ Drogues files are generally small and are not discussed further.

6.3.1 History file

The size of the history file is determined by:

- ◊ The number of monitoring points (observation points + cross-sections): H1.
- ◊ The number of quantities stored: H2.
- ◊ The number of additional process parameters, such as salinity, temperature, constituents and turbulence quantities, taken into account in the simulation: H3.
- ◊ The number of time the history data is written to the history file: H4.

You can estimate the size of a history file (in bytes) from the following equation:

$$\text{size history file} = H1 \cdot (H2 + H3) \cdot H4 \cdot 4 \text{ bytes.}$$

As a first approximation you can use H2 = 15.

Example

For a 2D simulation with density driven currents (salinity and temperature), a simulated period of 12 hrs 30 min, a time integration step of 5 minutes, 30 monitoring points and each time step being stored, the size of the history file will be of the order of 360 kBytes. For the same model but now with 10 layers in the vertical the file size will increase to about 4 MBytes. These estimates show that history files are rather small. Unless the number of monitoring points is excessively large the history files are typically much smaller than the map output files.

6.3.2 Map file

The size of the map file is determined by:

- ◊ The size of the model, i.e. the number of active grid cells multiplied by the number of layers ($M_{max} \cdot N_{max} \cdot K_{max}$): M1.
- ◊ The number of quantities stored: M2.
- ◊ The number of process parameters taken into account, such as salinity, temperature, constituents and turbulence quantities: M3.
- ◊ The number of time steps for which the map file is written: M4.

Remark:



- ◊ For a more refined estimate you should distinguish between parameters that depend or not on the number of layers used (such as the water level). For a 3D simulation the latter quantities can be neglected, for a 2D simulation they must be accounted for. As a first estimate we double the number of quantities M2 in a 2D simulation.

As a first approximation you can use $M_2 = 15$ for a 3D simulation and $M_2 = 30$ for a 2D simulation.

You can estimate the size of a map file (in bytes) from the following equation:

$$\text{size map file} = M_1 \cdot (M_2 + M_3) \cdot M_4 \cdot 4 \text{ bytes.}$$

Example

For a 3D simulation with 50 by 50 points and 5 layers, simulated with density driven currents (salinity and temperature), simulation results stored for a period of 12 hours and 30 minutes, and the file is written with an interval of 30 minutes the size of the map file will be about 31 MBytes. For larger models the map file can easily become excessively large, as result the map file is less frequently written, for instance every 2 or 3 hours.

6.3.3 Print file

The print file is a formatted *ASCII* file with a size that can be large, varying from hundreds of Kbytes to hundreds of Mbytes. Its size depends on:

- ◊ The number of grid points: P1.
- ◊ The number of layers in the vertical: Kmax.
- ◊ The print flags for the map data requested: P2.
- ◊ The number of times the map data is written to the file: P3.
- ◊ The number of monitoring stations: P4.
- ◊ The print flags for the history data requested: P5.
- ◊ The number of times the history data is written to the file: P6.

You can estimate the size of a print file (in bytes) from the following equation:

$$\text{size print file} = 2 \cdot (P_1 \cdot K_{max} \cdot P_2 \cdot P_3 + P_4 \cdot P_5 \cdot P_6) \cdot 4 \text{ bytes}$$

The factor of 2 is applied for all explaining text included in the file.

Example

For a similar simulation as in the former section and all print flags checked, i.e. roughly 15 items, the print file is found to be in the order of 36 MBytes; its size is found to be determined by the map data.

6.3.4 Communication file

The size of the communication or COM-file (e.g. for the other Delft3D modules, such as the water quality module WAQ) from the hydrodynamic simulation is determined by:

- ◊ The number of grid cells in horizontal and vertical direction, i.e. $M_{max} \cdot N_{max} \cdot K_{max}$: C1.
- ◊ The number of quantities stored: C2.
- ◊ The number of process parameters taken into account, such as salinity, temperature, constituents and turbulence quantities: C3.
- ◊ The number of time steps, for which the communication file is written: C4.

You can estimate the size of a communication file (in bytes) from the following formula:

$$\text{size communication file} = C1 \cdot (C2 + C3) \cdot C4 \cdot 4 \text{ bytes.}$$

As a first approximation you can use C2 = 15.

Example

A COM-file size of 20 MBytes should be expected for a model containing 50 by 50 points by 5 layers, simulated with density driven currents and simulation results stored for a period of 12 hrs 30 min, and the file is written with an interval of 15 minutes.

For a more refined estimate you should account for parameters that are stored only once and parameters that are stored for each layer. However, the contribution of the latter to the file size easily dominates by far the contribution of the other parameters.

**Remark:**

- ◊ The file sizes given are indicative and the figures may not be linearly extrapolated to determine the file size when the number of grid points is enlarged.

6.4 Command-line arguments

When using Delft3D-MENU to start a Delft3D-FLOW calculation, an XML-formatted configuration file will be generated automatically named <config_flow2d3d.xml>. The calculation itself is started by executing:

```
d_hydro.exe <XML config file>
```

With:

```
d_hydro.exe      The name of the executable to be started  
<XML config file> The (only) command-line argument with default name <config_flow2d3d.xml>
```

When using a script to start a Delft3D-FLOW calculation, the XML-formatted configuration file has to be generated manually. There are three ways to do this:

- 1 Use one of the examples at <https://svn.oss.deltares.nl/repos/delft3d/trunk/examples> as a start, both for example configuration files and for run scripts.

- 2 Use the tcl script <.../menu/bin/create_config_xml.tcl> inside the Delft3D installation directory.
- 3 Use a text editor and write the file from scratch. Use the following (minimum) example as a start:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<deltaresHydro xmlns="http://schemas.deltares.nl/deltaresHydro"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://schemas.deltares.nl/deltaresHydro
        http://content.oss.deltares.nl/schemas/d_hydro-1.xsd">

    <control>
        <sequence>
            <start>myNameFlow</start>
        </sequence>
    </control>
    <flow2D3D name="myNameFlow">
        <library>flow2d3d</library>
        <mdfFile>f34.mdf</mdfFile>
    </flow2D3D>
</deltaresHydro>
```

Remarks:



- ◊ The automatically generated XML configuration file contains comments about optional parameters.
- ◊ The automatically generated XML configuration file itself is generated using the script <.../menu/bin/create_config_xml.tcl> mentioned above.
- ◊ The site <http://oss.deltares.nl/web/delft3d> contains a lot of information about running a calculation.

6.5 Frequently asked questions

This chapter aims to help you with common questions that may arise while using Delft3D-FLOW.

1 Question

What to do if the message *No convergence in UZD* appear often in the diagnostic file

Answer

In general this message appear in the beginning of your simulation and it means that the Red-Black Jacobi iteration is not converged. If this iteration process is not converged the computed momentum has not reached the desired accuracy, after the maximum number of iterations (i.e. hard coded on 50 iterations). As *no convergence* is reported for the first few time steps (tidal cycles) there is no need to adjust the input parameters of your simulation.

The simulations of Delft3D-FLOW are boundary driven and the mismatch of momentum should disappears after a few tidal cycles. But if the message *No convergence in UZD* still appears after a few tidal cycles you have to decrease the timestep.

2 Question

What to do if the message *Water level gradient too high > x m (per 0.5 DT) after nst timesteps in the following points: m, n* appear in the diagnostic file

Answer

In general this message appear in the beginning of your simulation and it means that the water level changes are too high for the given time step (*nst*) and at location (*m, n*).

The simulations of Delft3D-FLOW are boundary driven and the mismatch should disappears after a few tidal cycles. But if the message *Water level gradient too high ...* still appears after a few tidal cycles you have to decrease the timestep.

DRAFT

7 Visualise results

7.1 Introduction

Visualisation and animation of results is an essential part of presenting the results of your project. In Delft3D-FLOW you have several options:

- ◊ Online Visualisation: to inspect your results during the computation, see [subsection 4.5.12](#).
- ◊ GPP: to display and print a large variety of graphs, store the defined sets of figures in a session file and use the same definitions in a next computation.
- ◊ Delft3D-QUICKPLOT: to display quickly and print a large variety of graphs, save the defined sets of figures, etc.
- ◊ Delft3D-MATLAB¹ interface: to load, process or display your results in the MATLAB environment.
- ◊ GISVIEW interface: to load, process or display your results in the ArcGIS² environment.

The interfaces to MATLAB and ArcGIS are additional utilities in Delft3D and must be acquired separately.

In the following sections we will briefly discuss each of these options; for full details you are referred to the respective User Manuals.

7.2 Working with GPP

GPP offers a comprehensive selection and plotting facility to visualise or animate simulation results, to import and visualise other data such as measurements, or to export selected data sets of the results for use in other programs. You can define a single figure or a set of figures and inspect it on screen or make a hardcopy of it on one of the supported hard copy devices. The figures can be processed in an interactive manner or in the background (batch) mode. In this section we only give a very concise description of the post-processor. For a detailed description of its use and functionalities you are referred to the GPP User Manual.

7.2.1 Overview

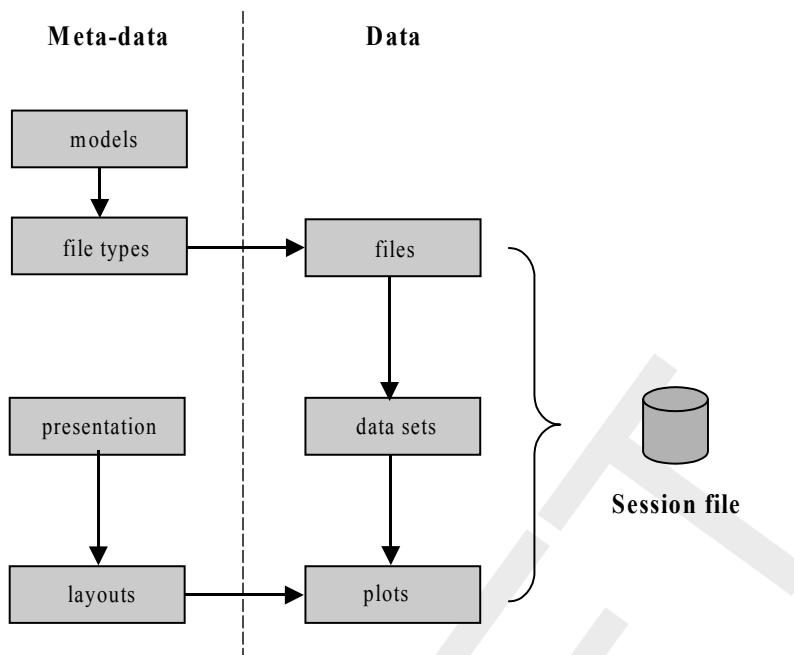
When executing a project with many simulations the amount of data from which a set will be visualised can be enormously large; also the files in which the data are stored can be very large. Therefore it is not optimal to search the original result files over and over again for each parameter or for each new figure. GPP, instead, provides a mechanism to make a selection of the various results and parameters before starting the actual visualisation process and makes a kind of reference list to these sets of data. Next you can define one or more graphs and fill them with data from these data sets. As GPP knows where to find this data, retrieving the data is executed very efficiently. This efficiency is further increased by the option to select all observation points for a certain quantity or to select all time instances at which a quantity is stored in the map file and let you make the final selection when producing the figure.

GPP has access to the communication file and to the result files of all Delft3D modules and in fact to many other programs of Deltares, so you can combine almost any kind of data in a figure.

GPP uses a certain hierarchy in the data and the meta-data, see [Figure 7.1](#).

¹ MATLAB is software for PC desktop and is a trademark of The MathWorks, Inc, Natick, MA, USA.

² ArcGIS is GIS-software for desktop and is a trademark of ESRI, Inc. Redlands, CA, USA.

**Figure 7.1: Hierarchy of GPP**

We distinguish meta-data to specify the definitions of a figure at a high level of abstraction (left part of [Figure 7.1](#)) and the actual data (right part of [Figure 7.1](#)).

Meta-data:

models	Defines the set of models the results of which can be visualised. You can change this set of models to limit the options presented in GPP menus.
file types	Defines the set of file types that can be used. You can change this set of file types to limit the options presented in GPP menus.
presentations	Defines the set of data-presentation methods, such as contour maps or xy-graphs.
layouts	Defines the set of layouts that can be used in a figure. You specify the general set-up of a figure by defining the appropriate layout, the size of the graph, the plot areas, their position, additional text etc.

Data:

files	The actual files to be used in your plot session.
data sets	The actual data sets selected from the files and to be used in the visualisation.
plots	The actual figures, including the data, which will be presented on your monitor or printed on paper.
session file	An ASCII file containing all information that defines the figures. For the data only the references to the data is stored in the session file, not the data itself.

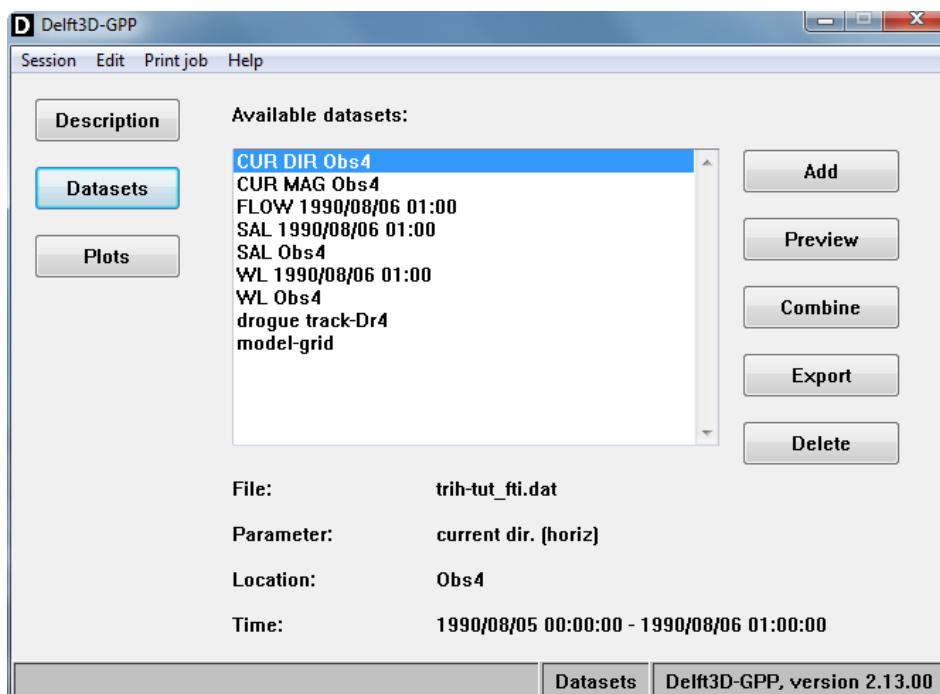


Figure 7.2: Main window of GPP

7.2.2 Launching GPP

To start GPP:

- Select from the Delft3D-MENU *FLOW - GPP* and next [Figure 7.2](#) is displayed.

The basic functions are shortly described below; for full details you are referred to the GPP User Manual.

<i>Session</i>	To load an existing session file or to save the settings and selections of the current session in a session file for later use.
<i>Description</i>	To give a short description of a session file; this information is used only for reference.
<i>Datasets</i>	List of pre-selected data sets to be used in the current plot session. You can give selected data sets a useful name. At start-up the selections are displayed of the previous plot session in the current directory.
<i>Plots</i>	List of pre-selected plot layouts to be used in the current plot session. At start-up the selections are displayed of the previous plot session in the current directory.
<i>Add</i>	To add a data set or plot layout, depending which function on the left side of the list box has been selected.
<i>Preview</i>	To preview a selected data set or plot layout from the list displayed in the list box.
<i>Combine</i>	To combine any of the available (single) data sets to a new data set, such as multiply, divide, take the maximum value etc. and save the new data set under a unique name.
<i>Export</i>	To export the selected single or combined data set to an ASCII file or GIS-file (for single data sets only).
<i>Delete</i>	To delete the selected data set or plot layout.

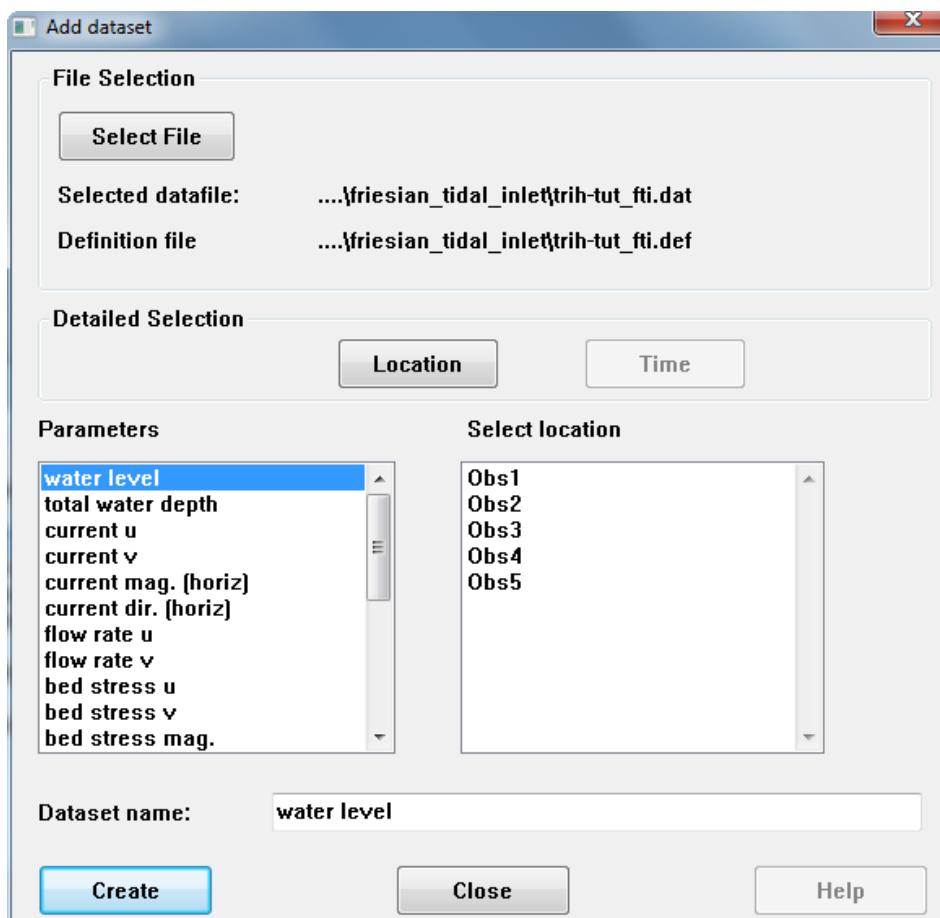


Figure 7.3: Parameters and locations in the <trih-tut_fti.dat> file

To add a data set of a specific result file to the *Available data sets* in [Figure 7.2](#):

- Select *Datasets - Add* in [Figure 7.2](#).
- Click *Select File* in the **Add dataset** window, [Figure 7.3](#).
- Select the required data file in the file selection window that is being displayed.

The parameters and locations (or time in case of map-results) available in a selected result file are displayed in the *Add dataset* window, see [Figure 7.3](#).

You can make as many selections from a specific result file, or from different results files (to combine results from different computations or models) as you like.

To have a quick view on a data set:

- Select in [Figure 7.2](#) the required data set and click *Preview*.

The selected data set and a default plot layout will be displayed in the **Plot window**, see [Figure 7.3](#).



Remark:

- ◊ GPP recognises the type of data selected and uses an appropriate default presentation method to display the results.

In [Figure 7.4](#) some of the options are displayed to change attributes of the figure.

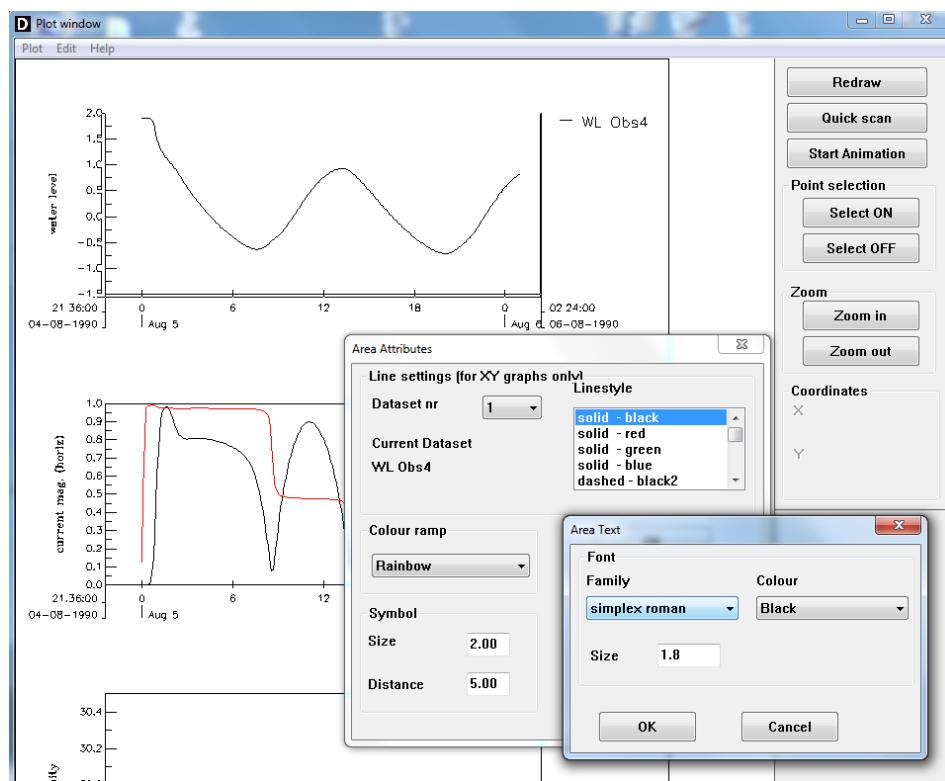


Figure 7.4: Some options to change the plot attributes

You are referred to the User Manual of GPP for full details on how to use GPP.

7.3 Working with Delft3D-QUICKPLOT

A graphical interface, Delft3D-QUICKPLOT, see [Figure 7.5](#), is provided to make a basic graphical representation of the data or make a quick animation. The interface can be started from the Delft3D-MENU or by typing d3d_qp at the MATLAB command prompt.

The interface allows to open the binary Delft3D output files (both NEFIS and NetCDF) and to select data fields. After making a selection of time steps, stations and (m, n, k) indices, you can either load the data (*Open a data file* button) or plot the data (*Quick View* button). See the Delft3D-QUICKPLOT User Manual for full details and a description of the routines and their use.

7.4 GISVIEW interface

To access, visualise and process Delft3D map-results in a GIS-environment, an interface between Delft3D and ArcGIS was developed. This interface is provided as an extension to ArcGIS. From a user point of view three menu-options are added to the standard menu-bar of ArcGIS.

For full details of the GISVIEW interface you are referred to the GISVIEW User Manual ([Delft3D-GISVIEW UM, 2013](#)).

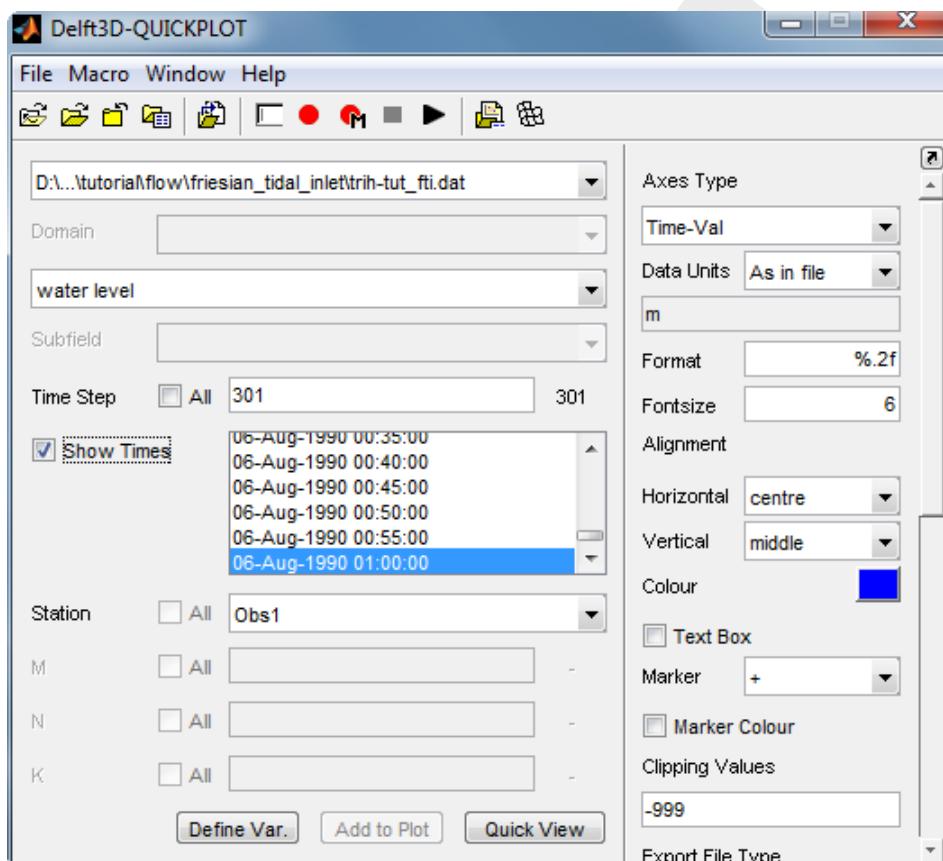


Figure 7.5: Delft3D-QUICKPLOT interface to Delft3D result

8 Manage projects and files

8.1 Introduction

When executing a project generally a large number of files are created. Typically, some of these files have a rather general use and will not or hardly change during the project, such as the curvilinear grid, while other files may change appreciably for each simulation, such as the bed stress or wind field during a calibration phase. Organising the files in your project in a logical sense is important to keep track of all simulations and changes you have investigated. We suggest to organise your project directory along the hierarchy of steps typical for your project.

A typical hierarchy of steps for a hydrodynamic – water quality assessment study might be:

Hydrodynamics:

- ◊ Inspection and analysis of the problem to be solved, of the available data and information of the area and of the processes involved.
- ◊ Setting up of the hydrodynamic model, i.e. definition of the model extent, the grid size, the number of layers (in case of a 3D simulation), generation of the land boundary outline and the curvilinear grid.
- ◊ Collecting and processing the bathymetric data and generating the model bathymetry.
- ◊ Preparing boundary conditions.
- ◊ Processing field measurements for calibration and verification data in stations, cross-sections and verticals, both as time-series and as maps.
- ◊ Defining a couple of simulations to make a first assessment of the model behaviour.
- ◊ Defining the calibration runs to be executed.
- ◊ Accessing the results, performing a (in most cases several) sensitivity analysis and iterating on the calibration process if necessary.
- ◊ Determining and executing the verification runs and accessing the final accuracy of the simulations.
- ◊ Determining the simulation period to be stored for the water quality study and generating the hydrodynamic database for the water quality simulations.

Water quality:

- ◊ A similar set of steps can be defined for the water quality assessment study.
- ◊ Determining if the same or a coarser grid shall be used for the water quality simulations and transfer the hydrodynamics defined on the hydrodynamic grid to the water quality grid.
- ◊ Determining the water quality processes and parameters that must be involved.

Remark:

- ◊ The number of water quality computations can be very large due to the number of processes that must be simulated and the number of parameters that must be investigated in a sensitivity analysis.



This hierarchy of steps, the related computations each with its specific input, attribute and results files must be organised in some way to control the large amounts of files. This is necessary to guarantee the quality of the results and the consistency between input data and result.

8.1.1 Managing projects

It is of utmost importance to structure your project in a logical way. This will make it easy to understand its structure by colleagues working on the same project. Furthermore, it helps to retrieve results when after some time you have to answer additional questions.

There are many ways to structure a project and there is probably not just one best way. The structure you want to apply may depend on company rules or personal preferences. Below we give some general ideas to structure your projects.

To manage your projects you might apply the following steps:

- ◊ Define for each project a separate directory or file system.
- ◊ Organise the directory in-line with the various steps mentioned in the introduction, i.e. prepare sub-directories for preparing the boundary conditions, the field data, the bathymetry, the grid, etc. and give them useful names.
- ◊ Execute each dominant step in the project, such as preliminary assessment, calibration and verification in separate sub-directories and execute sensitivity studies within these steps in sub-sub-directories.



Remark:

- ◊ Never combine the program files with project data; define a project directory.

Currently, the FLOW Graphical User Interface (GUI) only supports short file names in a working directory, i.e. all files used in a scenario must be in the current working directory. Organising your project as sketched above means that many unchanged files must be copied to each (sub-)directory and so will reside many times on your file system, thereby increasing your disk requirements. However, the alternative of 'all scenarios in one and the same directory' is not attractive either as you easily will have hundreds of files. You are advised to prefer the structuring of your simulations above optimising the amount of disk space used for the attribute files. This problem is solved to a large extent if the FLOW-GUI can handle long file names and paths. This option will be available in a next release.

A probably more urgent problem is to handle the large result files; they may easily result in full file systems. An option in case of a full file system is to move the large result files to a background storage facility and only load them back on the file system when necessary. The result files are usually the real big ones so the storage problem is relieved quite a bit by moving these files. Online compression is only a partial solution as a typical compression factor for result files is only 2 or 3. Fortunately, the costs of mass storage devices are decreasing at a rapid rate.

8.1.2 Managing files

The length of the MDF-file name is limited to 256 characters, and has the form <*runid.mdf*>. Do not use blanks in the runid.

9 Conceptual description

9.1 Introduction

Increasing awareness of environmental issues has focused the attention of scientists and engineers on the problem of predicting the flow and dispersion of contaminants in water systems. Reliable information on water flow, waves, water quality, sediment transport and morphology can be obtained from appropriate mathematical models. In general the first step in such modelling activities concerns the simulation of the flow itself. Whether the problem is related, for example, to the stability of a hydraulic structure, to salt intrusion, to the dispersion of pollutants or to the transport of silt and sediment, flow simulations usually form the basis of the investigations to be carried out.

Delft3D is the integrated flow and transport modelling system of Deltares for the aquatic environment. The flow module of this system, viz. Delft3D-FLOW, provides the hydrodynamic basis for other modules such as water quality, ecology, waves and morphology. For steady and non-steady modelling of the far-field water quality and ecology, it is coupled with the far-field water quality module D-Water Quality. Non-steady modelling of the mid-field water quality is performed by coupling the flow module to the particle tracking simulation module D-Waq PART. For the interaction between waves and currents the flow module may be coupled with the short-waves model Delft3D-WAVE.

In the vertical direction Delft3D-FLOW offers two different vertical grid systems: a so-called σ co-ordinate system (σ -model) introduced by [Phillips \(1957\)](#) for ocean models and the Cartesian Z co-ordinate system (Z -model).

This chapter gives some background information on the conceptual model of the Delft3D-FLOW module. The numerical algorithms are described in [chapter 10](#). Most of the concepts and algorithms are applicable to both the σ -model and Z -model. However, the specifics of the Z co-ordinate system will be presented in more detail in [chapter 12](#).

9.2 General background

9.2.1 Range of applications of Delft3D-FLOW

The hydrodynamic module Delft3D-FLOW simulates two-dimensional (2DH, depth-averaged) or three-dimensional (3D) unsteady flow and transport phenomena resulting from tidal and/or meteorological forcing, including the effect of density differences due to a non-uniform temperature and salinity distribution (density-driven flow). The flow model can be used to predict the flow in shallow seas, coastal areas, estuaries, lagoons, rivers and lakes. It aims to model flow phenomena of which the horizontal length and time scales are significantly larger than the vertical scales.

If the fluid is vertically homogeneous, a depth-averaged approach is appropriate. Delft3D-FLOW is able to run in two-dimensional mode (one computational layer), which corresponds to solving the depth-averaged equations. Examples in which the two-dimensional, depth-averaged flow equations can be applied are tidal waves, storm surges, tsunamis, harbour oscillations (seiches) and transport of pollutants in vertically well-mixed flow regimes.

Three-dimensional modelling is of particular interest in transport problems where the horizontal flow field shows significant variation in the vertical direction. This variation may be generated by wind forcing, bed stress, Coriolis force, bed topography or density differences. Examples are dispersion of waste or cooling water in lakes and coastal areas, upwelling and downwelling of nutrients, salt intrusion in estuaries, fresh water river discharges in bays and

thermal stratification in lakes and seas.

9.2.2 Physical processes

The numerical hydrodynamic modelling system Delft3D-FLOW solves the unsteady shallow water equations in two (depth-averaged) or in three dimensions. The system of equations consists of the horizontal equations of motion, the continuity equation, and the transport equations for conservative constituents. The equations are formulated in orthogonal curvilinear co-ordinates or in spherical co-ordinates on the globe. In Delft3D-FLOW models with a rectangular grid (Cartesian frame of reference) are considered as a simplified form of a curvilinear grid. In curvilinear co-ordinates, the free surface level and bathymetry are related to a flat horizontal plane of reference, whereas in spherical co-ordinates the reference plane follows the Earth's curvature.

The flow is forced by tide at the open boundaries, wind stress at the free surface, pressure gradients due to free surface gradients (barotropic) or density gradients (baroclinic). Source and sink terms are included in the equations to model the discharge and withdrawal of water.

The Delft3D-FLOW model includes mathematical formulations that take into account the following physical phenomena:

- ◊ Free surface gradients (barotropic effects).
- ◊ The effect of the Earth's rotation (Coriolis force).
- ◊ Water with variable density (equation of state).
- ◊ Horizontal density gradients in the pressure (baroclinic effects).
- ◊ Turbulence induced mass and momentum fluxes (turbulence closure models).
- ◊ Transport of salt, heat and other conservative constituents.
- ◊ Tidal forcing at the open boundaries.
- ◊ Space and time varying wind shear-stress at the water surface.
- ◊ Space varying shear-stress at the bottom.
- ◊ Space and time varying atmospheric pressure on the water surface.
- ◊ Time varying sources and sinks (e.g. river discharges).
- ◊ Drying and flooding of tidal flats.
- ◊ Heat exchange through the free surface.
- ◊ Evaporation and precipitation.
- ◊ Tide generating forces.
- ◊ Effect of secondary flow on depth-averaged momentum equations.
- ◊ Lateral shear-stress at wall.
- ◊ Vertical exchange of momentum due to internal waves.
- ◊ Influence of waves on the bed shear-stress (2D and 3D).
- ◊ Wave induced stresses (radiation stress) and mass fluxes.
- ◊ Flow through hydraulic structures.
- ◊ Wind driven flows including tropical cyclone winds.

9.2.3 Assumptions underlying Delft3D-FLOW

In Delft3D-FLOW the 2D (depth-averaged) or 3D non-linear shallow water equations are solved. These equations are derived from the three dimensional Navier-Stokes equations for incompressible free surface flow. The following assumptions and approximations are applied:

- ◊ In the σ co-ordinate system the depth is assumed to be much smaller than the horizontal length scale. For such a small aspect ratio the shallow water assumption is valid, which means that the vertical momentum equation is reduced to the *hydrostatic pressure* relation. Thus, vertical accelerations are assumed to be small compared to the gravitational acceleration and are therefore not taken into account. When this assumption is not valid then Delft3D provides an option to apply the so-called Non-hydrostatic pressure model in the Z -model. For details, we refer to [chapter 12](#) and [section B.12](#) of this User Manual.
- ◊ The effect of variable density is only taken into account in the pressure term (Boussinesq approximation).
- ◊ In the σ co-ordinate system, the immediate effect of buoyancy on the vertical flow is not considered. In Delft3D-FLOW vertical density differences are taken into account in the horizontal pressure gradients and in the vertical turbulent exchange coefficients. So the application of Delft3D-FLOW is restricted to mid-field and far-field dispersion simulations of discharged water.
- ◊ For a dynamic online coupling between morphological changes and flow the 3D sediment and morphology feature is available.
- ◊ In a Cartesian frame of reference, the effect of the Earth's curvature is not taken into account. Furthermore, the Coriolis parameter is assumed to be uniform unless specifically specified otherwise.
- ◊ In spherical co-ordinates the inertial frequency depends on the latitude.
- ◊ At the bottom a slip boundary condition is assumed, a quadratic bottom stress formulation is applied.
- ◊ The formulation for the enhanced bed shear-stress due to the combination of waves and currents is based on a 2D flow field, generated from the velocity near the bed using a logarithmic approximation.
- ◊ The equations of Delft3D-FLOW are capable of resolving the turbulent scales (large eddy simulation), but usually the hydrodynamic grids are too coarse to resolve the fluctuations. Therefore, the basic equations are Reynolds-averaged introducing so-called Reynolds stresses. These stresses are related to the Reynolds-averaged flow quantities by a turbulence closure model.
- ◊ In Delft3D-FLOW the 3D turbulent eddies are bounded by the water depth. Their contribution to the vertical exchange of horizontal momentum and mass is modelled through a vertical eddy viscosity and eddy diffusivity coefficient (eddy viscosity concept). The coefficients are assumed to be proportional to a velocity scale and a length scale. The coefficients may be specified (constant) or computed by means of an algebraic, $k-L$ or $k-\varepsilon$ turbulence model, where k is the turbulent kinetic energy, L is the mixing length and ε is the dissipation rate of turbulent kinetic energy.
- ◊ In agreement with the aspect ratio for shallow water flow, the production of turbulence is based on the vertical (and not the horizontal) gradients of the horizontal flow. In case of small-scale flow (partial slip along closed boundaries), the horizontal gradients are included in the production term.
- ◊ The boundary conditions for the turbulent kinetic energy and energy dissipation at the free surface and bottom assume a logarithmic law of the wall (local equilibrium).
- ◊ The eddy viscosity is an-isotropic. The horizontal eddy viscosity and diffusivity coefficients should combine both the effect of the 3D turbulent eddies and the horizontal motions that cannot be resolved by the horizontal grid. The horizontal eddy viscosity is generally much larger than the vertical eddy viscosity.
- ◊ For large-scale flow simulations, the tangential shear-stress at lateral closed boundaries can be neglected (free slip). In case of small-scale flow partial slip is applied along closed

- boundaries.
- ◊ For large-scale flow simulations, the horizontal viscosity terms are reduced to a bi-harmonic operator along co-ordinate lines. In case of small-scale flow the complete Reynold's stress tensor is computed. The shear-stress at the side walls is calculated using a logarithmic law of the wall.
 - ◊ In the σ co-ordinate system, Delft3D-FLOW solves the so-called long wave equation. The pressure is hydrostatic and the model is not capable of resolving the scales of short waves. Therefore, the basic equations are averaged in analogy with turbulence introducing so-called radiation stresses. These stresses are related to the wave quantities of Delft3D-WAVE by a closure model.
 - ◊ It is assumed that a velocity point is set dry when the actual water depth is below half of a user-defined threshold. If the point is set dry, then the velocity at that point is set to zero. The velocity point is set wet again when the local water depth is above the threshold. The hysteresis between drying and flooding is introduced to prevent drying and flooding in two consecutive time steps. The drying and flooding procedure leads to a discontinuous movement of the closed boundaries at tidal flats.
 - ◊ A continuity cell is set dry when the four surrounding velocity points at the grid cell faces are dry or when the actual water depth at the cell centre is below zero (negative volume).
 - ◊ The flux of matter through a closed wall and through the bed is zero.
 - ◊ Without specification of a temperature model, the heat exchange through the free surface is zero. The heat loss through the bottom is always zero.
 - ◊ If the total heat flux through the water surface is computed using a temperature excess model the exchange coefficient is a function of temperature and wind speed and is determined according to [Sweers \(1976\)](#). The natural background temperature is assumed constant in space and may vary in time. In the other heat flux formulations the fluxes due to solar radiation, atmospheric and back radiation, convection, and heat loss due to evaporation are modelled separately.
 - ◊ The effect of precipitation on the water temperature is accounted for.

9.3 Governing equations

In this section, we will present in detail the governing equations. Below an overview of the symbols that are used in the equations is presented.

List of Symbols

Symbol	Units	Meaning
$A^{(m,n)}$	m^2	area of computational cell at location (m, n)
\hat{A}_δ	m	near-bed peak orbital excursion
a	m	reference height as defined by Van Rijn
α	-	artificial compression coefficient
B_k	m^2/s^3	buoyancy flux term in transport equation for turbulent kinetic energy
B_e	m^2/s^4	buoyancy flux term in transport equation for the dissipation of turbulent kinetic energy
C, C_{2D}	$\text{m}^{1/2}/\text{s}$	2D Chézy coefficient
C_{3D}	$\text{m}^{1/2}/\text{s}$	3D Chézy coefficient
C_d	-	wind drag coefficient
c	kg/m^3	mass concentration
c_a	kg/m^3	mass concentration at reference height a

Symbol	Units	Meaning
c_D	-	constant relating mixing length, turbulent kinetic energy and dissipation in the $k-\varepsilon$ model
c_p	J/(kg°C)	specific heat of sea water
c_μ	-	calibration constant
c'_μ	-	constant in Kolmogorov-Prandtl's eddy viscosity formulation
D	kg/s ³	dissipation rate wave energy
D	kg/(m ² s)	deposition rate cohesive sediment
$DPSED$	m	depth of sediment available at the bed
D_*	-	non-dimensional particle diameter
d	m	depth below some horizontal plane of reference (datum)
D_s	m	representative diameter of suspended sediment
D_{50}	m	median diameter of sediment
D_{90}	m	sediment diameter
E	m/s	evaporation
E	kg/(m ² s)	erosion rate cohesive sediment
e_a	mbar	vapour pressure at a given air temperature
e_s	mbar	saturated vapour pressure at a given temperature
F_c	-	cloudiness factor
F_x	m/s ²	radiation stress gradient in x -direction
F_y	m/s ²	radiation stress gradient in y -direction
F_ξ	m/s ²	turbulent momentum flux in ξ -direction
F_η	m/s ²	turbulent momentum flux in η -direction
f	1/s	Coriolis parameter (inertial frequency)
$f(F_c)$	-	attenuation function for the solar and atmospheric (long wave) radiation
f_{FIXFAC}	-	fixed layer proximity factor
f_{MORFAC}	-	user-defined morphological acceleration factor
f_w	-	total wave friction factor
f'_w	-	grain-related wave friction factor
f_{wind}	-	dependency of λ of the wind speed in temperature excess model
$\sqrt{G_{\xi\xi}}$	m	coefficient used to transform curvilinear to rectangular coordinates
$\sqrt{G_{\eta\eta}}$	m	coefficient used to transform curvilinear to rectangular coordinates
g	m/s ²	acceleration due to gravity
H	m	total water depth ($H = d + \zeta$; d positive downward)
H_{rms}	m	root-mean-square wave height
I	m/s	spiral motion intensity (secondary flow)
I_{be}	m/s	equilibrium intensity of spiral motion due to curvature of stream lines
I_{ce}	m/s	equilibrium intensity of spiral motion due to Coriolis
k	m ² /s ²	turbulent kinetic energy
k_a	m	increased (apparent) bed roughness height felt by the current in the presence of waves
k_s	m	Nikuradse roughness length
$k_{s,c}$	m	current related bed roughness height
k_x, k_y	1/m	wave number
$k_{s,w}$	m	wave related bed roughness height
L	m	mixing length

Symbol	Units	Meaning
ℓ	-	index number of sediment fraction
M_x^S	kg m/s	depth-averaged mass flux due to Stokes drift in x -direction
M_y^S	kg m/s	depth-averaged mass flux due to Stokes drift in y -direction
M_ξ	m/s ²	source or sink of momentum in ξ -direction
M_η	m/s ²	source or sink of momentum in η -direction
n	m ^{-1/3} s	Manning's coefficient
P	kg/(ms ²)	hydrostatic water pressure
P	m/s	precipitation
P_k	m ² /s ³	production term in transport equation for turbulent kinetic energy
P_ξ	kg/(m ² s ²)	gradient hydrostatic pressure in ξ -direction
P_η	kg/(m ² s ²)	gradient hydrostatic pressure in η -direction
P_ε	m ² /s ⁴	production term in transport equation for the dissipation of turbulent kinetic energy
Q	m/s	global source or sink per unit area
Q_a	J/(m ² s)	atmospheric radiation (long wave radiation)
Q_{an}	J/(m ² s)	net atmospheric radiation
Q_{br}	J/(m ² s)	back radiation (long wave radiation)
Q_{co}	J/(m ² s)	heat loss due to convection
Q_{ev}	J/(m ² s)	heat loss due to evaporation
Q_s	J/(m ² s)	solar radiation (short wave radiation)
Q_{sc}	J/(m ² s)	solar radiation with a clear sky
Q_{sn}	J/(m ² s)	net solar insolation
Q_{tot}	J/(m ² s)	heat flux through free surface
q_{in}	1/s	local source per unit volume
q_{out}	1/s	local sink per unit volume
R	m	radius of the Earth
Ri	-	gradient Richardson's number
R_s	m	radius of curvature of a streamline in the secondary flow model
r	-	reflection coefficient for solar and atmospheric (long wave) radiation
r_{hum}	%	relative air humidity at a given temperature
S	ppt	salinity
S_{area}	m ²	representative surface area in heat model
S_{max}	ppt	maximum salinity for flocculation calculations
$S_{b,n}$		additional bedload transport vector. The direction of this vector is normal to the unadjusted bedload transport vector, in the down slope direction
$ S'_b $		magnitude of the unadjusted bedload transport vector (adjusted for longitudinal bed slope only)
S''_b	kg/(ms)	mass bedload transport rate
$ S''_b $	kg/(ms)	magnitude of the bedload transport vector
$S_{b,uu}^{(m,n)}$	kg/(ms)	computed bedload sediment transport vector in u -direction, held at the u -point of the computational cell at location (m, n)
s	-	relative density of sediment fraction ($= \rho_s / \rho_w$)
S	ppt	salinity
T	°C	temperature (general reference)
\bar{T}	K	temperature (general reference)
T	-	non-dimensional bed-shear stress

Symbol	Units	Meaning
T_a	-	non-dimensional bed-shear stress for reference concentration
T_a	°C	air temperature
T_{back}	°C	natural background temperature of water in the model area
T_p	s	peak wave period
$T_{TH_{bottom}}$	s	Thatcher Harleman return time for the bed layer
$T_{TH_{surface}}$	s	Thatcher Harleman return time for the surface layer
T_s	°C	water temperature at free surface
$T_{\xi\xi}, T_{\eta\eta}, T_{\xi\eta}$	kg/(ms ²)	contributions secondary flow to shear stress tensor
t	s	time
U	m/s	depth-averaged velocity in ξ -direction
\hat{U}	m/s	velocity of water discharged in ξ -direction
\hat{U}_δ	m/s	peak orbital velocity at the bed
u	m/s	flow velocity in the x - or ξ -direction
$u_{b,cr}$	m/s	critical (threshold) near-bed fluid velocity
u_b	m/s	near-bed fluid velocity vector
u_*	m/s	friction velocity due to currents or due to current and waves
u'_*	m/s	effective bed shear velocity
\bar{u}_*	m/s	vertically averaged friction velocity
u_{*b}	m/s	friction velocity at the bed
u_*^b	m/s	modified friction velocity near bed
u_*^s	m/s	friction velocity at the free surface
$ \vec{U} $	m/s	magnitude of depth-averaged horizontal velocity vector $(U, V)^T$
U_{10}	m/s	averaged wind speed at 10 m above free surface
\hat{U}	m/s	velocity of water discharged in x - or ξ -direction
\hat{U}_{orb}	m/s	amplitude of the near-bottom wave orbital velocity
u^S	m/s	Stokes drift in x - or ξ -direction
\tilde{u}	m/s	total velocity due to flow and Stokes drift in x - or ξ -direction
v	m/s	fluid velocity in the y - or η -direction
v_b	m/s	velocity at bed boundary layer in y - or η -direction
V	m/s	depth-averaged velocity in y - or η -direction
\hat{V}	m/s	velocity of water discharged in y - or η -direction
v^S	m/s	Stokes drift in y - or η -direction
\tilde{v}	m/s	total velocity due to flow and Stokes drift in y - or η -direction
w	m/s	fluid velocity in z -direction
$w_{s,0}$	m/s	particle settling velocity in clear water (non-hindered)
w_s	m/s	particle (hindered) settling velocity in a mixture
x, y, z	m	Cartesian co-ordinates
z_0	m	bed roughness length
\tilde{z}_0	m	enhanced bed roughness length due to current and waves
z_{0n}	m	roughness length normal to the boundary
$\frac{\partial z_b}{\partial s}$	-	bed slope in the direction of bedload transport
$\frac{\partial z_{(u)}}{\partial x}$	-	bed slope in the positive x -direction evaluated at the downwind u -point
$\frac{\partial z_{(v)}}{\partial y}$	-	bed slope in the positive y -direction evaluated at the downwind v -point
$\frac{\partial z_b}{\partial n}$	-	bed slope in the direction normal to the bedload transport vector

Symbol	Units	Meaning
α_{bn}	-	user-defined coefficient
β	-	ratio of sediment and fluid mixing
β_c	-	coefficient to account for secondary flow in momentum equations
δ_m	m	thickness of wave boundary mixing layer
δ_w	m	thickness of wave boundary layer
χ	deg	astronomical argument of a tidal component
Δ_r	m	ripple height
$\Delta_{SED}^{(m,n)}$	kg/m ²	change in quantity of bottom sediment at location (m, n)
Δt	s	computational time-step
$\Delta x^{(m,n)}$	m	cell width in the x -direction, held at the v -point of cell (m, n)
$\Delta y^{(m,n)}$	m	cell width in the y -direction, held at the u -point of cell (m, n)
Δz_b	m	thickness of the bed layer
Δz_s	m	thickness of the surface layer
$\Delta \sigma_b$	-	thickness of the bed boundary layer in relative co-ordinates
ε_ϵ	m ² /s ⁴	dissipation in transport equation for dissipation of turbulent kinetic energy
ε	m ² /s ³	dissipation in transport equation for turbulent kinetic energy
ε	-	emissivity coefficient of water at air-water interface
ε_f	m ² /s	fluid diffusion in the z -direction
$\varepsilon_{f,x}, \varepsilon_{f,y}, \varepsilon_{f,z}$	m ² /s	fluid diffusion coefficients in the x, y, z -directions, respectively
ε_s	m ² /s	sediment diffusion in the z -direction
$\varepsilon_{s,x}, \varepsilon_{s,y}, \varepsilon_{s,z}$	m ² /s	sediment diffusion coefficients in the x, y, z -directions, respectively
γ	m ⁻¹	extinction coefficient (measured)
γ	mbar/°C	Bowen's constant
ϕ	deg	latitude co-ordinate in spherical co-ordinates
ϕ	J/(sm ² °C)	exchange coefficient for the heat flux in excess temperature model
ϕ	deg	internal angle of friction of bed material (assumed to be 30°)
φ	m	equilibrium tide
φ^e	m	perturbation of the equilibrium tide due to earth tide
φ^{eo}	m	perturbation of the equilibrium tide due to tidal load
φ^o	m	perturbation of the equilibrium tide due to oceanic tidal load
λ	deg	longitude co-ordinate in spherical co-ordinates
λ_d	1/s	first order decay coefficient
ν_{mol}	m ² /s	kinematic viscosity (molecular) coefficient
ν_H^{back}	m ² /s	background horizontal eddy viscosity (ξ - and η -direction)
ν_V^{back}	m ² /s	background vertical eddy viscosity for momentum equations
D_H^{back}	m ² /s	background horizontal eddy diffusivity (ξ - and η -direction)
D_V^{back}	m ² /s	background vertical eddy diffusivity for transport equation
ν_V	m ² /s	vertical eddy viscosity
ν_{2D}	m ² /s	part of eddy viscosity due to horizontal turbulence model
ν_{3D}	m ² /s	part of eddy viscosity due to turbulence model in vertical direction
η	-	relative availability of the sediment fraction in the mixing layer
μ_c	-	efficiency factor current
ρ	kg/m ³	density of water

Symbol	Units	Meaning
ρ_a	kg/m ³	density of air
ρ_0	kg/m ³	reference density of water
σ	J/(m ² s K ⁴)	Stefan-Boltzmann's constant
σ	—	scaled vertical co-ordinate; $\sigma = \frac{z-\zeta}{d+\zeta}$; (surface, $\sigma = 0$; bed level, $\sigma = -1$)
σ_c	—	Prandtl-Schmidt number
σ_{c_0}	—	Prandtl-Schmidt number for constituent (0.7 for salinity and temperature, 1.0 for suspended sediments)
σ_{mol}	—	Prandtl-Schmidt number for molecular mixing (700 for salinity, 6.7 for temperature)
τ_b	N/m ²	bed shear stress due to current and waves
$\tau_{b\xi}$	kg/(ms ²)	bed shear stress in ξ -direction
$\tau_{b\eta}$	kg/(ms ²)	bed shear stress in η -direction
τ_c	kg/(ms ²)	magnitude of the bed shear stress due to current alone
τ_w	kg/(ms ²)	magnitude of the at bed shear stress due to waves alone
τ_m	kg/(ms ²)	magnitude of the wave-averaged at bed shear stress for combined waves and current
$\tau_{b,c}$	N/m ²	bed shear stress due to current
$\tau_{b,cr}$	N/m ²	critical bed shear stress
$\tau_{b,cw}$	N/m ²	bed shear stress due to current in the presence of waves
$\tau_{b,w}$	N/m ²	bed shear stress due to waves
τ_{cr}	N/m ²	critical bed shear stress
$\tau_{cr,d}$	N/m ²	user-defined critical deposition shear stress
$\tau_{cr,e}$	N/m ²	user-defined critical erosion shear stress
τ_{cw}	N/m ²	mean bed shear stress due to current and waves
τ_{max}	N/m ²	maximum bottom shear stress with wave-current interaction
τ_{mean}	N/m ²	mean (cycle averaged) bottom shear stress with wave-current interaction
$\tau_{s\xi}$	kg/(ms ²)	shear stress at surface in ξ -direction
$\tau_{s\eta}$	kg/(ms ²)	shear stress at surface in η -direction
ω	m/s	velocity in the σ -direction in the σ -co-ordinate system
ω	1/s	angular frequency waves
ω	deg/hour	angular frequency of tide and/or Fourier components
ψ	J/(m ² s)	heat flux through free surface
ξ, η	-	horizontal, curvilinear co-ordinates
ζ	m	water level above some horizontal plane of reference (datum)
ζ^b	m	bottom tide
ζ^e	m	earth tide
ζ^{eo}	m	tidal loading

9.3.1 Hydrodynamic equations

Delft3D-FLOW solves the Navier Stokes equations for an incompressible fluid, under the shallow water and the Boussinesq assumptions. In the vertical momentum equation the vertical accelerations are neglected, which leads to the hydrostatic pressure equation. In 3D models the vertical velocities are computed from the continuity equation. The set of partial differential equations in combination with an appropriate set of initial and boundary conditions is solved on a finite difference grid, see [chapter 10](#) for the numerical aspects.

In the horizontal direction Delft3D-FLOW uses orthogonal curvilinear co-ordinates. Two co-ordinate systems are supported:

- ◊ Cartesian co-ordinates (ξ, η)
- ◊ Spherical co-ordinates (λ, ϕ)

The boundaries of a river, an estuary or a coastal sea are in general curved and are not smoothly represented on a rectangular grid. The boundary becomes irregular and may introduce significant discretization errors. To reduce these errors boundary fitted orthogonal curvilinear co-ordinates are used. Curvilinear co-ordinates also allow local grid refinement in areas with large horizontal gradients.

Spherical co-ordinates are a special case of orthogonal curvilinear co-ordinates with:

$$\begin{aligned}\xi &= \lambda, \\ \eta &= \phi, \\ \sqrt{G_{\xi\xi}} &= R \cos \phi, \\ \sqrt{G_{\eta\eta}} &= R,\end{aligned}\tag{9.1}$$

in which λ is the longitude, ϕ is the latitude and R is the radius of the Earth (6 378.137 km, WGS84).

In Delft3D-FLOW the equations are formulated in orthogonal curvilinear co-ordinates. The velocity scale is in physical space, but the components are perpendicular to the cell faces of the curvilinear grid. The grid transformation introduces curvature terms in the equations of motion.

In the vertical direction Delft3D-FLOW offers two different vertical grid systems: the σ co-ordinate system (σ -model) and the Cartesian Z co-ordinate system (Z -model). The hydrodynamic equations described in this section are valid for the σ co-ordinate system. The equations for the Z co-ordinate system are similar. Equations or relations that are fundamentally different will be discussed separately and in detail in [chapter 12](#).

The σ co-ordinate system

The σ -grid was introduced by [Phillips \(1957\)](#) for atmospheric models. The vertical grid consists of layers bounded by two σ -planes, which are not strictly horizontal but follow the bottom topography and the free surface. Because the σ -grid is boundary fitted both to the bottom and to the moving free surface, a smooth representation of the topography is obtained.

The number of layers over the entire horizontal computational area is constant, irrespective of the local water depth (see [Figure 9.2](#)). The distribution of the relative layer thickness is usually non-uniform. This allows for more resolution in the zones of interest such as the near surface area (important for e.g. wind-driven flows, heat exchange with the atmosphere) and the near bed area (sediment transport).

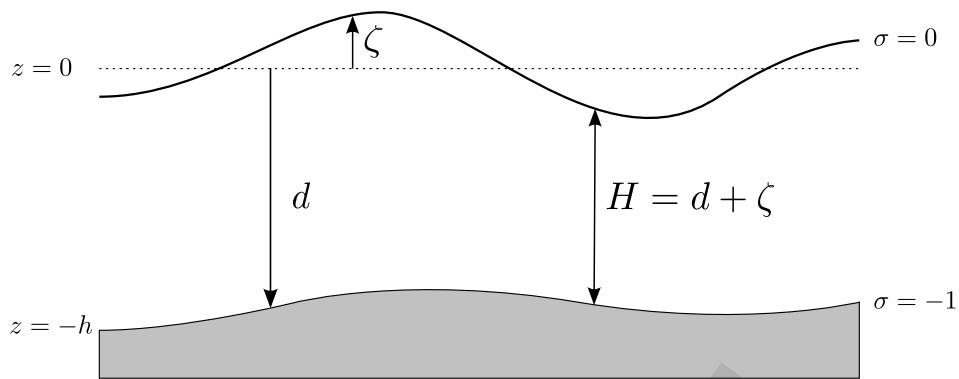


Figure 9.1: Definition of water level (ζ), depth (d) and total depth (H).

The σ co-ordinate system is defined as:

$$\sigma = \frac{z - \zeta}{d + \zeta} = \frac{z - \zeta}{H}, \quad (9.2)$$

with:

z	the vertical co-ordinate in physical space.
ζ	the free surface elevation above the reference plane (at $z = 0$).
d	the depth below the reference plane.
H	the total water depth, given by $d + \zeta$.

At the bottom $\sigma = -1$ and at the free surface $\sigma = 0$ (see Figure 9.1). The partial derivatives in the original Cartesian co-ordinate system are expressed in σ co-ordinates by the chain rule introducing additional terms (Stelling and Van Kester, 1994). The flow domain of a 3D shallow water model consists in the horizontal plane of a restricted (limited) area composed of open and closed (land) boundaries and in the vertical of a number of layers. In a σ co-ordinate system the number of layers is the same at every location in the horizontal plane, i.e. the layer interfaces are chosen following planes of constant σ , see Figure 9.2. For each layer a set of coupled conservation equations is solved.

Cartesian co-ordinate system in the vertical (Z -model)

In coastal seas, estuaries and lakes, stratified flow can occur in combination with steep topography. Although the σ -grid is boundary fitted (in the vertical), it will not always have enough resolution around the pycnocline. The co-ordinate lines intersect the density interfaces that may give significant errors in the approximation of strictly horizontal density gradients (Leendertse, 1990; Stelling and Van Kester, 1994). Therefore, recently a second vertical grid co-ordinate system based on Cartesian co-ordinates (Z -grid) was introduced in Delft3D-FLOW for 3D simulations of weakly forced stratified water systems.

The Z -grid model has horizontal co-ordinate lines that are (nearly) parallel with density interfaces (isopycnals) in regions with steep bottom slopes. This is important to reduce artificial mixing of scalar properties such as salinity and temperature. The Z -grid is not boundary-fitted in the vertical. The bottom (and free surface) is usually not a co-ordinate line and is represented as a staircase (zig-zag boundary; see Figure 9.2).

Continuity equation

The depth-averaged continuity equation is derived by integration the continuity equation for incompressible fluids ($\nabla \cdot \vec{u} = 0$) over the total depth, taken into account the kinematic

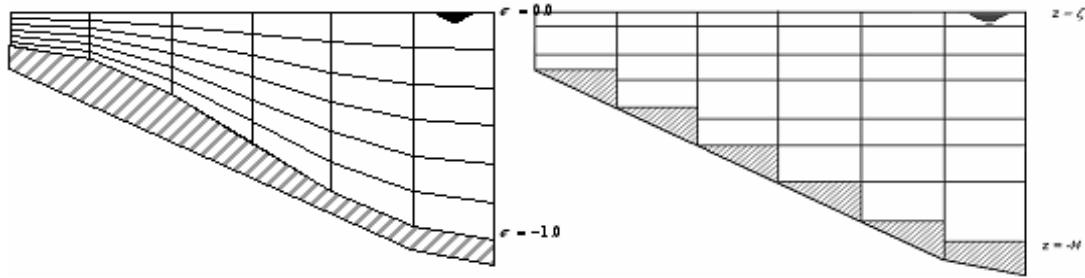


Figure 9.2: Example of σ - and Z -grid

boundary conditions at water surface and bed level, and is given by:

$$\frac{\partial \zeta}{\partial t} + \frac{1}{\sqrt{G_{\xi\xi}} \sqrt{G_{\eta\eta}}} \frac{\partial ((d + \zeta) U \sqrt{G_{\eta\eta}})}{\partial \xi} + \frac{1}{\sqrt{G_{\xi\xi}} \sqrt{G_{\eta\eta}}} \frac{\partial ((d + \zeta) V \sqrt{G_{\xi\xi}})}{\partial \eta} = (d + \zeta) Q, \quad (9.3)$$

with U and V the depth averaged velocities

$$U = \frac{1}{d + \zeta} \int_d^\zeta u dz = \int_{-1}^0 u d\sigma \quad (9.4)$$

$$V = \frac{1}{d + \zeta} \int_d^\zeta v dz = \int_{-1}^0 v d\sigma \quad (9.5)$$

and Q representing the contributions per unit area due to the discharge or withdrawal of water, precipitation and evaporation:

$$Q = \int_{-1}^0 (q_{in} - q_{out}) d\sigma + P - E, \quad (9.6)$$

with q_{in} and q_{out} the local sources and sinks of water per unit of volume [1/s], respectively, P the non-local source term of precipitation and E non-local sink term due to evaporation. We remark that the intake of, for example, a power plant is a withdrawal of water and should be modelled as a sink. At the free surface there may be a source due to precipitation or a sink due to evaporation.

Momentum equations in horizontal direction

The momentum equations in ξ - and η -direction are given by:

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{u}{\sqrt{G_{\xi\xi}}} \frac{\partial u}{\partial \xi} + \frac{v}{\sqrt{G_{\eta\eta}}} \frac{\partial u}{\partial \eta} + \frac{\omega}{d + \zeta} \frac{\partial u}{\partial \sigma} - \frac{v^2}{\sqrt{G_{\xi\xi}} \sqrt{G_{\eta\eta}}} \frac{\partial \sqrt{G_{\eta\eta}}}{\partial \xi} + \\ + \frac{uv}{\sqrt{G_{\xi\xi}} \sqrt{G_{\eta\eta}}} \frac{\partial \sqrt{G_{\xi\xi}}}{\partial \eta} - fv = -\frac{1}{\rho_0 \sqrt{G_{\xi\xi}}} P_\xi + F_\xi + \\ + \frac{1}{(d + \zeta)^2} \frac{\partial}{\partial \sigma} \left(\nu_V \frac{\partial u}{\partial \sigma} \right) + M_\xi, \end{aligned} \quad (9.7)$$

and

$$\begin{aligned} \frac{\partial v}{\partial t} + \frac{u}{\sqrt{G_{\xi\xi}}}\frac{\partial v}{\partial \xi} + \frac{v}{\sqrt{G_{\eta\eta}}}\frac{\partial v}{\partial \eta} + \frac{\omega}{d+\zeta}\frac{\partial v}{\partial \sigma} + \frac{uv}{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}}\frac{\partial \sqrt{G_{\eta\eta}}}{\partial \xi} + \\ - \frac{u^2}{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}}\frac{\partial \sqrt{G_{\xi\xi}}}{\partial \eta} + fu = -\frac{1}{\rho_0\sqrt{G_{\eta\eta}}}P_\eta + F_\eta + \\ + \frac{1}{(d+\zeta)^2}\frac{\partial}{\partial \sigma}\left(\nu_V\frac{\partial v}{\partial \sigma}\right) + M_\eta. \quad (9.8) \end{aligned}$$

The vertical eddy viscosity coefficient ν_V is defined in [Equation 9.20](#). Density variations are neglected, except in the baroclinic pressure terms, P_ξ and P_η represent the pressure gradients. The forces F_ξ and F_η in the momentum equations represent the unbalance of horizontal Reynold's stresses. M_ξ and M_η represent the contributions due to external sources or sinks of momentum (external forces by hydraulic structures, discharge or withdrawal of water, wave stresses, etc.). The effects of surface waves on the flow as modelled in Delft3D-FLOW are described in [section 9.7](#).

Vertical velocities

The vertical velocity ω in the adapting σ -co-ordinate system is computed from the continuity equation:

$$\begin{aligned} \frac{\partial \zeta}{\partial t} + \frac{1}{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}}\frac{\partial ((d+\zeta)u\sqrt{G_{\eta\eta}})}{\partial \xi} + \frac{1}{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}}\frac{\partial ((d+\zeta)v\sqrt{G_{\xi\xi}})}{\partial \eta} + \\ + \frac{\partial \omega}{\partial \sigma} = (d+\zeta)(q_{in} - q_{out}). \quad (9.9) \end{aligned}$$

At the surface the effect of precipitation and evaporation is taken into account. The vertical velocity ω is defined at the iso σ -surfaces. ω is the vertical velocity relative to the moving σ -plane. It may be interpreted as the velocity associated with up- or downwelling motions. The "physical" vertical velocities w in the Cartesian co-ordinate system are not involved in the model equations. Computation of the physical vertical velocities is only required for post-processing purposes. These velocities can be expressed in the horizontal velocities, water depths, water levels and vertical ω -velocity according to:

$$\begin{aligned} w = \omega + \frac{1}{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}}\left[u\sqrt{G_{\eta\eta}}\left(\sigma\frac{\partial H}{\partial \xi} + \frac{\partial \zeta}{\partial \xi}\right) + v\sqrt{G_{\xi\xi}}\left(\sigma\frac{\partial H}{\partial \eta} + \frac{\partial \zeta}{\partial \eta}\right)\right] + \\ + \left(\sigma\frac{\partial H}{\partial t} + \frac{\partial \zeta}{\partial t}\right). \quad (9.10) \end{aligned}$$

Hydrostatic pressure assumption (for σ -grid)

Under the shallow water assumption, the vertical momentum equation is reduced to a hydrostatic pressure equation. Vertical accelerations due to buoyancy effects and due to sudden variations in the bottom topography are not taken into account. So:

$$\frac{\partial P}{\partial \sigma} = -g\rho H. \quad (9.11)$$

After integration, the hydrostatic pressure is given by:

$$P = P_{atm} + gH \int_{\sigma}^0 \rho(\xi, \eta, \sigma', t) d\sigma'. \quad (9.12)$$

Firstly, we consider water with a constant density and next with a non-uniform density.

For water of constant density and taking into account the atmospheric pressure, the pressure gradients read:

$$\frac{1}{\rho_0 \sqrt{G_{\xi\xi}}} P_{\xi} = \frac{g}{\sqrt{G_{\xi\xi}}} \frac{\partial \zeta}{\partial \xi} + \frac{1}{\rho_0 \sqrt{G_{\xi\xi}}} \frac{\partial P_{atm}}{\partial \xi}, \quad (9.13)$$

$$\frac{1}{\rho_0 \sqrt{G_{\eta\eta}}} P_{\eta} = \frac{g}{\sqrt{G_{\eta\eta}}} \frac{\partial \zeta}{\partial \eta} + \frac{1}{\rho_0 \sqrt{G_{\eta\eta}}} \frac{\partial P_{atm}}{\partial \eta}. \quad (9.14)$$

The gradients of the free surface level are the so-called barotropic pressure gradients. The atmospheric pressure is included in the system for storm surge simulations. The atmospheric pressure gradients dominate the external forcing at peak winds during storm events. Space and time varying wind and pressure fields are especially important when simulating storm surges.

In case of a non-uniform density, the local density is related to the values of temperature and salinity by the equation of state, see [subsection 9.3.4](#). Leibniz rule is used to obtain the following expressions for the horizontal pressure gradients:

$$\frac{1}{\rho_0 \sqrt{G_{\xi\xi}}} P_{\xi} = \frac{g}{\sqrt{G_{\xi\xi}}} \frac{\partial \zeta}{\partial \xi} + g \frac{d + \zeta}{\rho_0 \sqrt{G_{\xi\xi}}} \int_{\sigma}^0 \left(\frac{\partial \rho}{\partial \xi} + \frac{\partial \rho}{\partial \sigma} \frac{\partial \sigma}{\partial \xi} \right) d\sigma' \quad (9.15)$$

$$\frac{1}{\rho_0 \sqrt{G_{\eta\eta}}} P_{\eta} = \frac{g}{\sqrt{G_{\eta\eta}}} \frac{\partial \zeta}{\partial \eta} + g \frac{d + \zeta}{\rho_0 \sqrt{G_{\eta\eta}}} \int_{\sigma}^0 \left(\frac{\partial \rho}{\partial \eta} + \frac{\partial \rho}{\partial \sigma} \frac{\partial \sigma}{\partial \eta} \right) d\sigma' \quad (9.16)$$

The first term in Eqs. 9.15 and 9.16 represents the barotropic pressure gradient (without atmospheric pressure gradients) and the second term the baroclinic pressure gradient. In the horizontal gradient a vertical derivative is introduced by the σ co-ordinate transformation. In estuaries and coastal seas the vertical grid may deteriorate strongly in case of steep bottom slopes. In order to avoid artificial flow the numerical approximation of the baroclinic pressure terms requires a special numerical approach. The discretisations of Delft3D-FLOW are discussed in Section 10.10, see also [Stelling and Van Kester \(1994\)](#).

Floating structures

To simulate the influence of floating structures, such as ships or pontoons, both the pressure term in the momentum equations and the continuity equation are modified.

The pressure of a floating structure at the free surface is dependent on the draft of the structure, generically called 'ship' (d_{ship}):

$$P_{atm} = g\rho_0 d_{\text{ship}} \quad (9.17)$$

If the free surface is covered by a floating structure, the free surface cannot move freely. The water level should be fixed, using a “rigid lid” boundary condition. This boundary condition is implemented by reducing the storage area of the grid cell with a factor α (the parameter `Riglid` in subsection B.3.6):

$$\alpha \frac{\partial \zeta}{\partial t} \quad (9.18)$$

with $0 < \alpha \leq 1$. The wave speed below the floating structure increases to $\sqrt{gH/\alpha}$. To prevent numerical inaccuracy for wave propagation α *must* be larger than zero. We suggest $\alpha \approx 0.1$.

Coriolis force

The Coriolis parameter f depends on the geographic latitude and the angular speed of rotation of the earth, Ω : $f = 2\Omega \sin \phi$. For a curvilinear grid you should specify the space varying Coriolis parameter, using a suitable projection.

Reynold's stresses

The forces F_ξ and F_η in the horizontal momentum equations represent the unbalance of horizontal Reynolds stresses. The Reynolds stresses are modelled using the eddy viscosity concept, (for details e.g. [Rodi \(1984\)](#)). This concept expresses the Reynolds stress component as the product between a flow as well as grid-dependent eddy viscosity coefficient and the corresponding components of the mean rate-of-deformation tensor. The meaning and the order of the eddy viscosity coefficients differ for 2D and 3D, for different horizontal and vertical turbulence length scales and fine or coarse grids. In general the eddy viscosity is a function of space and time.

For 3D shallow water flow the stress tensor is an-isotropic. The horizontal eddy viscosity coefficient, ν_H , is much larger than the vertical eddy viscosity ν_V ($\nu_H \gg \nu_V$). The horizontal viscosity coefficient may be a superposition of three parts:

- 1 a part due to “sub-grid scale turbulence”,
- 2 a part due to “3D-turbulence” see [Uittenbogaard et al. \(1992\)](#) and
- 3 a part due to dispersion for depth-averaged simulations.

In simulations with the depth-averaged momentum and transport equations, the redistribution of momentum and matter due to the vertical variation of the horizontal velocity is denoted as dispersion. In 2D simulations this dispersion is not simulated as the vertical profile of the horizontal velocity is not resolved. Then this dispersive effect may be modelled as the product of a viscosity coefficient and a velocity gradient. The dispersion term may be estimated by the Elder formulation.

If the vertical profile of the horizontal velocity is not close to a logarithmic profile (e.g. due to stratification or due to forcing by wind) then a 3D-model for the transport of matter is recommended.

The horizontal eddy viscosity is mostly associated with the contribution of horizontal turbulent motions and forcing that are not resolved by the horizontal grid (“sub-grid scale turbulence”) or by (a priori) the Reynolds-averaged shallow-water equations. For the former we introduce the sub-grid scale (SGS) horizontal eddy viscosity ν_{SGS} and for the latter the horizontal eddy viscosity ν_H^{back} . Delft3D-FLOW simulates the larger scale horizontal turbulent motions through a methodology called Horizontal Large Eddy Simulation (HLES). The associated horizontal vis-

Table 9.2: Overview of eddy viscosity options in Delft3D-FLOW

model description	ν_{SGS}	ν_H^{back} (represents)	ν_{3D}	ν_V^{back}
2D, no HLES	-	2D-turbulence + dispersion coefficient	-	-
2D, with HLES	computed by HLES	3D-turbulence + dispersion coefficient	-	-
3D, no HLES	-	2D-turbulence	computed by vertical turbulence model.	background vertical viscosity
3D, with HLES	computed by HLES	-	computed by vertical turbulence model.	background vertical viscosity

cosity coefficient ν_{SGS} will then be computed by a dedicated SGS-turbulence model, including the Elder contribution if requested. For details of this approach, see [section B.8](#).

The background horizontal viscosity, user-defined through the input file is represented by ν_H^{back} . Consequently, in Delft3D-FLOW the horizontal eddy viscosity coefficient is defined by

$$\nu_H = \nu_{SGS} + \nu_V + \nu_H^{back}. \quad (9.19)$$

The 3D part ν_V is referred to as the three-dimensional turbulence and in 3D simulations it is computed following a 3D-turbulence closure model.

For turbulence closure models responding to shear production only, it may be convenient to specify a background or “ambient” vertical mixing coefficient in order to account for all other forms of unresolved mixing, ν_V^{back} . Therefore, in addition to all turbulence closure models in Delft3D-FLOW a constant (space and time) background mixing coefficient may be specified by you, which is a background value for the vertical eddy viscosity in the momentum Eqs. [9.7](#) and [9.8](#). Consequently, the vertical eddy viscosity coefficient is defined by:

$$\nu_V = \nu_{mol} + \max(\nu_{3D}, \nu_V^{back}), \quad (9.20)$$

with ν_{mol} the kinematic viscosity of water. The 3D part ν_{3D} is computed by a 3D-turbulence closure model, see [section 9.5](#). Summarizing, since in Delft3D-FLOW several combinations of horizontal and vertical eddy viscosity are optional, Table [9.2](#) presents an overview of these combinations.



Remarks:

- ◊ We note that the “background horizontal eddy viscosity” represents a series of complicated hydrodynamic phenomena. Table [9.2](#) shows that this background horizontal eddy viscosity ν_H^{back} either contains zero, one or two contributions.
- ◊ The background horizontal eddy viscosity ν_H^{back} has to be specified by you in the GUI in addition to the Elder formulation for 3D-turbulence and dispersion in 2D simulations.
- ◊ For the description of the (vertical) turbulence models for 3D simulations, we refer to [section 9.5](#).
- ◊ It is important to emphasize that the lower limit of vertical eddy viscosity ν_V^{back} is only used in the momentum equations (see Eqs. [9.7](#) and [9.8](#)) and is not used in the vertical turbulence models (see for example Eqs. [9.115](#), [9.127](#) and [9.128](#)). For the transport equation there is similar lower limit.

For the sake of completeness; to model secondary flow in river bends in depth-averaged simulations, Delft3D-FLOW contains a formulation to account for the effect of this spiral motion. Then, the horizontal stress tensor is extended with additional shear stresses, see [section 9.6](#).

The σ co-ordinate system rotates the Cartesian stress tensor and introduces additional derivatives ([Stelling and Van Kester, 1994](#)). In Delft3D-FLOW, the stress tensor is redefined in the σ co-ordinate system assuming that the horizontal length scale is much larger than the water depth ([Blumberg and Mellor, 1985](#)) and that the flow is of boundary-layer type. The horizontal gradients are taken along σ -planes. This approach guarantees a positive definite operator, also on the numerical grid ([Beckers et al., 1998](#)). Thus, the forces F_ξ and F_η are of the form:

$$F_\xi = \frac{1}{\sqrt{G_{\xi\xi}}} \frac{\partial \tau_{\xi\xi}}{\partial \xi} + \frac{1}{\sqrt{G_{\eta\eta}}} \frac{\partial \tau_{\xi\eta}}{\partial \eta}, \quad (9.21)$$

$$F_\eta = \frac{1}{\sqrt{G_{\xi\xi}}} \frac{\partial \tau_{\eta\xi}}{\partial \xi} + \frac{1}{\sqrt{G_{\eta\eta}}} \frac{\partial \tau_{\eta\eta}}{\partial \eta}. \quad (9.22)$$

For small-scale flow, i.e. when the shear stresses at the closed boundaries must be taken into account, the shear stresses $\tau_{\xi\xi}$, $\tau_{\xi\eta}$, $\tau_{\eta\xi}$ and $\tau_{\eta\eta}$ are determined according to:

$$\tau_{\xi\xi} = \frac{2\nu_H}{\sqrt{G_{\xi\xi}}} \left(\frac{\partial u}{\partial \xi} + \frac{\partial u}{\partial \sigma} \frac{\partial \sigma}{\partial \xi} \right), \quad (9.23)$$

$$\tau_{\xi\eta} = \tau_{\eta\xi} = \nu_H \left\{ \frac{1}{\sqrt{G_{\eta\eta}}} \left(\frac{\partial u}{\partial \eta} + \frac{\partial u}{\partial \sigma} \frac{\partial \sigma}{\partial \eta} \right) + \frac{1}{\sqrt{G_{\xi\xi}}} \left(\frac{\partial v}{\partial \xi} + \frac{\partial v}{\partial \sigma} \frac{\partial \sigma}{\partial \xi} \right) \right\}, \quad (9.24)$$

$$\tau_{\eta\eta} = \frac{2\nu_H}{\sqrt{G_{\eta\eta}}} \left(\frac{\partial v}{\partial \eta} + \frac{\partial v}{\partial \sigma} \frac{\partial \sigma}{\partial \eta} \right). \quad (9.25)$$

For large-scale flow simulated with coarse horizontal grids, i.e. when the shear stresses along the closed boundaries may be neglected, the forces F_ξ and F_η are simplified. The horizontal viscosity terms in Delft3D-FLOW are then reduced to the Laplace operator along grid lines:

$$F_\xi = \nu_H \left(\frac{1}{\sqrt{G_{\xi\xi}} \sqrt{G_{\xi\xi}}} \frac{\partial^2 u}{\partial \xi^2} + \frac{1}{\sqrt{G_{\eta\eta}} \sqrt{G_{\eta\eta}}} \frac{\partial^2 u}{\partial \eta^2} \right), \quad (9.26)$$

$$F_\eta = \nu_H \left(\frac{1}{\sqrt{G_{\xi\xi}} \sqrt{G_{\xi\xi}}} \frac{\partial^2 v}{\partial \xi^2} + \frac{1}{\sqrt{G_{\eta\eta}} \sqrt{G_{\eta\eta}}} \frac{\partial^2 v}{\partial \eta^2} \right), \quad (9.27)$$

where the eddy viscosity has been assumed to be a constant.

The choice between the formulations represented by Eqs. 9.21 to 9.25, or Eqs. 9.26 and 9.27 depends on the application (velocity scales) and can be specified by you. Eqs. 9.21 to 9.25 are used in combination with rough side walls (partial slip). Their numerical implementation is explicit and this introduces a stability condition, see [chapter 10](#).

In the expressions for the horizontal viscosity terms, the curvature of the grid in the horizontal and vertical plane has been neglected.

In strongly stratified flows, the turbulent eddy viscosity at the interface reduces to zero and the vertical mixing reduces to molecular diffusion (laminar flow). This is physically not realistic because in strongly stratified flow random internal waves exist. These internal waves are generated continuously by various sources; they increase the production of turbulence and generate vertical mixing of momentum. Because the effect of internal waves is not explicitly taken into account in the present implementation of the turbulence models, the effect of internal waves may be assigned to a constant (in space and time) background eddy viscosity at input, which is added to the momentum equations only.

Discharge or withdrawal of momentum

The discharge of water, when momentum is taken into account, gives an additional term in the U and V momentum equation:

$$M_\xi = q_{in}(\hat{U} - u) \quad (9.28)$$

$$M_\eta = q_{in}(\hat{V} - v) \quad (9.29)$$

with \hat{U} and \hat{V} the velocity components of the momentum discharged in ξ - and η -direction, respectively. If \hat{U} and \hat{V} are set to zero, then the flow is decelerated.

9.3.2 Transport equation (for σ -grid)

The flows in rivers, estuaries, and coastal seas often transport dissolved substances, salinity and/or heat. In Delft3D-FLOW, the transport of matter and heat is modelled by an advection-diffusion equation in three co-ordinate directions. Source and sink terms are included to simulate discharges and withdrawals. Also first-order decay processes may be taken into account. A first-order decay process corresponds to a numerical solution which is exponentially decreasing. For more complex processes e.g. eutrophication, biological and/or chemical processes the water quality module D-Water Quality should be used.

The transport equation here is formulated in a conservative form in orthogonal curvilinear co-ordinates in the horizontal direction and σ co-ordinates in the vertical direction:

$$\begin{aligned} \frac{\partial(d + \zeta)c}{\partial t} + \frac{1}{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}} &\left\{ \frac{\partial[\sqrt{G_{\eta\eta}}(d + \zeta)uc]}{\partial\xi} + \frac{\partial[\sqrt{G_{\xi\xi}}(d + \zeta)vc]}{\partial\eta} \right\} + \frac{\partial\omega c}{\partial\sigma} = \\ \frac{d + \zeta}{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}} &\left\{ \frac{\partial}{\partial\xi} \left(D_H \frac{\sqrt{G_{\eta\eta}}}{\sqrt{G_{\xi\xi}}} \frac{\partial c}{\partial\xi} \right) + \frac{\partial}{\partial\eta} \left(D_H \frac{\sqrt{G_{\xi\xi}}}{\sqrt{G_{\eta\eta}}} \frac{\partial c}{\partial\eta} \right) \right\} + \\ &+ \frac{1}{d + \zeta} \frac{\partial}{\partial\sigma} \left(D_V \frac{\partial c}{\partial\sigma} \right) - \lambda_d(d + \zeta)c + S, \end{aligned} \quad (9.30)$$

with D_H the horizontal diffusion coefficient, D_V the vertical diffusion coefficient, λ_d representing the first order decay process and S the source and sink terms per unit area due to the discharge q_{in} or withdrawal q_{out} of water and/or the exchange of heat through the free surface Q_{tot} :

$$S = (d + \zeta)(q_{in}c_{in} - q_{out}c) + Q_{tot}. \quad (9.31)$$

The total horizontal diffusion coefficient D_H is defined by

$$D_H = D_{SGS} + D_V + D_H^{back} \quad (9.32)$$

with D_{SGS} the diffusion due to the sub-grid scale turbulence model (Equation B.4). We remark that in Equation 9.32 a user defined diffusion coefficient for the horizontal is used D_H^{back} . The reason is that you may use this horizontal diffusion coefficient for calibration independent of the horizontal eddy viscosity. This eddy diffusivity depends on the constituent.

The vertical diffusion coefficient D_V is defined by

$$D_V = \frac{\nu_{mol}}{\sigma_{mol}} + \max(D_{3D}, D_V^{back}), \quad (9.33)$$

with D_{3D} the diffusion due to turbulence model in vertical direction (Equation 9.103) and with ν_{mol} the kinematic viscosity of water and σ_{mol} is either the (molecular) Prandtl number for heat diffusion or the Schmidt number for diffusion of dissolved matter.

For vertical mixing, the eddy viscosity ν_V is defined by you at input via the GUI (e.g. in case of a constant vertical eddy viscosity) or computed by a 3D-turbulence model, see Equation 9.20. Delft3D-FLOW will also compute the vertical diffusion coefficients by taking into account the turbulence Prandtl-Schmidt number.

For shallow-water flow, the diffusion tensor is an-isotropic. Typically, the horizontal eddy diffusivity D_H exceeds the vertical eddy diffusivity D_V . The horizontal diffusion coefficient is assumed to be a superposition of three parts:

- 1 The 2D part D_{SGS} (*sub-grid scale turbulence*), due to motions and mixing that are not resolved by the horizontal grid.
- 2 The contribution due to 3D turbulence, D_{3D} , which is related to the turbulent eddy viscosity and is computed by a 3D turbulence closure model, see Uittenbogaard *et al.* (1992).
- 3 D_H^{back} representing the Reynolds-averaged equations and/or accounting for other unresolved horizontal mixing.

For depth-averaged simulations, the horizontal eddy diffusivity should also include dispersion by the vertical variation of the horizontal flow (so-called Taylor-shear dispersion). Delft3D-FLOW has an option (so-called HLES approach) solving explicitly the larger scale horizontal turbulent motions. Then the 2D part D_{SGS} is associated with mixing by horizontal motions and forcing that cannot be resolved on the grid. For details of the HLES-approach, see section B.8. The horizontal background diffusion coefficient is represented by D_H^{back} and it is specified by you at input and it should account for all other forms of unresolved mixing (in most cases it is a calibration parameter).

Remark:

- ◊ We remark that for small scale applications the default value set by the GUI of 10 m²/s is too large.



In addition to the vertical eddy diffusivity estimated by a 3D turbulence model you may specify a background or “ambient” vertical mixing coefficient accounting for all other forms of unresolved mixing. In strongly stratified flows, breaking internal waves may be generated which are not modelled by the available 3D turbulence models of Delft3D-FLOW. Instead, you can specify the so-called Ozmidov length scale L_{oz} , see section 9.5 for a more detailed description. Then, the 3D-turbulence part D_{3D} is defined as the maximum of the vertical eddy diffusivity computed by the turbulence model and the Ozmidov length scale (see also Equation 9.101):

$$D_{3D} = \max \left(D_{3D}, 0.2L_{oz}^2 \sqrt{-\frac{g}{\rho} \frac{\partial \rho}{\partial z}} \right). \quad (9.34)$$

Table 9.3: Overview of eddy diffusivity options in Delft3D-FLOW

model description	D_{SGS}	D_H^{back} (represents)	D_{3D}	D_V^{back}
2D, no HLES	0	2D-turbulence + dispersion coefficient	-	-
2D, with HLES	computed by HLES	3D-turbulence + dispersion coefficient	-	-
3D, no HLES	0	2D-turbulence	maximum of value of turbulence model and the Ozmidov length scale.	background eddy diffusivity
3D, with HLES	computed by HLES	-	maximum of value of turbulence model and the Ozmidov length scale.	background eddy diffusivity

For all turbulence closure models in Delft3D-FLOW a vertical background mixing coefficient D_V^{back} has to be specified by you via the GUI, [Equation 9.33](#)

For two-dimensional depth-averaged simulations, the horizontal eddy diffusivity D_H^{back} should also contain a contribution due to the vertical variation of the horizontal flow (Taylor shear dispersion). This part is discarded by depth averaging the 3D advection diffusion equation. For the application of a depth-averaged transport model, the water column should be well-mixed and the Taylor shear dispersion should be of secondary importance, otherwise a 3D model should be used for the simulation of the transport of matter.

Summarizing, in Delft3D-FLOW many options are available for the eddy diffusivity; [Table 9.3](#) presents an overview of these options. We note that this is rather similar to the eddy viscosity options (see [Table 9.2](#)).



Remarks:

- ◊ The horizontal background diffusion-type coefficient is represented by D_H^{back} and prescribed by you in the GUI. The default value of 10 m²/s is too large for small scale applications.
- ◊ The vertical eddy diffusivity D_{3D} is the maximum value of the Ozmidov length scale and the vertical eddy viscosity computed by the turbulence model. If you have not switched on the Ozmidov length scale, which is the case in most of the simulations, then the Ozmidov length scale is equal to zero.

Artificial vertical diffusion

The σ co-ordinate system rotates the Cartesian stress tensor and introduces additional derivatives ([Stelling and Van Kester, 1994](#)). In Delft3D-FLOW the diffusion tensor is redefined in the σ co-ordinate system assuming that the horizontal length scale is much larger than the water depth ([Blumberg and Mellor, 1985](#)) and that the flow is of boundary-layer type. The horizontal gradients are taken along σ -planes. This approach guarantees a positive definite operator, also on the numerical grid ([Beckers et al., 1998](#)).

For steep bottom slopes in combination with vertical stratification, the horizontal diffusion along σ -planes introduces artificial vertical diffusion ([Huang and Spaulding, 1996](#)). Strict horizontal diffusion along Z -planes gives a more realistic description of the physical transport process. Numerical implementation of the complete transformation behaved poorly ([Blumberg and Mellor, 1985](#)). [Stelling and Van Kester \(1994\)](#) developed an algorithm to approximate the horizontal diffusion along Z -planes in a σ -co-ordinate framework, see Section [10.10](#).

9.3.3 Coupling between intake and outfall

For a power plant, the discharge at the outfall is set equal to the inflow at the intake, i.e.:

$$q_{out} = q_{in}. \quad (9.35)$$

The temperature, salinity and concentrations of substances prescribed as a function of time for the Intake - Outfall combination are added to the values at the intake, i.e. they represent the influence of the power plant on their value or concentration:

$$T_{out} = T_{in} + \Delta T_{\text{discharge}}, \quad (9.36)$$

$$S_{out} = S_{in} + \Delta S_{\text{discharge}}, \quad (9.37)$$

$$C_{out} = C_{in} + \Delta C_{\text{discharge}}, \quad (9.38)$$

where the increase in temperature, salinity or concentration due to the power plant can be negative.

Remark:

- ◊ The temperature, salinity, or concentration at the outfall is set to zero when its value would become negative. The computation is continued, but a warning is written to the diagnostic file.



9.3.4 Equation of state

The density of water ρ is a function of salinity (s) and temperature (t). In Delft3D-FLOW you can choose between two different formulations for the equation of state (Eckart or UNESCO).

The first empirical relation derived by [Eckart \(1958\)](#) is based on a limited number of measurements dating from 1910 (only two salinities at 5 temperatures). In the original equation the pressure is present, but at low pressures the effect on density can be neglected.

Eckart formulation

The Eckart formulation is given by ([Eckart, 1958](#)):

Range: $0 < t < 40^\circ\text{C}$, $0 < s < 40 \text{ ppt}$

$$\rho = \frac{1,000 P_0}{\lambda + \alpha_0 P_0}, \quad (9.39)$$

where:

$$\lambda = 1779.5 + 11.25t - 0.0745t^2 - (3.80 + 0.01t) s, \quad (9.40)$$

$$\alpha_0 = 0.6980, \quad (9.41)$$

$$P_0 = 5890 + 38t - 0.375t^2 + 3s. \quad (9.42)$$

with the salinity s in ppt and the water temperature t in $^{\circ}\text{C}$.

For fresh water the Eckart formulation does not give the maximum density at the temperature of 4 degrees Celsius. This is an important physical aspect when simulating the thermocline formation in lakes or stagnant basins. Therefore also the so-called UNESCO formulation is implemented in Delft3D-FLOW.

UNESCO formulation

The UNESCO formulation is given by ([UNESCO, 1981a](#)):

Range: $0 < t < 40 \ ^{\circ}\text{C}$, $0.5 < s < 43$ ppt

$$\rho = \rho_0 + As + Bs^{3/2} + Cs^2 \quad (9.43)$$

where

$$\begin{aligned} \rho_0 = 999.842594 + 6.793952 \cdot 10^{-2}t - 9.095290 \cdot 10^{-3}t^2 + \\ + 1.001685 \cdot 10^{-4}t^3 - 1.120083 \cdot 10^{-6}t^4 + 6.536332 \cdot 10^{-9}t^5 \end{aligned} \quad (9.44)$$

$$\begin{aligned} A = 8.24493 \cdot 10^{-1} - 4.0899 \cdot 10^{-3}t + 7.6438 \cdot 10^{-5}t^2 + \\ - 8.2467 \cdot 10^{-7}t^3 + 5.3875 \cdot 10^{-9}t^4 \end{aligned} \quad (9.45)$$

$$B = -5.72466 \cdot 10^{-3} + 1.0227 \cdot 10^{-4}t - 1.6546 \cdot 10^{-6}t^2 \quad (9.46)$$

$$C = 4.8314 \cdot 10^{-4} \quad (9.47)$$



Remarks:

- ◊ [Equation 9.45](#) is known as the International Equation of State for Seawater (EOS80) and is based on 467 data points. The standard error of the equation is $3.6 \cdot 10^{-3} \text{ kg/m}^3$ ([Millero and Poisson, 1981](#)).
- ◊ The Practical Salinity Scale ([UNESCO, 1981b](#), Par. 3.2) and the International Equation of State are meant for use in all oceanic waters. However, these equations should be used with caution in waters that have a chemical composition different from standard seawater. In such waters, densities derived with the methods based on practical salinity measurements and the International Equation of State may deviate measurably from the true densities. However, in water masses different in composition from standard seawater the *differences* in densities derived by the new equations involve only very small errors.

Recommendation

The UNESCO formulae serve as an international standard. Further, the UNESCO formulae show the correct temperature of 4 degrees Celsius where fresh water has its maximum density. The latter is of importance for thermal stratification in deep lakes in moderate climate zones. Therefore we recommend the application of the UNESCO formulae and to select the Eckart formulation only for consistence with previous projects in which it has been used. At this moment the default is UNESCO.

Nevertheless, the UNESCO formulae have their limitations as they are based on the general properties of seawater mixed with fresh water. Particularly in cases where marginal density differences play a role, typically lakes, a variable mineral content of the water may create density differences not detected by just temperature and salinity. For such dedicated cases, you are warranted to check the accuracy of the UNESCO formulae against experimental density-relations derived from the (lake) water. In case of deviations or other constituents determining the water density, we then recommend to contact the Helpdesk for further assistance such as the implementation of a more dedicated density formulation.

How to specify the equation of state

The equation of state cannot yet be specified from the FLOW-GUI. You have to edit the MDF-file yourself:

If you want to apply the Eckart formulation, add `DenFrm = #Eckart#`.

If you want to apply the UNESCO formulation you do not need to change the MDF-file, since the default formulation is UNESCO. However, you may add `DenFrm = #UNESCO#` if you want to.

9.4 Boundary conditions

The 3D and 2D depth-averaged shallow water or long-wave equations applied in Delft3D-FLOW represent a hyperbolic (inviscid case) or parabolic (viscid case) set of partial differential equations. To get a well-posed mathematical problem with a unique solution, a set of initial and boundary conditions for water levels and horizontal velocities must be specified. The contour of the model domain consists of parts along “land-water” lines (river banks, coast lines) which are called closed boundaries and parts across the flow field which are called open boundaries. Closed boundaries are natural boundaries. The velocities normal to a closed boundary are set to zero. Open boundaries are always artificial “water-water” boundaries. In a numerical model open boundaries are introduced to restrict the computational area and so the computational effort. They are situated as far away as possible from the area of interest. Long waves propagating out of the model area should not be hampered by the open boundaries. The reflection should be minimal.

The solution of the shallow water equations can be split in a steady state solution and a transient solution. The steady state solution depends on the boundary conditions and the forcing terms while the transient solution follows from the deviation between the initial condition and the steady state solution at the start of the simulation, the reflection at the open boundaries and the amount of dissipation.

Usually it is assumed that initially the model is at rest (“cold start”). The start of the simulation will generate short progressive waves with a wavelength coupled with the length of the model area (eigen oscillations). When disturbances are reflected at the open boundaries and are trapped in the model area, a standing wave may be generated. A water level boundary acts as a nodal point and a velocity or closed boundary acts as an anti-nodal point. The transient solution may propagate out of the model area through the open boundaries (Riemann boundary condition) or be reduced at the open boundaries (low pass filter) by use of the α -coefficient, Eqs. 9.74 and 9.75, or die out due to bottom friction and viscosity. The spin-up time of a model is dependent on the time-scale with which the transient solution dies out.

The steady state solution is completely dependent on the boundary conditions. It is reached after the transient solution died out. The number of boundary conditions specified at any particular point of the boundary should be equal to the number of characteristics entering the

region at that point, see [Courant and Hilbert \(1962\)](#) or [Vreugdenhil \(1989\)](#). Delft3D-FLOW assumes that the flow at the open boundaries is sub-critical, which means that the magnitude of the flow is smaller than the velocity of wave propagation. Sub-critical flow means that the Froude number, which is defined by:

$$Fr = \frac{|U|}{\sqrt{gH}}, \quad (9.48)$$

is smaller than one. For sub-critical flow we distinguish two situations, viz. inflow and outflow. At inflow, we have to specify two boundary conditions and at outflow we have to specify one boundary condition. For tidal flow the number of required boundary conditions varies between ebb and flood. The first boundary condition is an external forcing by the water level, the normal velocity, the discharge rate or the Riemann invariant specified by you. The second boundary condition is a built-in boundary condition. At inflow the velocity component along the open boundary is set to zero. The influence of this built-in boundary condition is in most cases restricted to only a few grid cells near the open boundary. It would be better to specify the tangential velocity component. The tangential velocity may be determined from measurements or from model results of a larger domain (nesting of models). In the present implementation of the open boundary conditions in Delft3D-FLOW it is not possible yet to specify the tangential velocity component at input. To get a realistic flow pattern near the open boundary, it is suggested to define the model boundaries at locations where the grid lines of the boundary are perpendicular to the flow.

For modelling the region outside the computational domain, incident waves should be known and are prescribed at the open boundaries. To reduce the reflections at the open boundaries a so-called weakly reflecting boundary condition may be applied. This boundary condition is derived using the Riemann invariants of the linearised shallow water equations without Coriolis and bottom friction ([Verboom and Slob, 1984](#)). The boundary condition is non-reflective for waves, which pass normal to the boundary (wave front parallel with open boundary). If we do not prescribe exactly the incoming Riemann invariants at an open boundary, the outgoing wave will reflect at the boundary and propagate back as a disturbance into the model area. Usually information on both water level and velocity is not available and consequently we cannot specify the Riemann invariant. A water level or velocity boundary condition can be made weakly reflective for short wave components originating from the initial conditions or the eigen frequencies by adding the time derivative of the Riemann invariant see Eqs. [9.74](#) and [9.75](#). If these Riemann invariants are not used then these short wave components may disturb the solution for a long time.

In the vertical direction the momentum equations are parabolic. The vertical velocity profile is determined by the vertical eddy viscosity and the boundary conditions at the bed (bed stress) and at the free surface (wind stress). The spin-up time of the vertical profiles is dependent on the vertical eddy viscosity.

When the grid size near the boundaries is larger than the thickness of the boundary layers occurring in the flow, the shear-stresses along the lateral boundaries can be neglected. A so-called free slip boundary condition is applied at all lateral boundaries. For the simulation of laboratory flumes the effect of side wall friction can be added to the system of equations in Delft3D-FLOW by a so-called partial slip boundary condition.

9.4.1 Flow boundary conditions

9.4.1.1 Vertical boundary conditions

Kinematic boundary conditions

In the σ co-ordinate system, the free surface ($\sigma = 0$, or $z = \zeta$) and the bottom ($\sigma = -1$, or $z = -d$) are σ co-ordinate surfaces. ω is the vertical velocity relative to the σ -plane. The impermeability of the surface and the bottom is taken into account by prescribing the following kinematic conditions:

$$\omega|_{\sigma=-1} = 0 \text{ and } \omega|_{\sigma=0} = 0 \quad (9.49)$$

For the Z -grid the kinematic conditions read:

$$w|_{z=-d} = 0 \text{ and } w|_{z=\zeta} = 0 \quad (9.50)$$

Bed boundary condition

At the seabed, the boundary conditions for the momentum equations are:

$$\frac{\nu_V}{H} \frac{\partial u}{\partial \sigma} \Big|_{\sigma=-1} = \frac{1}{\rho_0} \tau_{b\xi}, \quad (9.51)$$

$$\frac{\nu_V}{H} \frac{\partial v}{\partial \sigma} \Big|_{\sigma=-1} = \frac{1}{\rho_0} \tau_{b\eta}, \quad (9.52)$$

with $\tau_{b\xi}$ and $\tau_{b\eta}$ the components of the bed stress in ξ - and η -direction, respectively. The bed stress may be the combined effect of flow and waves. In this section we restrict ourselves to the resistance due to flow only. For the parameterisations for the combination of flow and waves, see [section 9.7](#). At first we discuss the formulations for 2D depth-averaged flow, followed by the formulations for 3D flow.

Depth-averaged flow

For 2D depth-averaged flow the shear-stress at the bed induced by a turbulent flow is assumed to be given by a quadratic friction law:

$$\vec{\tau}_b = \frac{\rho_0 g \vec{U} |\vec{U}|}{C_{2D}^2}, \quad (9.53)$$

where $|\vec{U}|$ is the magnitude of the depth-averaged horizontal velocity.

The 2D-Chézy coefficient C_{2D} can be determined according to one of the following three formulations:

◊ **Chézy formulation:**

$$C_{2D} = \text{Chézy coefficient [m}^{1/2}/\text{s}]. \quad (9.54)$$

◊ **Manning's formulation:**

$$C_{2D} = \frac{\sqrt[6]{H}}{n} \quad (9.55)$$

where:

H is the total water depth [m].
 n is the Manning coefficient [$m^{-1/3}s$].

◇ **White-Colebrook's formulation:**

$$C_{2D} = 18^{10} \log \left(\frac{12H}{k_s} \right) \quad (9.56)$$

where:

H is the total water depth
 k_s is the Nikuradse roughness length.

Alluvial bed properties and land use data can be used to determine the roughness and flow resistance. Specific functionality has been implemented to make the translation of bed and land use properties into roughness and flow resistance characteristics. See [subsection 9.11.2](#) for details.

Three-dimensional flow

For 3D models, a quadratic bed stress formulation is used that is quite similar to the one for depth-averaged computations. The bed shear stress in 3D is related to the current just above the bed:

$$\vec{\tau}_{b_{3D}} = \frac{g\rho_0 \vec{u}_b |\vec{u}_b|}{C_{3D}^2}, \quad (9.57)$$

with $|\vec{u}_b|$ the magnitude of the horizontal velocity in the first layer just above the bed. The contribution of the vertical velocity component to the magnitude of the velocity vector is neglected. The first grid point above the bed is assumed to be situated in the logarithmic boundary layer. Let Δz_b be the distance to the computational grid point closest to the bed, so:

$$\vec{u}_b = \frac{\vec{u}_*}{\kappa} \ln \left(1 + \frac{\Delta z_b}{2z_0} \right), \quad (9.58)$$

where z_0 is user-defined. In the numerical implementation of the logarithmic law of the wall for a rough bottom, the bottom is positioned at z_0 . The magnitude of the bottom stress is defined as:

$$|\vec{\tau}_b| = \rho_0 \vec{u}_* |\vec{u}_*|. \quad (9.59)$$

Using Eqs. 9.57 to 9.59, C_{3D} can be expressed in the roughness height z_0 of the bed:

$$C_{3D} = \frac{\sqrt{g}}{\kappa} \ln \left(1 + \frac{\Delta z_b}{2z_0} \right). \quad (9.60)$$

The parameter z_0 can be linked to the actual geometric roughness as a fraction of the RMS-value of the sub-grid bottom fluctuations. For rough walls [Nikuradse \(1933\)](#) found:

$$z_0 = \frac{k_s}{30}, \quad (9.61)$$

where k_s is known as the Nikuradse roughness length scale and has been determined experimentally.

Often a 3D-calculations are preceded by depth-averaged calculations. Then, the Chézy-coefficients C_{2D} may be used for calibration of the 3D model. Under the assumption of a logarithmic velocity profile [Equation 9.54](#), equality of bottom stress and the assumption $z_0 \ll H$, the magnitude of the depth-averaged velocity is given by:

$$|\vec{U}| = \frac{|\vec{u}_*|}{\kappa} \ln \left(1 + \frac{H}{ez_0} \right). \quad (9.62)$$

The Chézy coefficient C_{2D} can be converted into the bed roughness height z_0 using the relation:

$$z_0 = \frac{H}{e^{1+\frac{\kappa C_{2D}}{\sqrt{g}}} - e}. \quad (9.63)$$

The depth H in Eqs. [9.62](#) and [9.63](#) denotes the actual water depth. As a result the roughness height z_0 depends on the horizontal co-ordinates ξ , η and time t . Equality of the bed stress in 2D leads to the relation:

$$C_{2D} = \frac{\sqrt{g}}{\kappa} \ln \left(1 + \frac{H}{ez_0} \right). \quad (9.64)$$

Note: that the difference between the 2D- and the 3D-Chézy coefficient [9.64](#) and [9.60](#) depends on the relative thickness of the computational bed layer. Using a 2D simulation for the calibration of waterlevels is only accurate if the vertical velocity profile of the horizontal velocity is almost logarithmic. Examples are discharge waves in rivers, tidal flows in estuaries and seas. The method is not accurate for stratified flows and wind driven flows, because in these cases the vertical profile will deviate a lot from the logarithmic profile [Equation 9.54](#), so also [Equation 9.58](#) is not correct.



Surface boundary condition

At the free surface the boundary conditions for the momentum equations are:

$$\frac{\nu_V}{H} \frac{\partial u}{\partial \sigma} \Big|_{\sigma=0} = \frac{1}{\rho_0} |\vec{\tau}_s| \cos(\theta), \quad (9.65)$$

$$\frac{\nu_V}{H} \frac{\partial v}{\partial \sigma} \Big|_{\sigma=0} = \frac{1}{\rho_0} |\vec{\tau}_s| \sin(\theta), \quad (9.66)$$

where θ is the angle between the wind stress vector and the local direction of the grid-line η is constant. Without wind, the stress at the free surface is zero. The magnitude of the wind shear-stress is defined as:

$$|\vec{\tau}_s| = \rho_0 \vec{u}_{*s} |\vec{u}_{*s}|. \quad (9.67)$$

The magnitude is determined by the following widely used quadratic expression:

$$|\vec{\tau}_s| = \rho_a C_d U_{10}^2 \quad (9.68)$$

where:

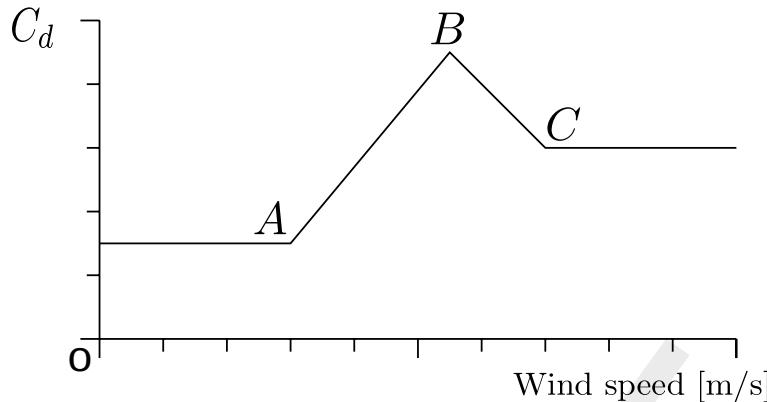


Figure 9.3: Examples wind drag coefficient depending on wind speed.

ρ_a	the density of air.
U_{10}	the wind speed 10 meter above the free surface (time and space dependent).
C_d	the wind drag coefficient, dependent on U_{10} .

Delft3D-FLOW offers the possibility to prescribe either global or local wind. Global wind corresponds to uniform in space and varying in time. Local winds vary both in space and time. Local wind is applied in combination with space and time varying atmospheric pressure.

The wind drag coefficient may be dependent on the wind velocity, reflecting increasing roughness of the sea surface with increasing wind speed, see [Smith and Banke \(1975\)](#). You may specify an empirical relation following:

$$C_d(U_{10}) = \begin{cases} C_d^A, & U_{10} \leq U_{10}^A, \\ C_d^A + (C_d^B - C_d^A) \frac{U_{10} - U_{10}^A}{U_{10}^B - U_{10}^A}, & U_{10}^A \leq U_{10} \leq U_{10}^B, \\ C_d^B + (C_d^C - C_d^B) \frac{U_{10} - U_{10}^B}{U_{10}^C - U_{10}^B}, & U_{10}^B \leq U_{10} \leq U_{10}^C, \\ C_d^C, & U_{10}^C \leq U_{10}, \end{cases} \quad (9.69)$$

where:

C_d^i	are the user-defined wind drag coefficients at respectively the wind speed U_{10}^i ($i = A, B, C$).
U_{10}^i	are user-defined wind speeds ($i = A, B, C$).

With [Equation 9.69](#) several relations for the wind drag coefficient are possible, see [Figure 9.3](#).

9.4.1.2 Open boundary conditions

Open boundaries are virtual “water-water” boundaries. They are introduced to obtain a limited computational area and so to reduce the computational effort. In nature, waves can cross these boundaries unhampered and without reflections. At an open boundary the water level, the normal velocity component or a combination should be prescribed to get a well-posed mathematical initial-boundary value problem. For an inflow boundary, also the tangential velocity component should be specified. In the present implementation, the tangential velocity component is set to zero. It is assumed that the flow is normal to the open boundary. The data needed for the boundary conditions can be obtained from measurements, tide tables or from a larger model, which encloses the model at hand (nesting). If we do not prescribe exactly

the incoming waves at an open boundary, the outgoing waves will reflect at the boundary and propagate as a disturbance into the area.

To reduce the reflections at the open boundary ([Verboom and Slob, 1984](#); [Verboom and Segal, 1986](#)) derived a so-called zero and first order weakly reflecting boundary condition based on the work of [Engquist and Majda \(1977, 1979\)](#). Assuming zero flow along the boundary, the zero order boundary condition may also be obtained using the so-called Riemann invariants for the linearised 1D equation normal to the open boundary:

$$R = U \pm 2\sqrt{gH}. \quad (9.70)$$

The two Riemann invariants are two waves moving in opposite direction with propagation speed $R = U \pm \sqrt{gH}$. The sign is dependent on the direction of propagation. At the open boundary, the incoming wave should be specified. We restrict ourselves to the positive sign (left boundary). The linearised Riemann invariant is given by:

$$U + 2\sqrt{gH} = U + 2\sqrt{g(d + \zeta)} \approx U + 2\sqrt{gd} + \zeta\sqrt{\frac{g}{d}}, \quad \frac{|\zeta|}{d} \ll 1. \quad (9.71)$$

The boundary condition which should be specified by you is: $f(t) = U + \zeta\sqrt{\frac{g}{d}}$, the term $2\sqrt{gd}$ is computed from the known depth-field and added in the computational part. It is assumed that the reference plane is chosen such that the mean water level is zero.

The boundary condition based on the Riemann invariant is weakly reflective in 1D. But in 2D this boundary condition is only weakly reflective for waves which pass normal to the boundary and if the Coriolis force and the bottom friction are negligible. Nevertheless, in practice it reduces the reflection also for waves, which have an oblique incidence. Usually we do not have information on both water level and velocity so we do not know the incoming Riemann invariant and another type of boundary should be chosen. A water level or velocity boundary condition can be made weakly reflective for short wave components originating from the initial conditions or the eigen frequencies by adding the time derivative of the Riemann invariant see Eqs. [9.74](#) and [9.75](#).

In the computational part the following type of boundary conditions are distinguished (for the sake of simplicity only a description for the U -direction is given here):

- ◊ Water level: $\zeta = F_\zeta(t) + \delta_{atm}$,
- ◊ Velocity (in normal direction): $U = F_U(t)$,
- ◊ Discharge (total and per cell): $Q = F_Q(t)$,
- ◊ Neumann $\frac{\partial \zeta}{\partial n} = f(t)$,
- ◊ Riemann invariant $U \pm \zeta\sqrt{\frac{g}{d}} = F_R(t)$.

If measured elevation data is not available, the barotropic forcing at the open boundaries is approximated by a linear superposition of:

- ◊ Free surface gradient for long-term circulation (geostrophic currents).
- ◊ Tidal fluctuations.
- ◊ Meteorological forcing (wind set-up).
- ◊ Waves (wave set-up).

In general, the boundary conditions are specified in a limited number of boundary points. Linear interpolation is used to generate the boundary conditions at the intermediate points along the boundary. This interpolation can generate physical unrealistic flows in the region close to the open boundary. The boundary conditions should allow these disturbances to

propagate out of the area or the number of points where the boundary condition is specified should be extended.

On water level boundaries, an input signal is prescribed that is consistent with some average pressure. Usually the signal corresponds to Mean Sea Level. One actually wants to prescribe an input signal corresponding to the local pressure prescribed by the space varying meteo input. To this end, it is possible to specify an average pressure ($p_{average}$) — which should correspond to your input signal on the open boundaries — which is then used to determine local pressure gradients that need to be applied along the open boundaries to obtain an input signal that is consistent with the local atmospheric pressure. This functionality need to be specified in the Data Group *Additional parameters*, using `PavBnd: Average Pressure on Boundaries`. Using this keyword one can specify an average pressure that is used on all water level open boundaries, independent of the type of wind input. The pressure must be specified in N/m².

$$\delta_{atm} = \frac{p_{average} - p_{atm}}{\rho g} \quad (9.72)$$

where

$p_{average}$	average pressure prescribed with the keyword <code>PavBnd</code> ,
p_{atm}	the local atmospheric pressure, given by the meteo module (see section B.7).
ρ	density of water
g	acceleration due to gravity

The same correction can also be used on Riemann boundaries, and is calculated as follows:

$$F_R(t) = F_R(t) \pm \frac{p_{average} - p_{atm}}{\rho g} \sqrt{\frac{g}{d}} \quad (9.73)$$

with d the local water depth, and the sign depends on the orientation of the open boundary related to the grid.

For the transport equation, concentrations should be specified at inflow, see [subsubsection 9.4.2.1](#), and the concentration should be free at outflow. For salinity and heat the concentration boundary conditions influence the flow by the baroclinic pressure term. At slack water, when the barotropic pressure gradients are small, this may lead to strange circulation cells along a long open boundary. Therefore an additional flag (keyword) has been implemented to switch off the baroclinic pressure term at the open boundary `BarocP = #NO#`. This is a global flag. The baroclinic pressure is switched off for all open boundaries. The default value is `#YES#`.

For the velocity, discharge and Riemann type of boundary condition, the flow is assumed to be perpendicular to the open boundary. Substitution of the Riemann boundary condition in the 1D linearised continuity equation leads to the well-known radiation boundary condition.

[Stelling \(1984\)](#) added the time-derivative of the Riemann invariant to the water level and velocity boundary conditions, to make the boundaries less reflective for disturbances with the eigen frequency of the model area. This reduces the spin-up time of a model from a cold start:

$$\text{Water level boundary: } \zeta + \alpha \frac{\partial}{\partial t} \left\{ U \pm 2\sqrt{gH} \right\} = F_\zeta(t), \quad (9.74)$$

$$\text{Velocity boundary: } U + \alpha \frac{\partial}{\partial t} \left\{ U \pm 2\sqrt{gH} \right\} = F_U(t). \quad (9.75)$$

The reflection coefficient α should be chosen sufficiently large to damp the short waves introduced at the start of the simulation. The following values are advised:

$$\text{Water level boundary: } \alpha = T_d \sqrt{\frac{H}{g}}, [\text{s}^2] \quad (9.76)$$

$$\text{Velocity boundary: } \alpha = T_d [\text{s}] \quad (9.77)$$

where T_d is the time it takes for a free surface wave to travel from the left boundary to the right boundary of the model area. In ocean and sea models, the period T_d is of the same order as the period of the tidal forcing. In that case α must be set to zero, otherwise effectively the amplitude of one of the components in the boundary condition is reduced. These values can be derived with Fourier analysis for the 1D linear long wave equation without advection by substituting a wave with period T_d .

For the velocity and discharge type of boundary condition, the flow is assumed to be perpendicular to the open boundary. In 3D models you can prescribe:

- ◊ a *uniform* profile
- ◊ a *logarithmic* profile

or

- ◊ a so-called *3D* profile

in the vertical. A 3D-profile means that the velocity at each σ -layer is specified as any of the forcing types, i.e. as harmonic components, time-series or astronomical components; see subsection 4.5.6 for details.

Remark:



- ◊ A 3D-profile can only be prescribed by importing a file containing the boundary condition at each layer. This file can be generated with the utility program Delft3D-NESTHD.

A logarithmic profile may be applied for velocity, discharge, or Riemann boundaries. The logarithmic profile depends on the local roughness height z_0 and the local water depth H . This profile is described for a velocity boundary condition in direction perpendicular to the open boundary. The vertical co-ordinate z is measured relative to the bed. The depth-averaged velocity \vec{U} is used to determine the friction velocity:

$$\vec{u}_* = \frac{\vec{U}}{\frac{1}{\kappa H} \int_0^H \ln\left(\frac{z+z_0}{z_0}\right) dz} = \frac{\vec{U}}{\frac{1}{\kappa H} \left[(H + z_0) \ln\left(\frac{H+z_0}{z_0}\right) - H \right]} \quad (9.78)$$

The velocity of layer k , u_k , is specified as:

$$\begin{aligned} \vec{u}_k &= \frac{\vec{u}_*}{\kappa (z_{k-1} - z_k)} \int_{z_k}^{z_{k-1}} \ln\left(\frac{z + z_0}{z_0}\right) dz, \\ &= \frac{\vec{u}_*}{\kappa (z_{k-1} - z_k)} \left[(z_{k-1} + z_0) \left\{ \ln\left(\frac{z_{k-1} + z_0}{z_0}\right) - 1 \right\} - (z_k + z_0) \left\{ \ln\left(\frac{z_k + z_0}{z_0}\right) - 1 \right\} \right], \end{aligned} \quad (9.79)$$

with (for σ -grid):

$$z_k = (1 + \sigma_{k-1}) H. \quad (9.80)$$

After substitution of u_* in this equation the layer velocity u_k is expressed in terms of U , H and z_0 .

The discharge is, in case of a total discharge boundary condition, at each time step distributed over the active points of the open boundary in the following manner derived from the Chézy equation:

$$q_i = \frac{B_i h_i^{1.5} C_i}{\sum_{j=1}^N B_j h_j^{1.5} C_j} Q. \quad (9.81)$$

where B_i , h_i and C_i are the width, water depth and roughness of grid cell i respectively, N is the number of boundary points, and Q is the total discharge imposed.

In case a QH-relation is used at a water level boundary the total discharge is computed from the discharges per grid cell

$$Q = \sum_{j=1}^N \left(B_i \sum_{k=1}^K u_{i,k} \Delta z_{i,k} \right) = \sum_{j=1}^N B_i \bar{u}_i h_i \quad (9.82)$$

where B_i , \bar{u}_i and h_i are the width, depth averaged velocity and water depth of grid cell i , u_i , k and $\Delta z_{i,k}$ are the velocity and thickness of layer k at grid cell i in case of a 3D simulation, N is the number of boundary points and K is the number of layers. The total discharge Q is used to derive a water level from the QH-relation specified by you, which is imposed uniformly at the boundary.

Neumann boundary conditions

Application of 2D and 3D hydrodynamic models to study the impact of engineering solutions such as the design of nourishments, groynes, offshore breakwaters on approximately uniform coasts is often hampered by the fact that they are considered to be difficult to set up and difficult to calibrate. One of the main problems is the specification of suitable boundary conditions at the open boundaries. This is due to a combination of processes acting on the model domain, a certain water level or velocity distribution will develop in cross-shore direction. For the boundary conditions to match this distribution the solution has to be known beforehand; if not, boundary disturbances will develop.

There are two ways to overcome this problem. The first is to try and predict the water level setup or the current velocity along the lateral boundary by solving a 1DH or 2DV problem along the boundary and to impose this. For simple cases this is possible but for more complex combinations of forcing conditions it is cumbersome. A better solution suggested by [Roelvink and Walstra \(2004\)](#) in the paper '*Keeping it simple by using complex models*' is to let the model determine the correct solution at the boundaries by imposing the alongshore water level gradient (a so-called Neumann boundary condition) instead of a fixed water level or velocity. In many cases the gradient can be assumed to be zero; only in tidal cases or in cases where a storm surge travels along a coast the alongshore gradient varies in time, but in model with a limited cross-shore extent, the alongshore gradient of the water level does not vary much in cross-shore direction.

The alongshore water level gradient can be assumed to be zero in case of a steady wind/wave field, or vary periodically for a tidal wave travelling along the coast. In the latter case, the alongshore gradient varies in time at the boundary. A constant periodical water level gradient for

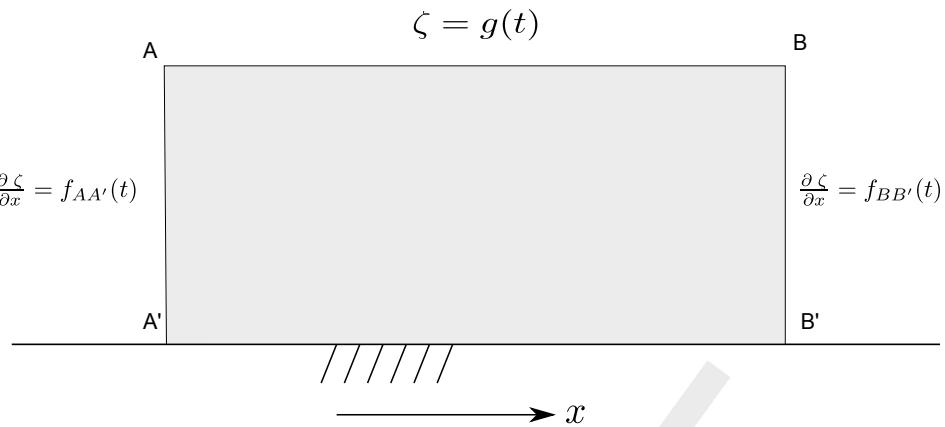


Figure 9.4: Hydrodynamic model of coastal area with three open boundaries with offshore boundary (A–B at deep water) and two cross shore boundaries (A–A', and B–B')

a boundary can be applied to models with a limited cross-shore extent, under the assumption that the alongshore gradient of the water level does not vary much in cross-shore direction. Neumann boundaries can only be applied on cross-shore boundaries in combination with a water level boundary at the offshore boundary, which is needed to make the solution of the mathematical boundary value problem well-posed (see Figure 9.4).

Defining boundary conditions with Neumann boundaries

Consider the example below; describing a small coastal region with a tidal wave travelling along the coast in eastward direction. On the cross-shore boundaries (sections A–A', B–B' in Figure 9.4), the water level gradient as a function of time, denoted as Neumann Boundary condition will be imposed: $\frac{\partial \zeta}{\partial x} = f(t)$. At the offshore boundary (A–B), the water level has to be prescribed to make the solution fully determined. At the water level boundary we set the reflection coefficient α to zero.

The offshore boundary is forced by a harmonic/astronomic water level according to a progressive wave given by:

$$\zeta = \sum_{j=1}^N \hat{\zeta}_j \cos(\omega_j t - k_j x) = \sum_{j=1}^N \hat{\zeta}_j \cos(\omega_j t - \varphi_j) \quad (9.83)$$

where: $\hat{\zeta}_j$ = amplitude [m]; $\omega_j = \frac{2\pi}{T_j}$ frequency [rad/h]; $k_j = \frac{2\pi}{L_j}$ wave number [rad/m]; Let d_{AB} be the distance between point A and B. The phase difference (φ_j) between points A and B is then given by $k_j d_{AB} = \frac{2\pi}{L_j} d_{AB}$ (to get degrees multiply it by $\frac{180}{\pi}$); L_j is the tidal wave length $L_j = \frac{2\pi}{\omega_j} c_j$, with c_j the velocity of the tidal wave. For shallow water the velocity is independent of the frequency and is given by \sqrt{gH} with H a characteristic water depth.

For the forcing of the two cross-shore boundaries a gradient (Neumann) type of boundary condition is used. The forcing term $f(t)$ is related to the offshore forcing by the relation:

$$\frac{\partial \zeta}{\partial x} = \sum_{j=1}^N k_j \hat{\zeta}_j \sin(\omega_j t - k_j x) = \sum_{j=1}^N k_j \hat{\zeta}_j \cos\left(\omega_j t - \left(\varphi_j + \frac{\pi}{2}\right)\right) \quad (9.84)$$

Table 9.4: Frequencies, phases and amplitude on alongshore waterlevel boundary and corresponding frequencies, phases and amplitudes for the cross-shore Neumann boundaries

	WL-boundary A–B	Neumann boundaries A–A', B–B'
Frequency [°/h]	$\omega_j(A) = \omega_j(B)$	$\omega_j(A) = \omega_j(A') =$ $\omega_j(B) = \omega_j(B')$
Phase [°]	$\varphi_j(A) = 0^\circ$ $\varphi_j(B) = \varphi_j$	$\varphi_j(A) = \varphi_j(A') = \pm 90^\circ$ $\varphi_j(B) = \varphi_j(B') = \pm 90^\circ + \varphi_j$
Amplitude A–B : [m] A–A', B–B' : [-]	$\hat{\zeta}_j(A) = \hat{\zeta}_j(B)$	$(\frac{\partial \zeta}{\partial x})_j(A) = (\frac{\partial \zeta}{\partial x})_j(A') = \frac{2\pi}{L_j} \hat{\zeta}_j$ $(\frac{\partial \zeta}{\partial x})_j(B) = (\frac{\partial \zeta}{\partial x})_j(B') = \frac{2\pi}{L_j} \hat{\zeta}_j$

In Table 9.4 we specify the relation between the tidal forcing at the offshore boundary (water level type) and the forcing at the cross-shore boundaries (Neumann type) for the coastal model of Figure 9.4.

General remarks on open boundaries

- ◊ At the start of the simulation, the boundary conditions often do not match with the initial conditions. Therefore, in Delft3D-FLOW at the open boundaries a linear interpolation is applied between the actual field value and the boundary condition, to reduce the spin up time of the model. The smoothing time is the last moment at which interpolation takes place.
- ◊ To define the phase difference in Delft3D-FLOW the unity *degrees* is used. To determine the amplitude of the water level gradient, radians should be used.
- ◊ The tidal wave length L_j may be estimated from the phase difference $\Delta\varphi_{AB} = \frac{2\pi}{L_j} d_{AB}$ between tidal stations along the boundary, [Roelvink and Walstra \(2004\)](#) or be approximated using $L_j = \frac{2\pi}{\omega_j} \sqrt{gH}$ with H a characteristic depth along the offshore boundary.
- ◊ If the tidal wave component is travelling in positive x -direction, the phase difference between the water level and Neumann boundary is $\pi/2$ ($\equiv +90^\circ$). If the tidal wave component is travelling in negative x -direction, the phase difference is $-\pi/2$ ($\equiv -90^\circ$).
- ◊ For each tidal component, the corresponding component of water level gradient can be determined on the basis of its frequency, phase difference between boundaries A and B and amplitude, see Table 9.4.
- ◊ The Neumann boundaries are interpreted as gradient type of boundaries for water level.
- ◊ Advection terms at the offshore boundary may generate an artificial boundary layer along the boundary. The advection terms containing normal gradients have to be switched off. This is done by adding an additional keyword to the MDF-file:
`Cstbnd =#YES#`
- ◊ When modelling salinity and/or heat transport, the boundary condition of these quantities should match their initial value. Otherwise the baroclinic pressure gradient can causes severe water level gradients, especially on deep water. These larger water level gradients generates waves which will propagate into the domain and enlarging the spin-up time of the model.

9.4.1.3 Shear-stresses at closed boundaries

A closed boundary is situated at the transition between land and water. At a closed boundary, two boundary conditions have to be prescribed. One boundary condition has to do with the flow normal to the boundary and the other one with the shear-stress along the boundary.

The boundary condition for flow normal to the boundary is:

- ◊ No flow through the boundary.

For the shear stress along the boundary one of the following conditions can be prescribed:

- 1 Zero tangential shear-stress (free slip).
- 2 Partial slip.

For large-scale simulations, the influence of the shear-stresses along closed boundaries can be neglected. Free slip is then applied for all closed boundaries. For simulations of small-scale flow (e.g. laboratory scale), the influence of the side walls on the flow may no longer be neglected. This option has been implemented for closed boundaries and dry points *but not for thin dams*. The reason is that the shear stress at a thin dam is dependent on the side (double valued).

Along the side walls, the tangential shear stress is calculated based on a logarithmic law of the wall:

$$\tau_{\xi\eta} = \tau_{\eta\xi} = \rho u_*^2. \quad (9.85)$$

The friction velocity u_* is determined by the logarithmic-law for a rough wall, with side wall roughness length z_{0n} and the velocity in the first grid point near the side wall. Let Δx_s be the grid size normal to the sidewall. Then:

$$|\vec{u}_{sidewall}| = \frac{u_*}{\kappa} \ln \left(1 + \frac{\Delta x_s}{2z_{0n}} \right). \quad (9.86)$$

9.4.2 Transport boundary conditions

9.4.2.1 Open boundary conditions for the transport equation

The horizontal transport of dissolved substances in rivers, estuaries, and coastal seas is dominated by advection. The horizontal diffusion in flow direction is of secondary importance. The consequence is that the transport equation is advection dominated, has a wave character, and from a mathematical point of view is of hyperbolic type. The equation has only one characteristic, which is parallel to the flow. Without diffusion, an observer moving with the flow, experiences a constant concentration. As described before, open boundaries have to limit the computational area. At an open boundary during inflow (flood) a boundary condition is needed. During outflow (ebb) the concentration must be free. At inflow we have to specify the concentration which may be determined by the concentration at outflow of the former ebb period. Because usually only the background concentration is known from measurements or from a larger model area, a special boundary condition based on the concentration in the interior area in combination with a return time is used, which does not completely fix the concentration at the background value. This is the so-called Thatcher-Harleman boundary condition ([Thatcher and Harleman, 1972](#)), which is discussed in the next section.

In Delft3D-FLOW, one of the following four vertical profiles may be specified at open boundaries:

1 Uniform profile

The concentration value specified is applied uniformly over the vertical.

2 Linear profile

The concentration value must be specified at the surface and near the bed. The intermediate values are then determined using linear interpolation:

$$c_k = c_{\text{bottom}} + (1 + \sigma_k) (c_{\text{surface}} - c_{\text{bottom}}) \quad (9.87)$$

or similarly for the Z -grid:

$$c_k = c_{\text{bottom}} + \left(\frac{d + z_k}{d + \zeta} \right) (c_{\text{surface}} - c_{\text{bottom}}) \quad (9.88)$$

3 Step profile

The concentration value must be specified both for the surface layer and the bed layer, together with the initial vertical position of the discontinuity. The concentration for an intermediate layer is determined following:

$$c_k = \begin{cases} c_{\text{surface}}, & z_k > z_d, \\ c_{\text{bottom}}, & z_k \leq z_d. \end{cases} \quad (9.89)$$

The z_d co-ordinate specifying the position of the discontinuity is used only initially to determine the index k of the related layer. The layer index of the discontinuity is kept fixed during the simulation. In case of σ -grid, the discontinuity is moving with the free surface, following the position of the corresponding σ -plane.

Remark:

- ◊ If the initial conditions are read from a restart file (hot start), then the position of the jump will be determined using the water elevation read from this file. Consequently, a mismatch between your boundary condition and the concentration distribution in the vertical, just inside the open boundary, may occur. This mismatch can produce oscillations in the simulation results.

4 3D profile

For a nested model Delft3D-FLOW offers the possibility to prescribe the concentration at every σ - or Z -layer using a time-series:

$$c_k = F_c(t). \quad (9.90)$$

9.4.2.2 Thatcher-Harleman boundary conditions

The transport of dissolved substances such as salt, sediment, and heat is described by the advection-diffusion equation. The horizontal transport is advection dominated and the equation is of hyperbolic type. At inflow, one boundary condition is needed and the concentration is specified. At outflow, no boundary condition is allowed. The concentration is determined by pure advection from the interior area:

$$\frac{\partial C}{\partial t} + \frac{U}{\sqrt{G_{\xi\xi}}} \frac{\partial C}{\partial \xi} = 0. \quad (9.91)$$

In Delft3D-FLOW the dispersive fluxes through the open boundaries at both inflow and outflow are zero:

$$\frac{D_H}{\sqrt{G_{\xi\xi}}} \frac{\partial C}{\partial \xi} = 0. \quad (9.92)$$

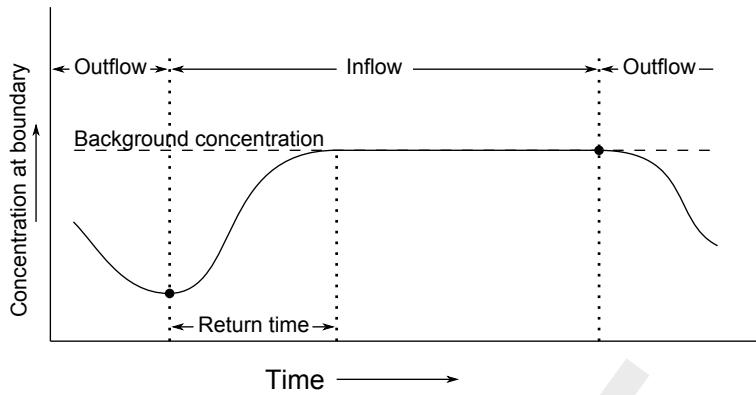


Figure 9.5: Illustration of memory effect for open boundary

If the concentration at outflow differs from the boundary condition at inflow, there is a discontinuity in the concentration at the turn of the flow. The transition of the concentration at the boundary from the outflow value to the inflow value may take some time. This depends on the refreshment of the water in the boundary region. The transition time is called the return time. The functional relationship describing the variation in concentration from the slack-water value to the present value is arbitrary. In Delft3D-FLOW, a half-cosine variation is used. After the return time, the boundary value remains constant until outward flow begins ([Leendertse and Gritton, 1971](#); [Thatcher and Harleman, 1972](#)). The mathematical formulation of this memory effect is given as follows:

$$C(t) = \begin{cases} C_{out} + \frac{1}{2}(C_{bnd} - C_{out}) \left(\cos\left(\pi \frac{\tau_{out} - \tau_{out}}{T_{ret}}\right) + 1 \right) & , \quad 0 \leq \tau_{out} \leq T_{ret} \\ C_{bnd} & , \quad \tau_{out} > T_{ret} \end{cases} \quad (9.93)$$

where

- C_{out} is the computed concentration at the open boundary at the last time of outward flow,
- C_{bnd} is the boundary concentration described by you,
- τ_{out} is the elapsed time since the last outflow $\tau_{out} = t - t_{out}$
- t_{out} the last time of outward flow, and
- T_{ret} is the constituent return period.

When the flow turns inward ($\tau_{out} = 0$), the concentration is set equal to C_{out} . During the interval $0 \leq \tau_{out} \leq T_{ret}$ the concentration will return to the background concentration C_{bnd} . After that period, the concentration will remain C_{bnd} . The mechanism is illustrated in [Figure 9.5](#).

For a stratified flow the return time of the upper layer will be longer than for the bottom layer. Delft3D-FLOW offers the opportunity to prescribe return times for the surface layer and the bed layer. For the intermediate layers the return time is determined using linear interpolation:

$$T_{TH_k} = T_{TH_{bottom}} + (1 + \sigma_k) (T_{TH_{surface}} - T_{TH_{bottom}}) , \quad (9.94)$$

or similarly for the Z -grid:

$$T_{TH_k} = T_{TH_{bottom}} + \left(\frac{d + z_k}{d + \zeta} \right) (T_{TH_{surface}} - T_{TH_{bottom}}) \quad (9.95)$$

with:

$T_{TH_{bottom}}$	return time bed layer.
$T_{TH_{surface}}$	return time surface layer.
z_k, σ_k	vertical position, respectively in Z - and σ co-ordinate system.
d	local depth value

Therefore an additional flag (key word) has been implemented to switch off the baroclinic pressure term at open boundaries: BaroCP = #NO#. This is a global flag. The baroclinic pressure is switched off for all open boundaries. The default value is #YES#.

9.4.2.3 Vertical boundary conditions transport equation

The vertical diffusive flux through the free surface and bed is zero with exception of the heat flux through the free surface. The mathematical formulations for the heat exchange at the free surface are given in [section 9.8](#):

$$\frac{D_V}{H} \frac{\partial c}{\partial \sigma} \Big|_{\sigma=0} = 0, \quad (9.96)$$

$$\frac{D_V}{H} \frac{\partial c}{\partial \sigma} \Big|_{\sigma=-1} = 0. \quad (9.97)$$

9.5 Turbulence

Delft3D-FLOW solves the Navier-Stokes equations for an incompressible fluid. Usually the grid (horizontal and/or vertical) is too coarse and the time step too large to resolve the turbulent scales of motion. The turbulent processes are “sub-grid”. The primitive variables are space- and time-averaged quantities. Filtering the equations leads to the need for appropriate closure assumptions.

For 3D shallow water flow the stress and diffusion tensor are an-isotropic. The horizontal eddy viscosity coefficient ν_H and eddy diffusivity coefficient D_H are much larger than the vertical coefficients ν_V and D_V , i.e. $\nu_H \gg \nu_V$ and $D_H \gg D_V$. The horizontal coefficients are assumed to be a superposition of three parts:

- 1 a part due to molecular viscosity.
- 2 a part due to “2D-turbulence”,
- 3 a part due to “3D-turbulence” see [Uittenbogaard et al. \(1992\)](#) and

The 2D part is associated with the contribution of horizontal motions and forcings that cannot be resolved (“sub-grid scale turbulence”) by the horizontal grid (Reynolds averaged or eddy resolving computations). The 3D part is referred to as the three-dimensional turbulence and is computed following one of the turbulence closure models, described in this section. For 2D depth-averaged simulations, the horizontal eddy viscosity and eddy diffusivity coefficient should also contain a contribution due to the vertical variation of the horizontal flow (Taylor shear dispersion).

The background horizontal viscosity coefficient ν_H^{back} and eddy diffusivity coefficient D_H^{back} (constant or space-varying) must be specified by you via the Delft3D-FLOW GUI. Delft3D-FLOW additionally has a sub-grid scale model, HLES for 2D-turbulence (see [section B.8](#)). These coefficients can also be used to damp small-scale noise introduced by the advection terms. They must be chosen dependent on the grid size. Furthermore the horizontal coefficients are an order of magnitude larger than the vertical coefficients determined by the turbulence closure model specified by you.

In Delft3D-FLOW, four turbulence closure models have been implemented to determine ν_V and D_V :

- 1 Constant coefficient.
- 2 Algebraic Eddy viscosity closure Model (AEM).
- 3 $k-L$ turbulence closure model.
- 4 $k-\varepsilon$ turbulence closure model.

The turbulence closure models differ in their prescription of the turbulent kinetic energy k , the dissipation rate of turbulent kinetic energy ε , and/or the mixing length L .

The first turbulence closure model is the simplest closure based on a constant value which has to be specified by you. We remark that a constant eddy viscosity will lead to parabolic vertical velocity profiles (laminar flow). The other three turbulence closure models are based on the so-called eddy viscosity concept of [Kolmogorov \(1942\)](#) and [Prandtl \(1945\)](#). The eddy viscosity is related to a characteristic length scale and velocity scale. The eddy viscosity has the following form:

$$\nu_{3D} = c'_\mu L \sqrt{k}, \quad (9.98)$$

where:

- | | |
|----------|--|
| c'_μ | a constant determined by calibration, derived from the empirical constant c_μ in the $k-\varepsilon$ model; $c'_\mu = c_\mu^{1/4}$, $c_\mu = 0.09$ (Rodríguez et al. 1984), |
| L | is the mixing length, and |
| k | is the turbulent kinetic energy. |

The algebraic eddy viscosity model (AEM) does not involve transport equations for the turbulent quantities. This so-called zero order closure scheme is a combination of two algebraic formulations. This model uses analytical (algebraic) formulas to determine k and L . The turbulent kinetic energy k depends on the (friction) [velocities or velocity gradients](#) and for the mixing length, L , the following function of the depth is taken ([Bakhmetev, 1932](#)):

$$L = \kappa (z + d) \sqrt{1 - \frac{z + d}{H}}, \quad (9.99)$$

with κ the Von Kármán constant, $\kappa \approx 0.41$. For a homogeneous flow this leads to a logarithmic velocity profile.

In case of vertical density gradients, the turbulent exchanges are limited by buoyancy forces and the mixing length L of [Equation 9.99](#) must be corrected. [On the other hand high velocity gradients, increase turbulent mixing and can weaken the stratification](#). Stratification stability can be described in the interaction between gravitational forces (buoyancy flux B_k , [Equation 9.120](#)) and turbulent shear production (P_k , [Equation 9.117](#)). The stratification is characterised by the gradient Richardson number, Ri , defined by:

$$Ri = \frac{-g \frac{\partial \rho}{\partial z}}{\rho \left[\left(\frac{\partial u}{\partial z} \right)^2 + \left(\frac{\partial v}{\partial z} \right)^2 \right]}, \quad (9.100)$$

see e.g. [Richardson \(1920\)](#); [Taylor \(1931\)](#); [Miles \(1987\)](#). For $Ri \geq 0$ the stratification is stable whereby for $Ri < 0$ the stratification is unstable. Stable stratification leads to damping of turbulent mixing while unstable stratification leads to higher mixing. Mathematically this is

described by a so-called damping function $F_L(Ri)$ that depends on the gradient Richardson number Ri ([Simonin et al., 1989](#)):

$$L = \kappa(z + d) \sqrt{1 - \frac{z + d}{H}} F_L(Ri). \quad (9.101)$$

These damping functions have been determined by fitting mathematical functions which fulfil the limiting conditions to laboratory data sets. Different formulations have been suggested. In Delft3D-FLOW the algebraic eddy viscosity model (AEM) is extended to stratified flows by the formulation of [Busch \(1972\)](#):

$$F_L(Ri) = \begin{cases} e^{-2.3Ri}, & Ri \geq 0, \\ (1 - 14Ri)^{0.25}, & Ri < 0. \end{cases} \quad (9.102)$$

The third closure model for the eddy viscosity involves one transport equation for k and is called a first order turbulence closure scheme. The mixing length L is prescribed analytically and the same formulation, including damping functions, is used as for the AEM turbulence model. However, to find the kinetic energy k , a transport equation is solved. This turbulence closure model is known as the k - L model.

The fourth model is the k - ε model that is a second order turbulence closure model. In this model both the turbulence energy k and dissipation rate of turbulent kinetic energy ε are calculated by a transport equation. From k and ε the mixing length L and viscosity are determined. The mixing length is now a property of the flow, and in the case of stratification no damping functions are needed.

A brief description of each of these turbulence closure model will be given further on in this section, for more details we refer to [Uittenbogaard et al. \(1992\)](#). The k - ε model has been used in tidal simulations by [Baumert and Radach \(1992\)](#) and [Davies and Gerritsen \(1994\)](#), for stratified flow of a river plume by [Postma et al. \(1999\)](#) and for the evolution of a thermocline by [Burchard and Baumert \(1995\)](#).

For the combination of flow and waves, the effect of the orbital velocities on the production of turbulence is modelled indirectly by the enhancement of the bed roughness; see [section 9.7](#). The shear production is determined only from the wave-averaged velocities.

The turbulence closure models do not take into account the vertical mixing induced by shearing and breaking of short and random internal gravity waves. This internal-wave-induced mixing can be determined using an additional transport equation for Internal Wave Energy (IWE-model) ([Uittenbogaard and Baron, 1989](#); [Uittenbogaard, 1995](#)). The IWE model describes the energy transfer from internal waves to turbulence as well as the excitation of internal waves by turbulence. The simplest approximation of the effect of internal waves is the introduction of constant vertical ambient mixing coefficients of momentum ν , and/or heat and matter D .

For strongly stratified flows it is important to introduce suitably chosen constant ambient (background) mixing coefficients, because the mixing coefficients computed by turbulence models with shear production only, reduce to zero. In the latter situation the vertical layers are completely de-coupled (frictionless). Disturbances are hardly damped and also the erosion of the vertical stratification is reduced to molecular diffusion.

Based on our experience with highly stratified flows we suggest applying an ambient or background vertical eddy viscosity in the order of 10^{-4} m²/s for the vertical exchange of mo-

mentum. This value corresponds with field measurements in the Rotterdam Waterway, The Netherlands.

Eddy diffusivity

The vertical eddy diffusivity is a scaled form of the eddy viscosity according to:

$$D_{3D} = \frac{\nu_{3D}}{\sigma_c}. \quad (9.103)$$

Parameter σ_c is the Prandtl-Schmidt number. Its numerical value depends on the substance C . Moreover, for zero order models (for example, the algebraic turbulence closure model), it is affected by the stratification. It slightly widens the applicability of an algebraic model for weakly stratified flows. In general the expression for the Prandtl-Schmidt number reads:

$$\sigma_c = \sigma_{c_0} F_\sigma(Ri). \quad (9.104)$$

The σ_{c_0} in this expression depends on the substance. It is a constant. $F_\sigma(Ri)$ is a so-called damping function that depends on the density stratification via the gradient Richardson's number Ri defined in [Equation 9.100](#), see [Simonin et al. \(1989\)](#).

In Delft3D-FLOW the following settings of σ_{c_0} and $F_\sigma(Ri)$ are used:

- ◊ In all cases, regardless the turbulence closure model, $\sigma_{c_0} = 0.7$ for the transport of heat, salinity, and tracer concentrations. For suspended sediment concentrations in online sediment transport computations, $\sigma_{c_0} = 1.0$.
- ◊ For the transport of turbulent kinetic energy k in the $k-L$ model and $k-\varepsilon$ model $\sigma_{c_0} = 1.0$, and for the transport of turbulent kinetic energy dissipation ε in the $k-\varepsilon$ model $\sigma_{c_0} = 1.3$.
- ◊ In case the algebraic turbulence closure model is applied, the damping function $F_\sigma(Ri)$ is the following Munk-Anderson formula ([Munk and Anderson, 1948](#)):

$$F_\sigma(Ri) = \begin{cases} \frac{(1 + 3.33Ri)^{1.5}}{\sqrt{1 + 10Ri}}, & Ri \geq 0 \\ 1, & Ri < 0. \end{cases} \quad (9.105)$$

The damping functions Eqs. 9.102 and 9.105 were obtained by measurements of steady-state stable stratified turbulent shear flows. In the mathematical formulation, the fluxes are instantaneously influenced by changes in the vertical gradients of velocity and density. A physical adjustment time of the turbulence to the variations of the vertical gradients, is not taken into account. The fluxes are not a monotone function of the gradients. For the transport equation of heat, for small temperature gradients the heat flux increases when the temperature gradient increases but for large temperature gradients the heat flux decreases because the vertical eddy diffusivity is damped. For large values of the density gradients and small values of the velocity gradients, the vertical diffusion equation becomes mathematically ill-posed [Barenblatt et al. \(1993\)](#), and the computed vertical profiles may become discontinuous (stepwise). The number of "steps" is dependent on the vertical grid. For strongly stratified flows ($Ri > 1$), the $k-\varepsilon$ model is advised.

In the $k-\varepsilon$ turbulence model no damping function is used since the influence of stratification on the mixing length is taken into account by the buoyancy terms in the transport equations for k and ε . Thus: $F_\sigma(Ri) = 1$ for all Ri . This is one of the main advantages of the $k-\varepsilon$ turbulence model. Damping functions, as used in the standard Level 2.5 turbulence closure

model of [Mellor and Yamada \(1974, 1982\)](#), may generate unphysical “stepwise” profiles as observed by [Deleersnijder and Luyten \(1994\)](#).

The numerical scheme for the vertical advection of heat and salt (central differences) may introduce small vertical oscillations. This computational noise may enhance the turbulent mixing. Delft3D-FLOW has a vertical filtering technique to remove this noise and to reduce the undesirable mixing. For more details, see [chapter 10](#).

In strongly-stratified flows, the turbulent eddy viscosity at the interface reduces to zero and the vertical mixing reduces to molecular diffusion. To account for the vertical mixing induced by shearing and breaking of short and random internal gravity waves, we suggest to apply an ambient eddy diffusivity in the order of 10^{-4} to 10^{-5} m²/s dependent on the Prandtl-Schmidt number. In Delft3D-FLOW for stable stratified flows, the minimal eddy diffusivity may be based on the Ozmidov length scale L_{oz} , specified by you and the Brunt-Väisälä frequency of internal waves:

$$D_V = \max \left(D_{3D}, 0.2L_{oz}^2 \sqrt{-\frac{g}{\rho} \frac{\partial \rho}{\partial z}} \right). \quad (9.106)$$

For a detailed description of the turbulence closure models of Delft3D-FLOW we refer to [Rodó \(1984\)](#) and [Uittenbogaard et al. \(1992\)](#).

9.5.1 Algebraic turbulence model (AEM)

The Algebraic Eddy viscosity Model (AEM) in Delft3D-FLOW is a combination of two zero order closure schemes. These models will be denoted by ALG (Algebraic closure model) and PML (Prandtl's Mixing Length model). Both models will be briefly described. The combination of the two models was made to broaden the applicability of the algebraic turbulence closure.

9.5.1.1 Algebraic closure model (ALG)

In the algebraic closure model, ALG, a logarithmic velocity profile is assumed. This leads to a linear relation between the turbulent kinetic energy at the bed and the turbulent kinetic energy at the free surface:

$$k = \frac{1}{\sqrt{c_\mu}} \left[(u_*^b)^2 \left(1 - \frac{z+d}{H} \right) + u_{*s}^2 \frac{z+d}{H} \right], \quad (9.107)$$

with:

c_μ A constant, $c_\mu \approx 0.09$, calibrated for local-equilibrium shear layers ([Rodó, 1984](#)).

u_{*s} The friction velocity at the free surface, see [Equation 9.67](#).
 u_*^b A modified form of the bed friction velocity u_{*b} .

The turbulent kinetic energy k at the bed and at the free surface is determined basing on the shear stresses. The bed friction velocity u_{*b} is determined from the magnitude of the velocity in the first grid point above the bed, under the assumption of a logarithmic velocity profile:

$$\vec{u}_{*b} = \frac{\kappa}{\ln \left(1 + \frac{\Delta z_b}{2z_0} \right)} \vec{u}_b. \quad (9.108)$$

In tidal flows, at slack water the velocities are almost zero near the bed. Higher in the water column, the turbulence intensity may still be large. To avoid a zero eddy viscosity, velocities

higher in the water column have to be used in the determination of the bed friction velocity. Under the assumption that the velocity profile is logarithmic in the discrete model for layer with index k :

$$\vec{u}_{*k} = \frac{\kappa}{\ln \left(1 + \frac{z_k + d}{z_0} \right)} \vec{u}_k, \quad (9.109)$$

and the average value is defined as:

$$\vec{u}_* = \frac{1}{K} \sum_{k=1}^K \vec{u}_{*k}. \quad (9.110)$$

In the linear profile for the turbulence energy, [Equation 9.107](#),

$$\vec{u}_*^b = \max(\vec{u}_*, \vec{u}_{*b}) \quad (9.111)$$

is used. The friction velocity at the free surface is dependent on the wind velocity at 10 m above the free surface [Eqs. 9.67 to 9.69](#).

In this algebraic turbulence closure model the wind stress and the bed stress affect the eddy viscosity over the total water depth instantaneously. In deep areas, this is physically not correct; the turbulent kinetic energy generated at the free surface and at the bed must be transported along the water column by vertical diffusion. This will lead to phase differences in the time-series of the turbulent energy for the different layers, see [Baumert and Radach \(1992\)](#).

Given k of [Equation 9.107](#) and L of [Equation 9.101](#) the eddy viscosity ν_{ALG} is prescribed by [Equation 9.98](#). The eddy viscosity is parabolic and gives in the case of wind-driven circulation, a double logarithmic vertical velocity profile.

9.5.1.2 Prandtl's Mixing Length model (PML)

A second algebraic expression for the eddy viscosity is obtained by assuming instantaneous local equilibrium between production and dissipation in the k - L model. This closure scheme is known as Prandtl's Mixing Length model (PML). In the production term the horizontal derivatives are left out, they are small compared to the vertical derivatives, leading to:

$$k = \frac{1}{\sqrt{c_\mu}} L^2 \left[\left(\frac{\partial u}{\partial z} \right)^2 + \left(\frac{\partial v}{\partial z} \right)^2 \right]. \quad (9.112)$$

The mixing length L is again prescribed by [Equation 9.101](#). Given k of [Equation 9.112](#) and L of [Equation 9.101](#) the eddy viscosity ν_{PML} is prescribed by [Equation 9.98](#). Note that for a logarithmic velocity profile, without wind, [Eqs. 9.107](#) and [9.112](#) give the same linear distribution of k .

For the turbulence closure scheme ALG, the turbulent kinetic energy over the whole water column is in phase with the bed stress. For the PML model there is a phase difference in k , introduced by phase differences in the vertical gradient of the horizontal velocities. The phase differences depend on the total water depth and the eddy viscosity:

$$T \approx \frac{H^2}{\nu_V}, \quad (9.113)$$

see e.g. [Baumert and Radach \(1992\)](#).

The PML-model leads to a zero eddy viscosity and eddy diffusivity at the position in the vertical where the vertical gradients of the velocity are zero (e.g. in wind-driven flows). This is physically incorrect.

For robustness, in Delft3D-FLOW the turbulence closure models PML and ALG are combined to one algebraic model, denoted as the AEM model. The eddy viscosity is calculated following:

$$\nu_{3D} = \max(\nu_{ALG}, \nu_{PML}) \quad (9.114)$$

The eddy diffusivity is derived from the eddy viscosity, using the Prandtl-Schmidt number. Tidal simulations using this parameterisation have been performed for the Irish Sea ([Davies and Gerritsen, 1994](#)).

9.5.2 k - L turbulence model

A so-called first order turbulence closure scheme implemented in Delft3D-FLOW is the k - L model. In this model the mixing length L is prescribed analytically, [Equation 9.101](#). The velocity scale is based on the kinetic energy of turbulent motion. The turbulent kinetic energy k follows from a transport equation that includes an energy dissipation term, a buoyancy term and a production term. The following two assumptions are made:

- ◊ The production, buoyancy and dissipation terms are the dominating terms.
- ◊ The horizontal length scales are much larger than the vertical ones (shallow water, boundary layer type of flows).

The conservation of the turbulent quantities is less important and the transport equation is implemented in a non-conservative form. The second assumption leads to simplification of the production term. The transport equation for k is as follows:

$$\begin{aligned} \frac{\partial k}{\partial t} + \frac{u}{\sqrt{G_{\xi\xi}}} \frac{\partial k}{\partial \xi} + \frac{v}{\sqrt{G_{\eta\eta}}} \frac{\partial k}{\partial \eta} + \frac{\omega}{d + \zeta} \frac{\partial k}{\partial \sigma} = \\ + \frac{1}{(d + \zeta)^2} \frac{\partial}{\partial \sigma} \left(D_k \frac{\partial k}{\partial \sigma} \right) + P_k + P_{kw} + B_k - \varepsilon. \end{aligned} \quad (9.115)$$

with

$$D_k = \frac{\nu_{mol}}{\sigma_{mol}} + \frac{\nu_{3D}}{\sigma_k} \quad (9.116)$$

In the production term P_k of turbulent kinetic energy, the horizontal gradients of the horizontal velocity and all the gradients of the vertical velocities are neglected. The production term is given by:

$$P_k = \nu_{3D} \frac{1}{(d + \zeta)^2} \left[\left(\frac{\partial u}{\partial \sigma} \right)^2 + \left(\frac{\partial v}{\partial \sigma} \right)^2 \right]. \quad (9.117)$$

For small-scale applications (e.g. simulation of laboratory flume), you can switch on a more extended production term P_k of turbulent kinetic energy (option “partial slip”, rough side wall)

given by:

$$P_k = 2\nu_{3D} \left[\frac{1}{2(d+\zeta)^2} \left\{ \left(\frac{\partial u}{\partial \sigma} \right)^2 + \left(\frac{\partial v}{\partial \sigma} \right)^2 \right\} + \left(\frac{1}{\sqrt{G_{\xi\xi}}} \frac{\partial u}{\partial \xi} \right)^2 \right] + 2\nu_{3D} \left[\frac{1}{2} \left(\frac{1}{\sqrt{G_{\eta\eta}}} \frac{\partial u}{\partial \eta} + \frac{1}{\sqrt{G_{\xi\xi}}} \frac{\partial v}{\partial \xi} \right)^2 + \left(\frac{1}{\sqrt{G_{\eta\eta}}} \frac{\partial v}{\partial \eta} \right)^2 \right]. \quad (9.118)$$

In this expression, ν_{3D} is the vertical eddy viscosity, prescribed by [Equation 9.98](#). In Eqs. 9.117 and 9.118 it has been assumed that the gradients of the vertical velocity w can be neglected with respect to the gradients of the horizontal velocity components u and v . The horizontal and vertical (σ -grid) curvature of the grid has also been neglected.

The turbulent energy production due to wave action is given by P_{kw} ; the exact formulation is in [subsection 9.7.4](#).

Near the closed walls the normal derivative of the tangential velocity is determined with the law of the wall:

$$\frac{\partial u}{\partial y} = \frac{u_*}{\kappa y}. \quad (9.119)$$

In stratified flows, turbulent kinetic energy is converted into potential energy. This is represented by a buoyancy flux B_k defined by:

$$B_k = \frac{\nu_{3D}}{\rho \sigma_\rho} \frac{g}{H} \frac{\partial \rho}{\partial \sigma} \quad (9.120)$$

with the Prandtl-Schmidt number $\sigma_\rho = 0.7$ for salinity and temperature and $\sigma_\rho = 1.0$ for suspended sediments.

In the k - L model, it is assumed that the dissipation ε depends on the mixing length L and kinetic turbulent energy k according to:

$$\varepsilon = c_D \frac{k\sqrt{k}}{L}, \quad (9.121)$$

where c_D is a constant determined by calibration, derived from the constant c_μ in the k - ε model:

$$c_D = c_\mu^{3/4} \approx 0.1925. \quad (9.122)$$

To solve the transport equation, boundary conditions must be specified. A local equilibrium of production and dissipation of kinetic energy is assumed at the bed which leads to the following Dirichlet boundary condition:

$$k|_{\sigma=-1} = \frac{u_{*b}^2}{\sqrt{c_\mu}}. \quad (9.123)$$

The friction velocity u_{*b} at the bed is determined from the magnitude of the velocity in the grid point nearest to the bed, under the assumption of a logarithmic velocity profile, see [Equation 9.58](#). The bed roughness (roughness length) may be enhanced by the presence of wind generated short crested waves, see [section 9.7](#).

In case of wind forcing, a similar Dirichlet boundary condition is prescribed for the turbulent kinetic energy k at the free surface:

$$k|_{\sigma=0} = \frac{u_{*s}^2}{\sqrt{c_\mu}}. \quad (9.124)$$

In the absence of wind, the turbulent kinetic energy k at the surface is set to zero.

At open boundaries, the turbulent energy k is computed using the equation for k [Equation 9.107](#) without horizontal advection. For a logarithmic velocity profile this will approximately lead to the following linear distribution based on the shear-stress at the bed and at the free surface:

$$k(z) = \frac{1}{\sqrt{c_\mu}} \left[u_{*b}^2 \left(1 - \frac{z+d}{H} \right) + u_{*s}^2 \frac{z+d}{H} \right]. \quad (9.125)$$

9.5.3 $k-\varepsilon$ turbulence model

In the $k-\varepsilon$ turbulence model, transport equations must be solved for both the turbulent kinetic energy k and for the energy dissipation ε . The mixing length L is then determined from ε and k according to:

$$L = c_D \frac{k\sqrt{k}}{\varepsilon}. \quad (9.126)$$

In the transport equations, the following two assumptions are made:

- ◊ The production, buoyancy, and dissipation terms are the dominating terms.
- ◊ The horizontal length scales are larger than the vertical ones (shallow water, boundary layer type of flows).

Because of the first assumption, the conservation of the turbulent quantities is less important and the transport equation is implemented in a non-conservation form.

The transport equations for k and ε are non-linearly coupled by means of their eddy diffusivity D_k , D_ε and the dissipation terms. The transport equations for k and ε are given by:

$$\begin{aligned} \frac{\partial k}{\partial t} + \frac{u}{\sqrt{G_{\xi\xi}}} \frac{\partial k}{\partial \xi} + \frac{v}{\sqrt{G_{\eta\eta}}} \frac{\partial k}{\partial \eta} + \frac{\omega}{d+\zeta} \frac{\partial k}{\partial \sigma} = &+ \\ &+ \frac{1}{(d+\zeta)^2} \frac{\partial}{\partial \sigma} \left(D_k \frac{\partial k}{\partial \sigma} \right) + P_k + P_{kw} + B_k - \varepsilon, \end{aligned} \quad (9.127)$$

$$\begin{aligned} \frac{\partial \varepsilon}{\partial t} + \frac{u}{\sqrt{G_{\xi\xi}}} \frac{\partial \varepsilon}{\partial \xi} + \frac{v}{\sqrt{G_{\eta\eta}}} \frac{\partial \varepsilon}{\partial \eta} + \frac{\omega}{d+\zeta} \frac{\partial \varepsilon}{\partial \sigma} = & \\ &+ \frac{1}{(d+\zeta)^2} \frac{\partial}{\partial \sigma} \left(D_\varepsilon \frac{\partial \varepsilon}{\partial \sigma} \right) + P_\varepsilon + P_{\varepsilon w} + B_\varepsilon - c_{2\varepsilon} \frac{\varepsilon^2}{k}. \end{aligned} \quad (9.128)$$

with

$$D_k = \frac{\nu_{mol}}{\sigma_{mol}} + \frac{\nu_{3D}}{\sigma_k} \quad \text{and} \quad D_\varepsilon = \frac{\nu_{3D}}{\sigma_\varepsilon} \quad (9.129)$$

The production term P_k is defined in Eqs. 9.117 and 9.118, and the buoyancy term B_k is defined in [Equation 9.120](#). The production term P_ε and the buoyancy flux B_ε are defined by:

$$P_\varepsilon = c_{1\varepsilon} \frac{\varepsilon}{k} P_k, \quad (9.130)$$

$$B_\varepsilon = c_{1\varepsilon} \frac{\varepsilon}{k} (1 - c_{3\varepsilon}) B_k, \quad (9.131)$$

with L prescribed by [Equation 9.126](#) and the calibration constants by ([Rodí, 1984](#)):

$$c_{1\varepsilon} = 1.44, \quad (9.132)$$

$$c_{2\varepsilon} = 1.92, \quad (9.133)$$

$$c_{3\varepsilon} = \begin{cases} 0.0 & \text{unstable stratification} \\ 1.0 & \text{stable stratification} \end{cases} \quad (9.134)$$

In Delft3D-FLOW in the ε -equation for stable stratification the buoyancy flux is switched off, so $c_{3\varepsilon} = 1.0$ and for unstable stratification the buoyancy flux is switched on $c_{3\varepsilon} = 0.0$.

The energy production and energy dissipation due to waves, the terms P_{kw} and $P_{\varepsilon w}$ in Eqs. 9.127 and 9.128, are given in [subsection 9.7.4](#).

The coefficients of the 3D k - ε turbulence closure model as implemented in Delft3D-FLOW are not the same as in the depth-averaged k - ε turbulence closure model ([Rodí, 1984](#)), therefore for depth-averaged simulations, the k - ε turbulence closure model is not available for you.

The vertical eddy viscosity ν_{3D} is determined by:

$$\nu_{3D} = c'_\mu L \sqrt{k} = c_\mu \frac{k^2}{\varepsilon}, \quad (9.135)$$

with:

$$c_\mu = c_D c'_\mu. \quad (9.136)$$

For the transport equation of the turbulent kinetic energy k the same Dirichlet boundary conditions are used as for the k - L model described earlier, see [Equation 9.123](#) and [9.124](#). For the transport equation of the dissipation ε , the following boundary condition is used at the bed:

$$\varepsilon|_{\sigma=-1} = \frac{u_{*b}^3}{\kappa z_0}, \quad (9.137)$$

and at the surface the dissipation ε is prescribed by:

$$\varepsilon|_{\sigma=0} = \frac{u_{*s}^3}{\frac{1}{2} \kappa \Delta z_s}. \quad (9.138)$$

In case of no wind, the dissipation ε is set to zero at the free surface.

At open boundaries, the energy dissipation ε is computed using [Equation 9.128](#) without horizontal advection. For a logarithmic velocity profile this will approximately lead to a hyperbolic

distribution which is the superposition of two hyperbola, corresponding to a double logarithmic velocity profile, on the basis of the shear-stresses at the bed and at the free surface:

$$\varepsilon(z) = \frac{u_{*b}^3}{\kappa(z+d)} + \frac{u_{*s}^3}{\kappa(H-z-d)}, \quad (9.139)$$

where z denotes the vertical co-ordinate.

The $k-\varepsilon$ turbulence model was successfully applied for the simulation of stratified flow in the Hong Kong waters ([Postma et al., 1999](#)) and verified for the seasonal evolution of the thermocline ([Burchard and Baumert, 1995](#)).

9.5.4 Low Reynolds effect

In laboratory flumes or highly stratified flows e.g. by mud, the flow is nearly laminar. Then the vertical eddy viscosity computed by the standard k -epsilon model is over-estimated because this model lacks viscous damping terms. Therefore, in literature so-called low-Re $k-\varepsilon$ models are proposed. Our experience (Uittenbogaard, pers. comm.), however, is that these models require large resolution and switch from laminar to turbulent flow conditions at too large Re-numbers. Instead, we extended the $k-\varepsilon$ turbulence model in Delft3D-FLOW by a continuous damping function reducing its eddy viscosity, see below. In Delft3D-FLOW it is possible to switch on this so-called Low Reynolds turbulence model by adding the keyword `LRdamp` in the MDF-file. The formulation implemented is given below.

The turbulence Reynolds number is defined as

$$Re_T = \frac{k^2}{\nu_{mol}\varepsilon} = \frac{\nu_{3D}}{c_\mu\nu_{mol}}, \quad (9.140)$$

If the turbulence Reynolds number is low, then the eddy viscosity and the eddy diffusivity computed from the $k-\varepsilon$ model are corrected by the damping function

$$\nu_{3D}(Re_{low}) = f(Re_T)\nu_{3D}(k - \varepsilon; Re_{high}) \quad (9.141)$$

in which

$$f(Re_T) = 1 - e^{-p^2} \quad \text{where} \quad p = \frac{Re_T}{100} = \frac{\nu_{3D}}{100c_\mu\nu_{mol}} \quad (9.142)$$

No critical value for the Reynolds number needs to be specified, because a value of $Re_T = 300$ ($f = 0.999877$) is applied. We remark that function f rapidly converges to 1 for large values of Re_T .

9.6 Secondary flow; σ -model only

This feature is only available in the σ -model.

The flow in a river bend is basically three-dimensional. The velocity has a component in the plane perpendicular to the river axis. This component is directed to the inner bend near the riverbed and directed to the outer bend near the water surface, see [Figure 9.6](#).

This so-called 'secondary flow' (spiral motion) is of importance for the calculation of changes of the riverbed in morphological models and the dispersion of matter. In a 3D model the

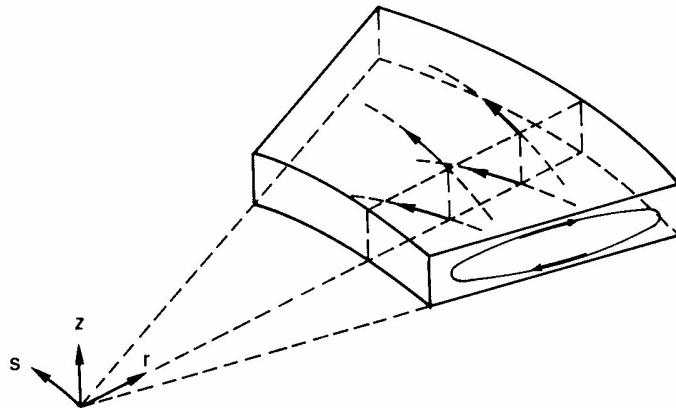


Figure 9.6: Spiral motion in river bend (from [Van Rijn \(1990\)](#))

secondary flow is resolved on the vertical grid, but in 2D depth-averaged simulations the secondary flow has to be determined indirectly using a secondary flow model. It strongly varies over the vertical but its magnitude is small compared to the characteristic horizontal flow velocity.

The secondary flow will be defined here as the velocity component $v(\sigma)$ normal to the depth-averaged main flow. The spiral motion intensity of the secondary flow I is a measure for the magnitude of this velocity component along the vertical:

$$I = \int_{-1}^0 |v(\sigma)| d\sigma. \quad (9.143)$$

The vertical distribution of the secondary flow is assumed to be a universal function of the vertical co-ordinate $f(\sigma)$. The actual local velocity distribution originates from a multiplication of this universal function with the spiral motion intensity, see [Kalkwijk and Booij \(1986\)](#):

$$v(\sigma) = f(\sigma) I. \quad (9.144)$$

A vertical distribution for a river bend is given in [Figure 9.7](#). The spiral motion intensity I leads to a deviation of the direction of the bed shear stress from the direction of the depth-averaged flow and thus affects the bedload transport direction. This effect can be taken into account in morphological simulations.

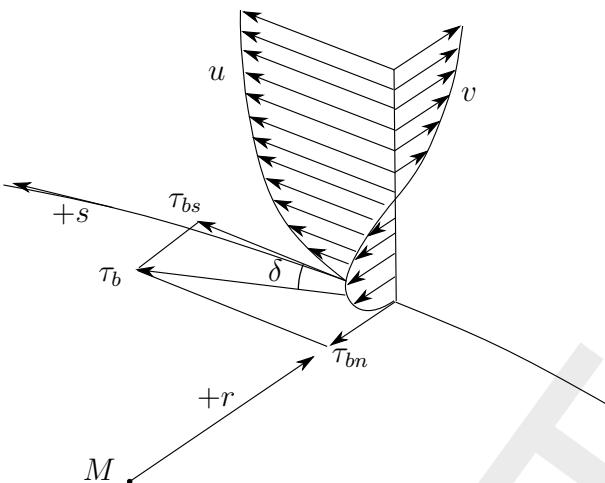


Figure 9.7: Vertical profile secondary flow (V) in river bend and direction bed stress

The component of the bed shear stress normal to the depth-averaged flow direction τ_{br} reads:

$$\tau_{br} = -2\rho\alpha^2(1-\alpha)\left|\vec{U}\right|I, \quad (9.145)$$

where α is defined in [Equation 9.156](#) and $\left|\vec{U}\right|$ is the magnitude of the depth-averaged velocity. To take into account the effect of the secondary flow on the depth-averaged flow, the depth-averaged shallow water equations have to be extended with:

- ◊ An additional advection-diffusion equation to account for the generation and adaptation of the spiral motion intensity.
- ◊ Additional terms in the momentum equations to account for the horizontal effective shear-stresses originating from the secondary flow.

Depth-averaged continuity equation

The depth-averaged continuity equation is given by:

$$\frac{\partial\zeta}{\partial t} + \frac{1}{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}}\frac{\partial[(d+\zeta)U\sqrt{G_{\eta\eta}}]}{\partial\xi} + \frac{1}{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}}\frac{\partial[(d+\zeta)V\sqrt{G_{\xi\xi}}]}{\partial\eta} = Q \quad (9.146)$$

where U and V indicate the depth-averaged velocities on an orthogonal curvilinear grid.

Momentum equations in horizontal direction

The momentum equations in ξ - and η -direction are given by:

$$\begin{aligned} \frac{\partial U}{\partial t} + \frac{U}{\sqrt{G_{\xi\xi}}}\frac{\partial U}{\partial\xi} + \frac{V}{\sqrt{G_{\eta\eta}}}\frac{\partial U}{\partial\eta} + \frac{UV}{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}}\frac{\partial\sqrt{G_{\xi\xi}}}{\partial\eta} - \frac{V^2}{\sqrt{G_{\xi\xi}}\sqrt{G_{\eta\eta}}}\frac{\partial\sqrt{G_{\eta\eta}}}{\partial\xi} + \\ - fV = -\frac{1}{\rho_0\sqrt{G_{\xi\xi}}}P_\xi - \frac{gU\sqrt{U^2+V^2}}{C_{2D}^2(d+\zeta)} + F_\xi + F_{s\xi} + M_\xi, \end{aligned} \quad (9.147)$$

and

$$\begin{aligned} \frac{\partial V}{\partial t} + \frac{U}{\sqrt{G_{\xi\xi}}} \frac{\partial V}{\partial \xi} + \frac{V}{\sqrt{G_{\eta\eta}}} \frac{\partial V}{\partial \eta} + \frac{UV}{\sqrt{G_{\xi\xi}} \sqrt{G_{\eta\eta}}} \frac{\partial \sqrt{G_{\eta\eta}}}{\partial \xi} - \frac{U^2}{\sqrt{G_{\xi\xi}} \sqrt{G_{\eta\eta}}} \frac{\partial \sqrt{G_{\xi\xi}}}{\partial \eta} + \\ + fU = -\frac{1}{\rho_0 \sqrt{G_{\eta\eta}}} P_\eta - \frac{gV \sqrt{U^2 + V^2}}{C_{2D}^2 (d + \zeta)} + F_\eta + F_{s\eta} + M_\eta. \quad (9.148) \end{aligned}$$

The fourth term in the right-hand side represents the effect of the secondary flow on the depth-averaged velocities (shear stresses by depth-averaging the non-linear acceleration terms).

Effect of secondary flow on depth-averaged momentum equations

To account for the effect of the secondary flow on the depth-averaged flow, the momentum equations have to be extended with additional shear stresses. To close the equations these stresses are coupled to parameters of the depth-averaged flow field. The main flow is assumed to have a logarithmic velocity profile and the secondary flow originates from a multiplication of a universal function with the spiral motion intensity, see [Kalkwijk and Booij \(1986\)](#). Depth averaging of the 3D equations leads to correction terms in the depth-averaged momentum equations for the effect of spiral motion:

$$\begin{aligned} F_{s\xi} = \frac{1}{d + \zeta} \left\{ \frac{1}{\sqrt{G_{\xi\xi}}} \frac{\partial [(d + \zeta) T_{\xi\xi}]}{\partial \xi} + \frac{1}{\sqrt{G_{\eta\eta}}} \frac{\partial [(d + \zeta) T_{\xi\eta}]}{\partial \eta} \right\} + \\ + \left\{ \frac{2T_{\xi\eta}}{\sqrt{G_{\xi\xi}} \sqrt{G_{\eta\eta}}} \frac{\partial \sqrt{G_{\xi\xi}}}{\partial \eta} + \frac{2T_{\xi\xi}}{\sqrt{G_{\xi\xi}} \sqrt{G_{\eta\eta}}} \frac{\partial \sqrt{G_{\eta\eta}}}{\partial \xi} \right\}, \quad (9.149) \end{aligned}$$

and:

$$\begin{aligned} F_{s\eta} = \frac{1}{d + \zeta} \left\{ \frac{1}{\sqrt{G_{\xi\xi}}} \frac{\partial [(d + \zeta) T_{\eta\xi}]}{\partial \xi} + \frac{1}{\sqrt{G_{\eta\eta}}} \frac{\partial [(d + \zeta) T_{\eta\eta}]}{\partial \eta} \right\} + \\ + \left\{ \frac{2T_{\eta\eta}}{\sqrt{G_{\xi\xi}} \sqrt{G_{\eta\eta}}} \frac{\partial \sqrt{G_{\xi\xi}}}{\partial \eta} + \frac{2T_{\eta\xi}}{\sqrt{G_{\xi\xi}} \sqrt{G_{\eta\eta}}} \frac{\partial \sqrt{G_{\eta\eta}}}{\partial \xi} \right\}, \quad (9.150) \end{aligned}$$

with the shear-stresses, resulting from the secondary flow, modelled as:

$$T_{\xi\xi} = -2\beta UV, \quad (9.151)$$

$$T_{\xi\eta} = T_{\eta\xi} = \beta (U^2 - V^2), \quad (9.152)$$

$$T_{\eta\eta} = 2\beta UV, \quad (9.153)$$

and:

$$\beta = \beta^* \frac{(d + \zeta)}{R_s^*}, \quad (9.154)$$

$$\beta^* = \beta_c (5\alpha - 15.6\alpha^2 + 37.5\alpha^3), \quad (9.155)$$

$\beta_c \in [0, 1]$, correction coefficient specified by you,

$$\alpha = \frac{\sqrt{g}}{\kappa C_{2D}} < \frac{1}{2}, \quad (9.156)$$

with R_s^* the effective radius of curvature of a 2D streamline to be derived from the intensity of the spiral motion and κ the Von Kármán constant. The spiral motion intensity is computed by [Equation 9.157](#). The limitation on α , [Equation 9.156](#), is set to ensure that the length scale L_a in [Equation 9.164](#) is always positive. For $\beta_c = 0$, the depth-averaged flow is not influenced by the secondary flow.



Remark:

- ◊ [Equation 9.156](#) effectively means a lower limit on C_{2D} .

The depth averaged transport equation for the spiral motion intensity

The variation of the spiral motion intensity I in space and time, is described by a depth-averaged advection-diffusion equation:

$$\frac{\partial [(d + \zeta) I]}{\partial t} + \frac{1}{\sqrt{G_{\xi\xi}} \sqrt{G_{\eta\eta}}} \left\{ \frac{\partial [(d + \zeta) U \sqrt{G_{\eta\eta}} I]}{\partial \xi} + \frac{\partial [(d + \zeta) V \sqrt{G_{\xi\xi}} I]}{\partial \eta} \right\} = \\ + \frac{d + \zeta}{\sqrt{G_{\xi\xi}} \sqrt{G_{\eta\eta}}} \left\{ \frac{\partial}{\partial \xi} \left[D_H \frac{\sqrt{G_{\eta\eta}}}{\sqrt{G_{\xi\xi}}} \frac{\partial I}{\partial \xi} \right] + \frac{\partial}{\partial \eta} \left[D_H \frac{\sqrt{G_{\xi\xi}}}{\sqrt{G_{\eta\eta}}} \frac{\partial I}{\partial \eta} \right] \right\} + (d + \zeta) S, \quad (9.157)$$

with:

$$S = -\frac{I - I_e}{T_a}, \quad (9.158)$$

$$I_e = I_{be} - I_{ce}, \quad (9.159)$$

$$I_{be} = \frac{d + \zeta}{R_s} |\vec{U}|, \quad (9.160)$$

$$I_{ce} = f \frac{d + \zeta}{2}, \quad (9.161)$$

$$|\vec{U}| = \sqrt{U^2 + V^2}, \quad (9.162)$$

$$T_a = \frac{L_a}{|\vec{U}|}, \quad (9.163)$$

$$L_a = \frac{(1 - 2\alpha)(d + \zeta)}{2\kappa^2\alpha}, \quad (9.164)$$

and R_s the radius of curvature of the stream-line defined by:

$$\frac{U_s}{R_s} = -\frac{\partial U_r}{\partial s}, \quad (9.165)$$

with U_s and U_r the components along and perpendicular to the streamline. The effective radius of curvature to be used for the evaluation of the coefficient β , [Equation 9.155](#), reads:

$$R_s^* = \frac{(d + \zeta) |\vec{U}|}{I}. \quad (9.166)$$

To guarantee stability the effective radius of curvature is bounded by the following empirical relation:

$$R_s^* \geq 10(d + \zeta). \quad (9.167)$$

The above formulas account for two sources of secondary flow:

- ◊ The centrifugal force in case of curved streamlines, I_{be} .
- ◊ The effect of the Coriolis force, I_{ce} .

The solution of the advection-diffusion equation will take some computer time. For steady state simulations, Delft3D-FLOW has an option to compute the spiral intensity I based on an algebraic expression assuming local equilibrium:

$$I = I_{be} - I_{ce}. \quad (9.168)$$

Boundary conditions for spiral motion

At open boundaries, during inflow the spiral motion intensity must be prescribed. The local equilibrium value [Equation 9.168](#) is used. For perpendicular inflow the radius of curvature goes to infinity and the spiral intensity reduces to zero.

9.7 Wave-current interaction

In relatively shallow areas (coastal seas) wave action becomes important because of several processes:

- ◊ The vertical mixing processes are enhanced due to turbulence generated near the surface by whitecapping and wave breaking, and near the bottom due to energy dissipation in the bottom layer.
- ◊ A net mass flux is generated which has some effect on the current profile, especially in cross-shore direction.
- ◊ In the surf zone long-shore currents and a cross-shore set-up is generated due to variations in the wave-induced momentum flux (radiation stress). In case of an irregular surf zone, bathymetry strong circulations may be generated (rip currents).
- ◊ The bed shear stress is enhanced; this affects the stirring up of sediments and increases the bed friction.

These processes are accounted for in a wave-averaged manner. Some processes basically act at a specific location or interface, such as the enhanced bed shear-stress or wave breaking at the surface, while others have a (certain) distribution over the vertical, such as the energy dissipation due to bottom friction in the wave boundary layer. A vertical distribution can, of course, only be accounted for in a 3D computation; in a 2D computation such a process is accounted for in a depth averaged form.

The computation of waves and wave-induced effects is the domain of wave models. Delft3D-WAVE supports currently one wave model: SWAN, a third generation wave model ([Ris, 1997](#)). SWAN computes the full (directional and frequency) spectrum on a rectilinear or curvilinear grid with many processes formulated in detail. For details and background you are referred to the Delft3D-WAVE User Manual ([WAVE UM, 2013](#)) and the references cited above.

For the sake of simplicity, we will describe in this section all wave-induced effects on a rectangular grid. In Delft3D-FLOW, however, these terms have been implemented in a curvilinear co-ordinate system. For a full wave-current interaction the currents from Delft3D-FLOW are used in Delft3D-WAVE (current refraction). The numerical grids used in the flow and wave computation often differ. Delft3D-FLOW and SWAN can both apply a rectilinear or curvilinear grid. To use the required grid resolution in the area of interest you can use nested grids. The transformation of results between the nested grids and the grids used in the flow and in the wave computations is executed automatically and is fully transparent to you. In SWAN, you

can use the same curvilinear grid as used in Delft3D-FLOW and obtain in one computation the required resolution in the area of interest.

The general procedure to derive the wave-induced forces is:

- ◊ Average the continuity equation and the momentum equations over the wave period.
- ◊ Express the residual terms that remain compared to the case without waves in terms of the wave properties, such as the wave energy, the phase and group velocity, and wave length or wave period.

Obviously, there are various averaging procedures. A straightforward approach is to define the mean motion by time averaging the equations over a wave period. The momentum equation in x -direction, averaged over the wave motion and expressed in Cartesian co-ordinates is given by:

$$\frac{\partial \bar{u}_j}{\partial t} + \bar{u}_i \frac{\partial \bar{u}_j}{\partial x_i} + \dots + g \frac{\partial \bar{\zeta}}{\partial x_j} - \frac{1}{\rho} \frac{\partial \bar{\tau}_{ij}}{\partial x_i} = F_j, \quad (9.169)$$

where for i and j the summation rule applies, $i, j = \{1, 2, 3\}$, the wave averaged quantity is indicated by an overhead bar, \bar{u}_j is the wave-mean velocity component, $\bar{\zeta}$ is the wave-mean free surface elevation, $\bar{\tau}_{ij}$ are the components of the wave-averaged normal stress tensor, and F_j is the wave-induced force that remains after averaging the momentum equation over the wave period. This wave induced force can generally be written as the gradient of the radiation-stress tensor S :

$$F_i = - \frac{\partial S_{ij}}{\partial x_j}. \quad (9.170)$$

Using wave propagation models we can express S_{ij} in terms of the wave parameters, such as the wave energy, the phase and group velocity, the wave length and the wave period. The wave-induced force is computed as the gradient of the radiation stress terms.

The above given procedure can only be applied when the mean motion is uniform with depth. In most practical cases this does not apply. Then the only way to split the mean and oscillating motion is through the *Generalised Lagrangian Mean, GLM* method of [Andrews and McIntyre \(1978\)](#). As shown by [Andrews and McIntyre \(1978\)](#), [Groeneweg and Klopman \(1998\)](#) and [Groeneweg \(1999\)](#) the (depth averaged and 3D) flow equations written in a co-ordinate system moving with the Stokes drift are very similar to the ordinary Eulerian formulation. However, the wave-induced driving force due to averaging over the wave motion is more accurately expressed in wave properties. The relation between the GLM velocity and the Eulerian velocity is given by:

$$\vec{u}^L = \vec{u}^E + \vec{u}^S, \quad (9.171)$$

where \vec{u}^L is the GLM-velocity vector, \vec{u}^E is the ordinary Eulerian-velocity vector and \vec{u}^S is the Stokes-drift vector.

The momentum equation in x -direction, averaged over the wave motion and expressed in GLM co-ordinates is given by:

$$\frac{\partial \bar{u}_j^L}{\partial t} + \bar{u}_i^L \frac{\partial \bar{u}_j^L}{\partial x_i} + \dots + g \frac{\partial \bar{\zeta}}{\partial x_j} - \frac{1}{\rho} \frac{\partial \bar{\tau}_{ij}^L}{\partial x_i} = F_j^L, \quad (9.172)$$

where for i and j the summation rule applies, $i, j = \{1, 2, 3\}$, and the quantities have the same meaning as in [Equation 9.169](#), but now in GLM co-ordinates. As shown by [Groeneweg](#)

(1999) the right-hand side of [Equation 9.172](#) contains a term related to a Stokes correction of the shear stresses. In the current implementation this term is neglected.

So in short:

- ◊ In Delft3D-FLOW, the hydrodynamic equations are written and solved in GLM-formulation, including the wave-current interactions.
- ◊ The GLM velocities are written to the communication file and are used in all (sediment) transport computations. Quantities moving with the flow are transported by the total, i.e. the GLM, velocity and not the Eulerian velocity.
- ◊ The Eulerian velocities are written to the Delft3D-FLOW result files to be used in comparisons of computational results with measurements.
- ◊ The only difference in the computational procedure as compared to the ordinary Eulerian formulation occurs at the boundaries. The velocity at the bottom is not the total velocity but the Eulerian velocity; so the bed-shear-stress is corrected for the Stokes drift. At lateral velocity boundaries, the total velocity must be prescribed; the implementation is such that the Eulerian velocity is prescribed in the boundary conditions (by you) and the Stokes drift is added by Delft3D-FLOW. At water level boundaries, just the wave-mean water level must be prescribed.

In the following sections, we discuss the wave-current interaction terms and their possible profile in the vertical. Where applicable we distinguish between the depth-averaged and the 3D formulations.

Remarks:



- ◊ For brevity, we suppress the index L for GLM quantities. So, in this section a velocity without super-script refers to the GLM velocity, unless explicitly mentioned differently.
- ◊ A super-script S is used to indicate the Stokes drift.

9.7.1 Forcing by radiation stress gradients

The wave-induced force, i.e. the right-hand side of [Equation 9.172](#), can be expressed in the wave parameters of the wave model that is being applied. For linear current refraction the expression can be derived analytically. To account for wave dissipation due to for instance bottom friction, wave breaking and whitecapping and wave growth due to wind one can rely on mild slope formulations with dissipation terms.

As shown by [Dingemans et al. \(1987\)](#), using the gradients of the radiation stresses in numerical models can result in spurious currents. [Dingemans et al. \(1987\)](#) showed that the divergence free part of the radiation stress is not capable of driving currents and can therefore be neglected if one is primarily interested in wave-driven currents. The remaining part of the radiation stress gradients is closely related to the wave energy dissipation, i.e. the right-hand side of [Equation 9.172](#) can be written as:

$$F_i = \frac{Dk_i}{\omega}, \quad (9.173)$$

where D is the total energy dissipation due to waves, k_i is the wave number in i -direction and ω is the wave frequency; see [Dingemans \(1997\)](#) for many details and discussions on this subject.

2D implementation

For a depth averaged model the momentum equations in x - and y -direction, leaving out most of the terms, can be written as:

$$\frac{\partial U}{\partial t} + \dots + \frac{gU\sqrt{U^2 + V^2}}{C_{2D}^2(d + \zeta)} + \dots = \dots + F_x, \quad (9.174)$$

$$\frac{\partial V}{\partial t} + \dots + \frac{gV\sqrt{U^2 + V^2}}{C_{2D}^2(d + \zeta)} + \dots = \dots + F_y, \quad (9.175)$$

where F_x and F_y are the depth averaged wave-induced forcings and given by the gradients of the radiation stress tensor S , or following [Dingemans et al. \(1987\)](#) approximated by wave energy dissipation:

$$F_x = -\frac{\partial S_{xx}}{\partial x} - \frac{\partial S_{yx}}{\partial y} = D \frac{k_x}{\omega}, \quad (9.176)$$

$$F_y = -\frac{\partial S_{xy}}{\partial x} - \frac{\partial S_{yy}}{\partial y} = D \frac{k_y}{\omega}. \quad (9.177)$$

The dissipation rate D (a negative quantity) is computed by the wave model and read from the communication file. In SWAN, the dissipation rate may be computed from the bottom friction (orbital motion), depth-induced breaking and whitecapping.

You can choose to apply the radiation stress or the dissipation rate to determine the wave-induced forces.

3D implementation

In version 5.01 (and higher) of Delft3D-FLOW the 3 different dissipation rates, calculated by SWAN, are handled separately: dissipation due to depth-induced breaking and whitecapping at the top layer and dissipation due to bottom friction at the bed layer. The effect of the divergence free part of the radiation stresses is (the remaining part of the radiation stress gradients, after the wave dissipation has been subtracted) is added to the momentum equations in Delft3D-FLOW (effect spread over the water column). This provides most adequate results, e.g. in situations with undertow.

9.7.2 Stokes drift and mass flux

In surface waves, fluid particles describe an orbital motion. The net horizontal displacement for a fluid particle is not zero. This wave induced drift velocity, the so-called Stokes-drift, is always in the direction of wave propagation. A particle at the top of the orbit beneath a wave crest moves slightly faster in the forward direction than it does in the backward direction beneath a wave trough. The mean drift velocity is a second order quantity in the wave height. The drift leads to additional fluxes in the wave averaged mass continuity equation.

The wave-induced mass fluxes M_x^S and M_y^S are found by integration of the components of the Stokes drift u^S and v^S over the wave-averaged total water depth:

$$M_x^S = \int_{-d}^{\bar{\zeta}} \rho_0 u^S dz = \frac{E}{\omega} k_x \quad (9.178)$$

$$M_y^S = \int_{-d}^{\bar{\zeta}} \rho_0 v^S dz = \frac{E}{\omega} k_y \quad (9.179)$$

with E the wave energy defined as:

$$E = \frac{1}{8} \rho_0 g H_{rms}^2. \quad (9.180)$$

The mass fluxes M_x^S and M_y^S are computed by an interface program and are written to the communication file.

Remarks:



- ◊ The mass flux effect is only taken into account when Delft3D-FLOW is used from within Delft3D-MOR.
- ◊ The velocities written to the communication file for use in Delft3D-MOR, Delft3D-WAVE, and D-Water Quality are based on the *total flux* velocities.
- ◊ The Eulerian velocities, which may be used in comparisons with measurements at a fixed location, are written to the hydrodynamic map and history files.

2D implementation

The depth-averaged Stokes drift is given by:

$$U^S = \frac{M_x^S}{\rho_0 (d + \zeta)}, \quad (9.181)$$

$$V^S = \frac{M_y^S}{\rho_0 (d + \zeta)}. \quad (9.182)$$

3D implementation

The Stokes drift is computed from the wave theory, see [Dean and Dalrymple \(1991, pg. 287, eq. 10.8\)](#):

$$\vec{u}^S(z) = \frac{\omega k a^2 \cosh(2kz)}{2 \sinh^2(kH)} (\cos \phi, \sin \phi)^T. \quad (9.183)$$

The wave is computed from the mass fluxes which are read from the communication file:

$$\phi = \tan^{-1} (M_x^S / M_y^S). \quad (9.184)$$

9.7.3 Streaming

Streaming (a wave-induced current in the wave boundary layer directed in the wave propagation direction) is modelled as a time-averaged shear-stress which results from the fact that the horizontal and vertical orbital velocities are not exactly 90° out of phase. It is based on the wave bottom dissipation and is assumed to decrease linearly to zero across the wave boundary layer ([Fredsøe and Deigaard, 1992](#)):

$$-\frac{\partial \bar{u} \bar{w}}{\partial z} = \frac{D_f k \cos \phi}{\rho_0 \omega \delta} \left(1 - \frac{d + \zeta - z'}{\delta} \right) \quad \text{for } d + \zeta - \delta \leq z' \leq d + \zeta \quad (9.185)$$

For the definition of z' see [Figure 9.8](#).

The left-hand side of [Equation 9.185](#) is the residual term after averaging a vertical derivative of the advection terms over the wave period. The dissipation due to bottom friction D_f is

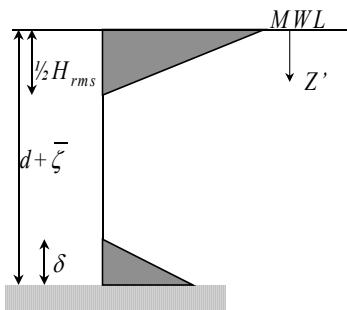


Figure 9.8: Vertical distribution of turbulent kinetic energy production

given by:

$$D_f = \frac{1}{2\sqrt{\pi}} \rho_0 f_w u_{orb}^3, \quad (9.186)$$

where u_{orb} is the orbital velocity near the bed, given by [Equation 9.204](#). The friction factor f_w is according to [Soulsby et al. \(1993b\)](#) given by:

$$f_w = \min \left\{ 0.3, 1.39 \left(\frac{A}{z_0} \right)^{-0.52} \right\}, \quad (9.187)$$

with:

$$A = \frac{u_{orb}}{\omega}, \quad (9.188)$$

and ϕ is given by [Equation 9.184](#).

The thickness of the wave boundary layer δ is given by:

$$\delta = H \min \left[0.5, 20 \max \left\{ \frac{ez_0}{H}, 0.09 \frac{k_s}{H} \left(\frac{A}{k_s} \right)^{0.82} \right\} \right]. \quad (9.189)$$

9.7.4 Wave induced turbulence

The vertical mixing processes are enhanced by the wave actions. This can best be accounted for by adding the wave energy production and dissipation terms in the turbulence model. The two main sources of wave energy decay that are included are wave breaking and bottom friction due to the oscillatory wave motion in the bottom boundary layer.

In the case of breaking waves, there is a production of turbulent energy directly associated with the energy dissipation due to breaking ([Deigaard, 1986](#)). The total depth-averaged contribution of wave breaking is given by D_w [W/m²], which is computed in the wave model (e.g. SWAN). Wave energy dissipation due to bottom friction is also considered to produce turbulent kinetic energy. These processes are incorporated by introducing source terms in the turbulent kinetic energy (for both the $k-L$ and the $k-\varepsilon$ turbulence model) and in the turbulent kinetic energy dissipation equation ($k-\varepsilon$ turbulence model).

Breaking waves

In a 3D-model the contribution due to wave breaking is linearly distributed over a half wave height below the mean water surface, $H_{rms}/2$. This is described by the following expression for the turbulent kinetic energy distribution:

$$P_{kw}(z') = \frac{4D_w}{\rho_w H_{rms}} \left(1 - \frac{2z'}{H_{rms}} \right) \quad \text{for } 0 \leq z' \leq \frac{1}{2}H_{rms}. \quad (9.190)$$

The source term $P_{\varepsilon w}$ in the ε -equation is coupled to P_{kw} according to:

$$P_{\varepsilon w}(z') = c_{1\varepsilon} \frac{\varepsilon}{k} P_{kw}(z'), \quad (9.191)$$

where $c_{1\varepsilon}$ is a calibration constant ($c_{1\varepsilon} = 1.44$).

Due to the breaking of waves the boundary conditions of k and ε have to be changed. The boundary conditions at the free surface of k and ε are given by:

$$k|_{\text{free surface}} = k_{\text{wind}} + k_{\text{waves}} = k_{\text{wind}} + \left(\frac{2D_w \kappa}{\rho_w c_D} \right)^{\frac{2}{3}} \quad (9.192)$$

$$\varepsilon|_{\text{free surface}} = \varepsilon_{\text{wind}} + \varepsilon_{\text{waves}} = \varepsilon_{\text{wind}} + \frac{4D_w}{\rho_w H_{rms}} \quad (9.193)$$

Bottom friction

The contribution due to bottom friction is linearly distributed over the thickness of the wave boundary layer, see [Figure 9.8](#).

$$P_{kw}(z') = \frac{2D_f}{\rho_0 \delta} \left(1 - \frac{d + \zeta - z'}{\delta} \right) \quad \text{for } d + \zeta - \delta \leq z' \leq d + \zeta, \quad (9.194)$$

due to wave energy decay in the bottom boundary layer. Here z' is the vertical co-ordinate with its origin at the (wave averaged) water level and so z' is positive downward, D_f represent wave energy dissipation due to bottom friction, D_f is given by [Equation 9.186](#).

The boundary conditions at the bottom, due to the non-linear interaction of the current and wave boundary layer is directly taken into account, by the adaptation of the bottom roughness height, see [Equation 9.212](#).

9.7.5 Enhancement of the bed shear-stress by waves

The boundary layers at the bed associated with the waves and the current interact non-linearly. This has the effect of enhancing both the mean and oscillatory bed shear-stresses. In addition the current profile is modified, because the extra turbulence generated close to the bed by the waves appears to the current as being equivalent to an enhanced bottom roughness. The bed shear-stress due to the combination of waves and current is enhanced beyond the value which would result from a linear addition of the bed shear-stress due to waves, $\vec{\tau}_w$, and the bed shear-stress due to current $\vec{\tau}_c$. For sediment transport modelling it is important to predict the maximum bed shear-stress, $\vec{\tau}_{\max}$, while the current velocity and the turbulent diffusion are determined by the combined wave-current bed shear-stress $\vec{\tau}_m$.

Various, often very complex, methods exist to describe the bottom boundary layer under combined current and wave action and the resulting virtual roughness. [Soulsby et al. \(1993a\)](#)

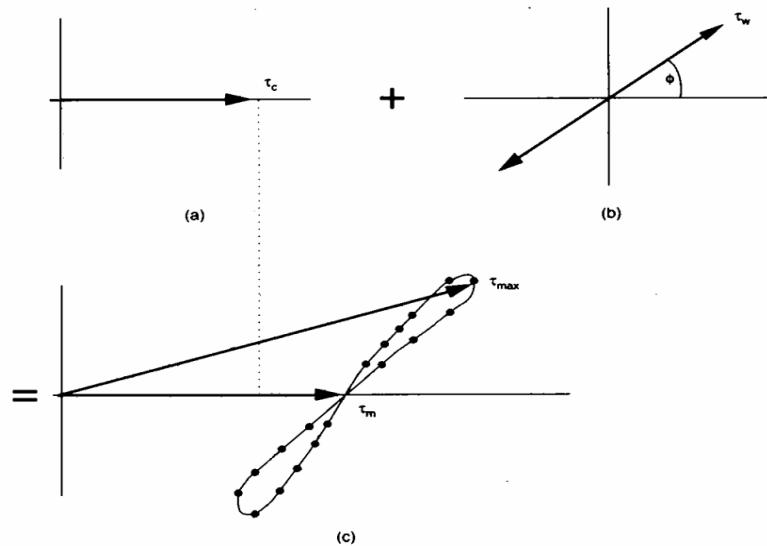


Figure 9.9: Schematic view of non-linear interaction of wave and current bed shear-stresses (from [Soulsby et al. \(1993b\)](#), Figure 16, p. 89))

developed a parameterisation of these methods allowing a simple implementation and comparison of various wave-current interaction models: [Fredsøe \(1984\)](#); [Myrhaug and Slaattelid \(1990\)](#); [Grant and Madsen \(1979\)](#); [Huynh-Thanh and Temperville \(1991\)](#); [Davies et al. \(1988\)](#); [Bijker \(1967\)](#); [Christoffersen and Jonsson \(1985\)](#); [O'Connor and Yoo \(1988\)](#); [Van Rijn et al. \(2004\)](#). All these methods have all been implemented in Delft3D-FLOW and can be applied in 2D and 3D modelling. However, as there are minor, but specific differences in determining certain quantities, such as determining the shear-stress at the bottom, we prefer to discuss the 2D and 3D implementation separately.

2D implementation

Following [Soulsby et al. \(1993b\)](#), Figure 9.9 gives a schematic overview of the bed shear-stresses for wave current interaction.

[Soulsby et al. \(1993b\)](#) fitted one standard formula to all of the models, each model having its own fitting coefficients. The parameterisation of Soulsby for the time-mean bed shear-stress is of the form:

$$|\vec{\tau}_m| = Y (|\vec{\tau}_c| + |\vec{\tau}_w|), \quad (9.195)$$

with

$$Y = X \{1 + bX^p(1 - X)^q\}, \quad (9.196)$$

and for the maximum bed shear-stress:

$$|\vec{\tau}_{\max}| = Z (|\vec{\tau}_c| + |\vec{\tau}_w|), \quad (9.197)$$

with

$$Z = 1 + aX^m(1 - X)^n. \quad (9.198)$$

and:

$$X = \frac{|\vec{\tau}_c|}{|\vec{\tau}_c| + |\vec{\tau}_w|}, \quad (9.199)$$

The value of the parameters a , b , p , q , m and n depends on the friction model which is parameterised, and:

$ \vec{\tau}_c $	magnitude of the bed stress due to current alone
$ \vec{\tau}_w $	magnitude of the bed stress for waves alone
$ \vec{\tau}_m $	magnitude of the mean bed stress for combined waves and current
$ \vec{\tau}_{\max} $	magnitude of the maximum bed stress for combined waves and current.

Remark:

- ◊ The stresses $\vec{\tau}_m$ and $\vec{\tau}_{\max}$ are assumed to have the same direction as $\vec{\tau}_c$.



Following [Soulsby et al. \(1993b\)](#) the expressions for the parameters χ ($= a, b, p, q, m, n$) and \mathcal{J} ($= I, J$; also depending on the friction model) have the form:

$$\chi = \left(\chi_1 + \chi_2 |\cos \phi|^{\mathcal{J}} \right) + \left(\chi_3 + \chi_4 |\cos \phi|^{\mathcal{J}} \right) {}^{10} \log \left(\frac{f_w}{C_{2D}} \right), \quad (9.200)$$

in which:

C_{2D}	drag coefficient due to current
f_w	wave friction factor
ϕ	the angle between the current direction and the direction of wave propagation.

As the radiation stress is always in the wave direction, we can derive ϕ from:

$$|\cos \phi| = \frac{|UF_x + VF_y|}{|\vec{U}| |\vec{F}|}. \quad (9.201)$$

Values of the parameters a, b, p, q and J in [Equation 9.200](#) have been optimised by [Soulsby et al. \(1993b\)](#), see [Table 9.5](#) and [Figure 9.10](#).

The bed shear-stress due to flow alone may be computed using various types of formulations like Chézy, Manning or White-Colebrook, see Eqs. [9.54](#) to [9.56](#). The bed shear-stress due to current alone can be written in the form:

$$\vec{\tau}_c = \frac{g \rho_0 \vec{U} |\vec{U}|}{C_{2D}^2}. \quad (9.202)$$

The magnitude of the wave-averaged bed shear-stress due to waves alone is related to the wave orbital velocity near the bottom \vec{u}_{orb} and the friction coefficient f_w :

$$|\vec{\tau}_w| = \frac{1}{2} \rho_0 f_w u_{orb}^2. \quad (9.203)$$

The orbital velocity is computed from the linear wave theory and is given by:

$$u_{orb} = \frac{1}{4} \sqrt{\pi} \frac{H_{rms} \omega}{\sinh(kH)}, \quad (9.204)$$

where the root-mean-square wave height H_{rms} and the wave period $T (= 2\pi/\omega)$ are read from the communication file. The variation of the wave friction factor with relative orbital excursion at the bed under purely oscillatory flow is given by [Swart \(1974\)](#) (\equiv Equations [11.77](#),

Table 9.5: Fitting coefficients for wave/current boundary layer model

Model ¹	a_1	a_2	a_3	a_4	m_1	m_2	m_3	m_4	n_1	n_2	n_3	n_4	I
FR84	-0.06	1.70	-0.29	0.29	0.67	-0.29	0.09	0.42	0.75	-0.27	0.11	-0.02	0.80
MS90	-0.01	1.84	-0.58	-0.22	0.63	-0.09	0.23	-0.02	0.82	-0.30	0.19	-0.21	0.67
HT91	-0.07	1.87	-0.34	-0.12	0.72	-0.33	0.08	0.34	0.78	-0.23	0.12	-0.12	0.82
GM79	0.11	1.95	-0.49	-0.28	0.65	-0.22	0.15	0.06	0.71	-0.19	0.17	-0.15	0.67
DS88	0.05	1.62	-0.38	0.25	1.05	-0.75	-0.08	0.59	0.66	-0.25	0.19	-0.03	0.82
BK67	0.00	2.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.50	0.00	0.00	1.00
CJ85	-0.01	1.58	-0.52	0.09	0.65	-0.17	0.18	0.05	0.47	-0.03	0.59	-0.50	0.64
OY88	-0.45	2.24	0.16	-0.09	0.71	0.27	-0.15	0.03	1.19	-0.66	-0.13	0.12	0.77
	b_1	b_2	b_3	b_4	p_1	p_2	p_3	p_4	q_1	q_2	q_3	q_4	J
FR84	0.29	0.55	-0.10	-0.14	-0.77	0.10	0.27	0.14	0.91	0.25	0.50	0.45	3.00
MS90	0.65	0.29	-0.30	-0.21	-0.60	0.10	0.27	-0.06	1.19	-0.68	0.22	-0.21	0.50
HT91	0.27	0.51	-0.10	-0.24	-0.75	0.13	0.12	0.02	0.89	0.40	0.50	-0.28	2.70
GM79	0.73	0.40	-0.23	-0.24	-0.68	0.13	0.24	-0.07	1.04	-0.56	0.34	-0.27	0.50
DS88	0.22	0.73	-0.05	-0.35	-0.86	0.26	0.34	-0.07	-0.89	2.33	2.60	-2.50	2.70
BK67	0.32	0.55	0.00	0.00	-0.63	0.05	0.00	0.00	1.14	0.18	0.00	0.00	3.00
CJ85	0.47	0.29	-0.09	-0.12	-0.70	0.13	0.28	-0.04	1.65	-1.19	-0.42	0.49	0.60
OY88	-0.06	0.26	0.08	-0.03	-1.00	0.31	0.25	-0.26	0.38	1.19	0.25	-0.66	1.50
VR04	$Y = 0.0$ and $Z = 1.0$												

¹ FR84=Fredsøe (1984), MS90=Myrhaug and Slaattelid (1990),
 HT91=Huynh-Thanh and Temperville (1991), GM79=Grant and Madsen (1979), DS88=Davies *et al.* (1988),
 BK67=Bijker (1967), CJ85=Christoffersen and Jonsson (1985), OY88=O' Connor and Yoo (1988),
 VR04=Van Rijn *et al.* (2004)

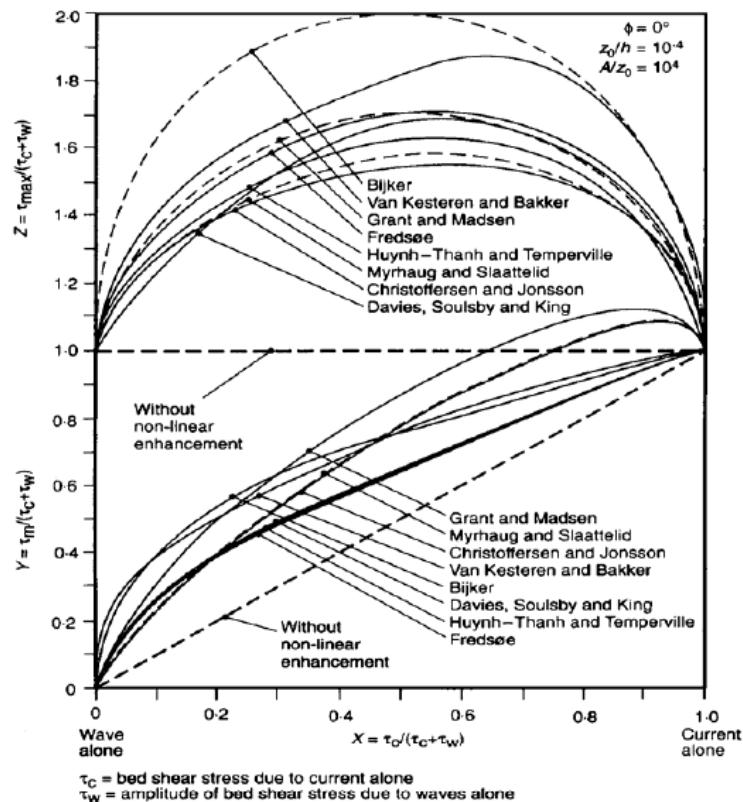


Figure 9.10: Inter-comparison of eight models for prediction of mean and maximum bed shear-stress due to waves and currents (from Soulsby *et al.* (1993b, Figure 17, p. 90))

[11.116](#) and [11.157](#)):

$$f_w = \begin{cases} 0.00251 \exp \left[5.21 \left(\frac{A}{k_s} \right)^{-0.19} \right], & \frac{A}{k_s} > \frac{\pi}{2}, \\ 0.3, & \frac{A}{k_s} \leq \frac{\pi}{2}, \end{cases} \quad (9.205)$$

with:

$$A = \frac{u_{orb}}{\omega}, \quad (9.206)$$

k_s is the Nikuradse roughness length-scale and ω is the wave angular frequency.

As the bed is in rest and the equations are formulated in GLM co-ordinates, we must correct the bed shear-stress used in the momentum equations for the Stokes drift. The total or effective bed shear stress is given by:

$$\vec{\tau}_b = \frac{|\vec{\tau}_m|}{|\vec{U}|} (\vec{U} - \vec{U}^S), \quad (9.207)$$

where the components of the depth-averaged Stokes drift \vec{U}^S are given by Eqs. [9.181](#) and [9.182](#).

3D implementation

For 3D simulations the enhanced bed shear-stress is computed along the same lines, but remark that the bed shear-stress is now a boundary condition. The magnitude of the depth-averaged velocity in [Equation 9.202](#) is determined from the velocity near the bed, assuming a logarithmic velocity profile.

The bed shear-stress corrected for the Stokes drift is given by:

$$\vec{\tau}_b = \frac{|\vec{\tau}_m|}{|\vec{U}_{2D}|} (\vec{u} - \vec{u}^S), \quad (9.208)$$

with:

$$\vec{U}_{2D} = \frac{1}{d + \zeta} \int_{-d}^{\zeta} \vec{u} dz. \quad (9.209)$$

The Stokes drift and the angle between near bed current and the waves are given by Eqs. [9.183](#) and [9.184](#).

Matching the velocity profile corresponding with the increased mean bed stress $\vec{\tau}_m$ for waves and current, with a logarithmic profile for the mean flow outside the wave boundary layer:

$$\vec{\tau}_m = \rho_0 \vec{\tilde{u}}_* |\vec{\tilde{u}}_*| \quad (9.210)$$

with $\vec{\tilde{u}}_*$ the shear-stress velocity for waves and current and following [Equation 9.58](#):

$$\vec{u}_b = \frac{\vec{\tilde{u}}_*}{\kappa} \ln \left(1 + \frac{\Delta z_b}{2z_0} \right), \quad (9.211)$$

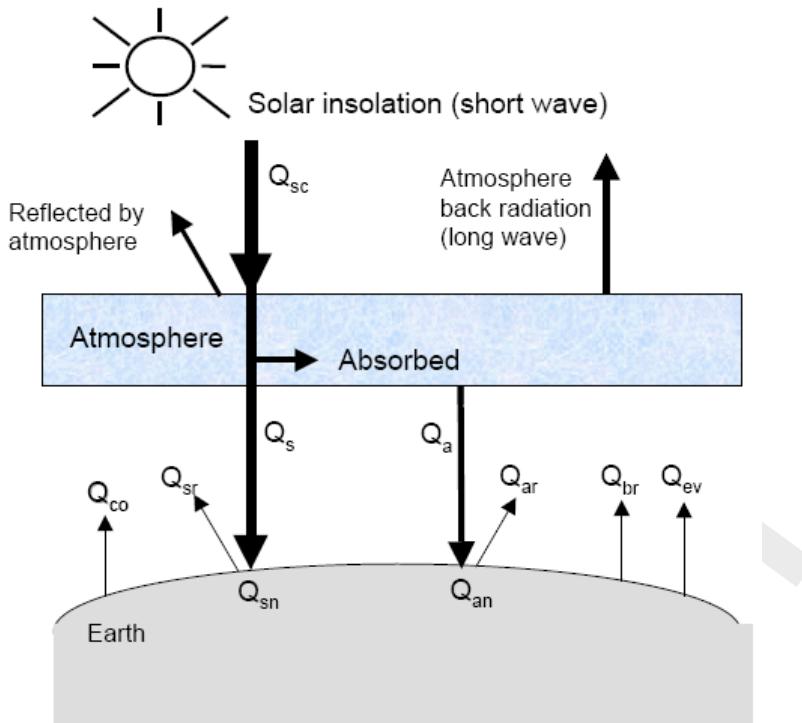


Figure 9.11: Overview of the heat exchange mechanisms at the surface

with \vec{u}_b the horizontal velocity in the first layer just above the bed. The increased roughness length for waves and current \tilde{z}_0 satisfies:

$$\tilde{z}_0 = \frac{\Delta z_b}{\exp\left(\kappa \frac{|\vec{u}_b|}{|\vec{u}_*|}\right) - 1}. \quad (9.212)$$

This increased roughness length is used in the turbulence closure schemes.

9.8 Heat flux models

The heat radiation emitted by the sun reaches the earth in the form of electromagnetic waves with wavelengths in the range of 0.15 to 4 μm . In the atmosphere the radiation undergoes scattering, reflection and absorption by air, cloud, dust and particles. On average neither the atmosphere nor the earth accumulates heat, which implies that the absorbed heat is emitted back again. The wavelengths of these emitted radiations are longer (between 4 and 50 μm) due to the lower prevailing temperature in the atmosphere and on Earth. Schematically the radiation process, along with the heat flux mechanisms at the water surface, is shown in Figure 9.11.

Legend for Figure 9.11:

Q_{sc}	radiation (flux) for clear sky condition in $[\text{J}/\text{m}^2\text{s}]$
Q_{co}	heat loss due to convection (sensible) in $[\text{J}/\text{m}^2\text{s}]$
Q_{sr}	reflected solar radiation in $[\text{J}/\text{m}^2\text{s}]$
Q_s	solar radiation (short wave radiation) in $[\text{J}/\text{m}^2\text{s}]$
Q_{sn}	net incident solar radiation (short wave), $= Q_s - Q_{sr}$
Q_a	atmospheric radiation (long wave radiation) in $[\text{J}/\text{m}^2\text{s}]$
Q_{an}	net incident atmospheric radiation (long wave)

Q_{ar}	reflected atmospheric radiation in [J/m ² s]
Q_{br}	back radiation (long wave radiation) in [J/m ² s]
Q_{ev}	heat loss due to evaporation (latent) in [J/m ² s]

In Delft3D-FLOW the heat exchange at the free surface is modelled by taking into account the separate effects of solar (short wave) and atmospheric (long wave) radiation, and heat loss due to back radiation, evaporation and convection. In literature there is a great variability of empirical formulations to calculate these heat fluxes across the sea surface. Most formulations differ in the dependency of the exchange on the meteorological parameters such as wind speed, cloudiness and humidity. Some formulations were calibrated for coastal seas others for lakes.

In Delft3D-FLOW five heat flux models have been implemented:

1 Heat flux model 1

The incoming (short wave) solar radiation for a clear sky is prescribed. The net atmospheric (long wave) radiation and the heat losses due to evaporation, back radiation and convection are computed by the model.

2 Heat flux model 2

The combined net (short wave) solar and net (long wave) atmospheric radiation is prescribed. The terms related to heat losses due to evaporation, back radiation and convection are computed by the model.

3 Excess temperature model

The heat exchange flux at the air-water interface is computed; only the background temperature is required.

4 Murakami heat flux model

The net (short wave) solar radiation is prescribed. The effective back radiation and the heat losses due to evaporation and convection are computed by the model. The incoming radiation is absorbed as a function of depth. The evaporative heat flux is calibrated for Japanese waters.

5 Ocean heat flux model

The fraction of the sky covered by clouds is prescribed (in %). The effective back radiation and the heat losses due to evaporation and convection are computed by the model. Additionally, when air and water densities and/or temperatures are such that free convection occurs, free convection of latent and sensible heat is computed by the model. This model formulation typically applies for large water bodies.

For the physical background of the heat exchange at the air-water interface and the definitions, we refer to [Octavio et al. \(1977\)](#) for model 1 and 2, to [Sweers \(1976\)](#) for model 3, to [Murakami et al. \(1985\)](#) for model 4 and to [Gill \(1982\)](#) and [Lane \(1989\)](#) for model 5.

With the exception of model 3 the excess temperature model, the heat fluxes through the water surface by incoming radiation, back radiation, evaporation and convection are computed separately. Evaporation and convection depend on the air temperature, the water temperature near the free surface, relative humidity, and wind speed. The excess temperature model computes the heat fluxes through the water surface in such a way that the temperature of the surface layer relaxates to the natural background temperature specified by you. The heat transfer coefficient mainly depends on the water temperature and the wind speed.

9.8.1 Heat balance

The total heat flux through the free surface reads:

$$Q_{tot} = Q_{sn} + Q_{an} - Q_{br} - Q_{ev} - Q_{co}, \quad (9.213)$$

with:

Q_{sn}	net incident solar radiation (short wave)
Q_{an}	net incident atmospheric radiation (long wave)
Q_{br}	back radiation (long wave)
Q_{ev}	evaporative heat flux (latent heat)
Q_{co}	convective heat flux (sensible heat).

The subscript n refers to a net contribution. Each of the heat fluxes in [Equation 9.213](#) will be discussed in detail.

The change in temperature in the top layer T_s (in $^{\circ}\text{C}$) is given by:

$$\frac{\partial T_s}{\partial t} = \frac{Q_{tot}}{\rho_w c_p \Delta z_s}, \quad (9.214)$$

where

Q_{tot}	is the total heat flux through the air-water surface, [$\text{J/m}^2\text{s}$],
c_p	is the specific heat capacity of sea water ($= 3930 \text{ J kg}^{-1} \text{ K}$),
ρ_w	is the specific density of water, [kg/m^3] and
Δz_s	is the thickness of the top layer, [m].

In Delft3D-FLOW, the heat exchange at the bed is assumed to be zero. This may lead to over-prediction of the water temperature in shallow areas. Also the effect of precipitation on the water temperature is not taken into account.



Remarks:

- ◊ The temperature T is by default expressed in $^{\circ}\text{C}$. However, in some formulas the absolute temperature \bar{T} in K is used. They are related by:

$$\bar{T} = T + 273.15. \quad (9.215)$$

- ◊ In [Equation 9.214](#) the total incoming heat flux is assumed to be absorbed in the top layer. This may result in an unrealistically high surface temperature when the top layer is thin. This can be prevented by absorbing the incoming radiation as a function of depth. Currently, this is only implemented in heat flux models 4 and 5.
- ◊ The input parameters for the heat flux models are model dependent. For example, the input for the solar radiation has to be specified (measured data) as incoming (short-wave) solar radiation for a clear sky (Q_{sc}) for heat flux model 1, as combined net solar radiation and net atmospheric radiation (sum of Q_{sn} and Q_{an}) for heat flux model 2, as net solar radiation for heat flux model 4, or is computed by the heat flux model itself (heat flux models 3 and 5).
- ◊ When using heat flux model 5, the free convection of latent and sensible heat is also determined. This can be switched off when needed, by setting the coefficient of free convection `cfrcon` to zero.

9.8.2 Solar radiation

The short-wave radiation emitted by the sun that reaches the earth surface under a clear sky condition can be evaluated by means of:

- ◊ Direct measurements.
- ◊ Applying Stefan-Boltzmann's law for radiation from a black-body:

$$Q = \sigma \bar{T}^4 \quad (9.216)$$

with $\sigma = \text{Stefan-Boltzmann's constant} = 5.67 \cdot 10^{-8} \text{ J/(m}^2\text{s K}^4\text{)}$ and \bar{T} the (absolute) temperature in K.

◊ Empirical formulae.

In the heat flux models 1, 2 and 4 the solar radiation has to be specified by the user. In the excess temperature model (model 3) the solar radiation does not play an explicit role, but is part of the background temperature. In the heat flux model 5 the solar radiation is computed by Delft3D-FLOW and is dependent on the geographical position at the earth and the local time.

Not all of the radiation is absorbed at the water surface. A part is transmitted to deeper water. Short waves can penetrate over a distance of 3 to 30 meters, depending on the clarity of the water, while the relatively longer waves are absorbed at the surface. Therefore, it is convenient to separate the incoming solar insolation into two portions:

- 1 βQ_{sn} , the longer wave portion, which is absorbed at the surface and
- 2 $(1 - \beta) Q_{sn}$, the remainder part, which is absorbed in deeper water.

The absorption of heat in the water column is an exponential function of the distance H from the water surface:

$$(1 - \beta) Q_{sn} = \int_0^H e^{-\gamma z} dz \Rightarrow \quad (9.217)$$

$$Q_{sn}(h) = \frac{\gamma e^{-\gamma h}}{1 - e^{-\gamma H}} (1 - \beta) Q_{sn}, \quad (9.218)$$

with:

β	part of Q_{sn} absorbed at the water surface which is a function of the wavelength. The default value of β in Delft3D-FLOW is 0.06.
γ	extinction coefficient (measured) in m^{-1} , also related to the so-called Secchi-depth $\gamma = \frac{1.7}{H_{\text{Secchi}}}$
h	distance to the water surface in meters.
H	total water depth.

Remark:

- ◊ The exponential decay function, [Equation 9.218](#) has only been implemented in the heat flux models 4 and 5. In the other heat flux models the incoming radiation is expected to be absorbed in the top layer.



In heat flux model 5 the flux is computed dependent on the geographical position and the local time. The incoming energy flux at the water surface depends on the angle (declination) between the incoming radiation and the Earth's surface. This declination depends on the geographical position on the Earth and the local time. The Earth axis is not perpendicular to the line connecting the Sun with Earth. This tilting (angle δ) varies with the time of the year and it leads to a seasonal variation of the radiation flux. At June 21, the declination is maximal, 23.5 degrees. Of course, by the rotation of the Earth the solar radiation also varies during the day. Near twelve o'clock local time, the sun elevation above the horizon is maximal. For an overview of the angles used to determine the solar elevation angle γ , see [Figure 9.12](#).

The temporal and latitude-dependent solar elevation angle γ is estimated by:

$$\sin(\gamma) = \sin(\delta) \sin\left(\frac{\pi\phi}{180}\right) - \cos(\delta) \cos\left(\frac{\pi\phi}{180}\right) \cos(\omega_1 t) \quad (9.219)$$

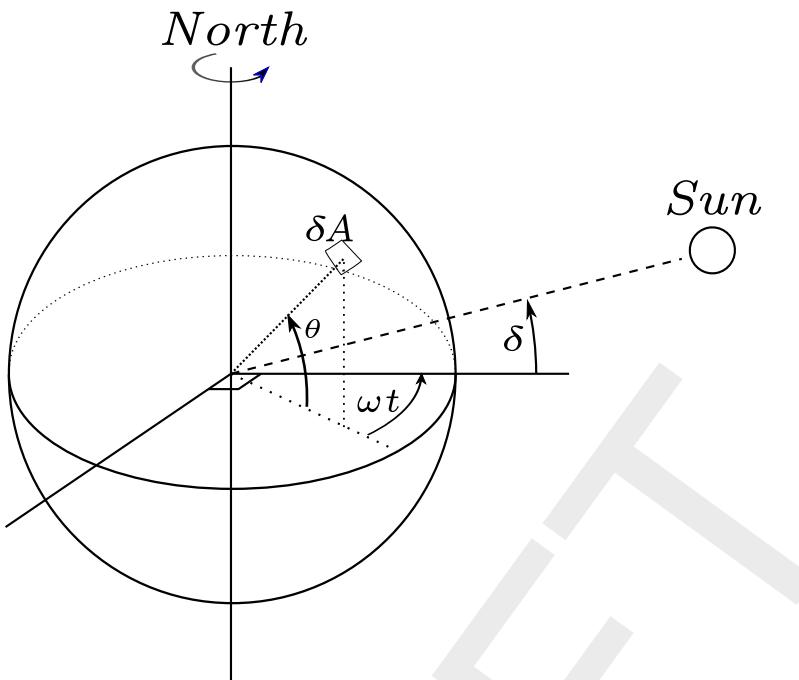


Figure 9.12: Co-ordinate system position Sun
 δ : declination; θ : latitude; ωt : angular speed

with:

$$\delta = \frac{23.5\pi}{180} \cos(\omega_0 t - 2.95), \quad (9.220)$$

where ω_0 is the frequency of the annual variation and ω_1 the frequency of the diurnal variation; ϕ is the latitude.

The incoming short-wave solar radiation through a clear sky at ground level Q_{sc} is about 0.76 of the flux incident at the top of the atmosphere (Gill, 1982):

$$Q_{sc} = \begin{cases} 0.76S \sin(\gamma), & \sin(\gamma) \geq 0, \\ 0.0, & \sin(\gamma) < 0. \end{cases} \quad (9.221)$$

The solar constant $S = 1368 \text{ J}/(\text{m}^2\text{s})$ or W/m^2 . This is the average energy flux at the mean radius of the Earth.

A part of the radiation that reaches the water surface is reflected. The fraction reflected or scattered (surface albedo) is dependent on latitude and season. Cloud cover will reduce the magnitude of the radiation flux that reaches the sea surface. The cloudiness is expressed by a cloud cover fraction F_c , the fraction of the sky covered by clouds. The correction factor for cloud cover is an empirical formula. The absorption of solar radiation is calculated (Gill, 1982) as the product of the net downward flux of short wave-radiation in cloudless conditions and factors correcting for reflection and cloud cover:

$$Q_{sn} = Q_s - Q_{sr} = (1 - \alpha) Q_{sc} f(F_c), \quad (9.222)$$

with:

Q_{sn}	net heat radiation (flux) from the Sun
Q_s	solar radiation (short wave radiation) in $[\text{J}/\text{m}^2\text{s}]$

Table 9.6: Albedo coefficient and cloud function

Heat flux model	albedo (α)	$f(F_c)$
1	0.06	$1.0 - 0.65F_c^2$
2	-	-
3	-	-
4	0.09	-
5	0.06	$1.0 - 0.4F_c - 0.38F_c^2$

Q_{sr} reflected solar radiation in [J/m²s]
 Q_{sc} radiation (flux) for clear sky condition
 α albedo (reflection) coefficient
 F_c fraction of sky covered by clouds (user-defined input)
 $f(F_c)$ function of F_c

The cloud function $f(F_c)$ and the albedo coefficient α used in Delft3D-FLOW are given in [Table 9.6](#). For the Ocean heat flux model the net solar radiation can also be specified directly by the user instead of letting the model determine it from the cloud coverage. Details on this functionality can be found in [section B.5](#).

9.8.3 Atmospheric radiation (long wave radiation)

Atmospheric radiation is primarily due to emission of absorbed solar radiation by water vapour, carbon dioxide and ozone in the atmosphere. The emission spectrum of the atmosphere is highly irregular. The amount of atmospheric radiation that reaches the earth is determined by applying the Stefan-Boltzmann's law that includes the emissivity coefficient of the atmosphere ε . Taking into account the effect of reflection by the surface and reflection and absorption by clouds, the relation for the net atmospheric radiation Q_{an} reads ([Octavio et al., 1977](#)):

$$Q_{an} = (1 - r) \varepsilon \sigma \bar{T}_a^4 g(F_c), \quad (9.223)$$

where \bar{T}_a is the air temperature (in K) and the reflection coefficient $r = 0.03$. The emissivity factor of the atmosphere ε may depend both on vapour pressure and air temperature. The emissivity of the atmosphere varies between 0.7 for clear sky and low temperature and 1.0. The presence of clouds increases the atmospheric radiation. This is expressed in the cloud function $g(F_c)$.

In heat flux model 1 a simple linear formula for the net atmospheric radiation is used ([Octavio et al., 1977](#)):

$$Q_{an} = (218.0 + 6.3T_a) g(F_c), \quad (9.224)$$

with T_a the air temperature (in °C). The cloud function $g(F_c)$ in Eqs. 9.223 and 9.224 is given by:

$$g(F_c) = 1.0 + 0.17F_c^2. \quad (9.225)$$

The linearisation of [Equation 9.223](#) is carried out around $T_a = 15$ °C.

Remarks:

- ◊ In heat flux model 1 the atmospheric radiation is computed.



- ◊ In heat flux model 2 the atmospheric radiation is part of the (measured) net solar radiation flux specified by you.
- ◊ In heat flux models 4 and 5 the atmospheric radiation is part of the total long-wave radiation flux, the so-called effective back radiation, see [subsection 9.8.5](#).
- ◊ The fraction of the sky covered by clouds F_c is fixed for the whole simulation period in the heat flux models 1 and 4. F_c is not used in heat flux models 2 and 3 and can be prescribed as a function of time in heat flux model 5.

9.8.4 Back radiation (long wave radiation)

Water radiates as a near black body, so the heat radiated back by the water can be described by Stefan-Boltzmann's law of radiation, corrected by an emissivity factor $\varepsilon = 0.985$ of water ([Sweers, 1976](#); [Octavio et al., 1977](#)) and the reflection coefficient for the air-water interface $r = 0.03$:

$$Q_{br} = (1 - r) \varepsilon \sigma \bar{T}_s^4, \quad (9.226)$$

with \bar{T}_s the (absolute) water surface temperature in K.

For heat flux models 1 and 2 [Equation 9.226](#) has been linearised around $T_s = 15^\circ\text{C}$:

$$Q_{br} = 303 + 5.2T_s. \quad (9.227)$$

Remark:



- ◊ In heat flux models 1 and 2 the back radiation from the water surface is computed explicitly. In heat flux models 4 and 5 the back radiation is part of the total long-wave radiation flux, the so-called effective back radiation, see [subsection 9.8.5](#). In heat model 3 the back radiation is part of the heat exchange coefficient ([Sweers, 1976](#)).

9.8.5 Effective back radiation

In heat flux models 4 and 5 the total net long wave radiation flux is computed. This is called the effective back radiation:

$$Q_{eb} = Q_{br} - Q_{an}. \quad (9.228)$$

The atmospheric radiation depends on the vapour pressure e_a , see [subsection 9.8.6](#), the air temperature T_a and the cloud cover F_c . The back radiation depends on the surface temperature T_s .

In heat flux model 4 the effective back radiation is computed using Berliand's formula:

$$Q_{eb} = \varepsilon \sigma \bar{T}_a^4 (0.39 - 0.058\sqrt{e_a}) (1.0 - 0.65F_c^2) + 4\varepsilon \sigma \bar{T}_a^3 (\bar{T}_s - \bar{T}_a), \quad (9.229)$$

with the actual vapour pressure e_a given by [Equation 9.235](#).

In heat flux model 5 the effective back radiation Q_{eb} is computed following:

$$Q_{eb} = \varepsilon \sigma \bar{T}_s^4 (0.39 - 0.05\sqrt{e_a}) (1.0 - 0.6F_c^2), \quad (9.230)$$

with the actual vapour pressure e_a given by [Equation 9.237](#).

9.8.6 Evaporative heat flux

Evaporation is an exchange process that takes place at the interface between water and air and depends on the conditions both in the water near the surface and the air above it. The evaporation depends on meteorological factors (wind-driven convection) and vapour pressures.

In heat flux models the evaporative heat flux Q_{ev} is defined by:

$$Q_{ev} = L_v E, \quad (9.231)$$

with L_v the latent heat of vaporisation in J/kg water:

$$L_v = 2.5 \cdot 10^6 - 2.3 \cdot 10^3 T_s. \quad (9.232)$$

The evaporation rate E is defined as the mass of water evaporated per unit area per unit time [kg/(m²s)]. It is computed differently for different heat flux models. Most heat flux models use some form of Dalton's law of mass transfer:

$$E = f(U_{10})(e_s - e_a), \quad (9.233)$$

where the actual vapour pressure e_a and the saturated vapour pressure e_s , can be given by different relations. Heat flux models 1, 2 and 4 use vapour pressures defined as:

$$e_s = 23.38 \exp\left(18.1 - \frac{5303.3}{\bar{T}_s}\right), \quad (9.234)$$

$$e_a = r_{hum} 23.38 \exp\left(18.1 - \frac{5303.3}{\bar{T}_a}\right), \quad (9.235)$$

for the actual and remote vapour pressure, respectively. Heat flux model 5 uses definitions for the vapour pressures as:

$$e_s = 10^{\frac{0.7859+0.03477T_s}{1.0+0.00412T_s}}, \quad (9.236)$$

$$e_a = r_{hum} 10^{\frac{0.7859+0.03477T_a}{1.0+0.00412T_a}}. \quad (9.237)$$

Here r_{hum} is the relative humidity in [-].

Remarks:

- ◊ The relative humidity r_{hum} is specified as a function of time in all heat flux models but the excess temperature model (model 3). In heat flux model 5 it can also be specified varying in space. It should be specified in the input in percentages.
- ◊ For heat flux model 5, the vapour pressures are calculated using air and water surface temperatures in °C, while for the other models the vapour pressures are computed using temperatures in K.
- ◊ The evaporation rate E computed from a heat flux model can be added as a source term to the right hand side of the continuity equation [Equation 9.3](#). The keyword required is `Maseva = #Y#`.
- ◊ By default E is computed from the heat flux model formulation selected, but E can also be prescribed as a function of time. In the latter case this user-defined evaporation overrules the computed evaporation both in the heat flux model and in the continuity equation; see [section B.6](#) on how to prescribe a time dependent evaporation.
- ◊ When the computed E is negative, it is replaced by zero, assuming that it is caused by modelling misfit and not by the actual physical process of water condensation out of the air into the water.



For models 1, 2 and 3 the wind speed function $f(U_{10})$ following [Sweers \(1976\)](#) is used:

$$f(U_{10}) = (3.5 + 2.0U_{10}) \left(\frac{5.0 \times 10^6}{S_{area}} \right)^{0.05}, \quad (9.238)$$

where

S_{area} is the exposed water surface in m^2 , defined in the input and fixed for the whole simulation.

The coefficients calibrated by Sweers were based on the wind speed at 3 meter above the free surface; the coefficients in [Equation 9.238](#) are based on the wind speed 10 meter above the water level.

In heat flux model 4 the wind speed function following [Murakami et al. \(1985\)](#) is used:

$$f(U_{10}) = c_{mur} U_{10}, \quad (9.239)$$

with c_{mur} a constant. This was calibrated for Japanese waters ([Murakami et al., 1985](#)) to be 1.2×10^{-9} , without the influence of free convection. It corresponds to a so-called Dalton number c_e of 0.0016.

Evaporative heat flux for model 5

In heat flux model 5 the evaporation rate is computed from the difference in relative humidity, rather than from the difference in vapour pressure. Another difference between heat flux model 5 and the others is that, for heat flux model 5, the contribution to the evaporative heat flux is split in a contribution by forced convection and a contribution by free convection.

The total heat flux due to evaporation then results from adding the forced convection of latent heat in [Equation 9.241](#) and the free convection of latent heat in [Equation 9.245](#):

$$Q_{ev} = Q_{ev,\text{forced}} + Q_{ev,\text{free}}. \quad (9.240)$$

Both kinds of evaporation are discussed separately.

Forced convection of latent heat

The latent heat flux due to forced convection for heat flux model 5 reads:

$$Q_{ev,\text{forced}} = L_V \rho_a f(U_{10}) \{q_s(T_s) - q_a(T_a)\}, \quad (9.241)$$

with q_s and q_a the specific humidity of respectively saturated air and remote air (10 m above water level):

$$q_s(T_s) = \frac{0.62e_s}{P_{atm} - 0.38e_s}, \quad (9.242)$$

$$q_a(T_a) = \frac{0.62e_a}{P_{atm} - 0.38e_a}. \quad (9.243)$$

The saturated and remote vapour pressures e_s and e_a are given by Eqs. [9.236](#) and [9.237](#).

The wind function in [Equation 9.241](#) is defined as:

$$f(U_{10}) = c_e U_{10}, \quad (9.244)$$

Without the influence of free convection, the Dalton number c_e in model 5 was calibrated for the North Sea to be $c_e = 0.0015$.

Free convection of latent heat

Loss of heat due to evaporation occurs not only by forced convection, wind driven, but also by free convection. Free convection is driven by buoyant forces due to density differences (by temperature and/or water vapour content) creating unstable conditions in the atmospheric boundary layer (ABL). Evaporation due to free convection is important in circumstances where inverse temperature/density gradients are present and wind speeds are almost negligible so that the amount of forced convection is small. Neglecting free convection in this situation will lead to underestimating the heat loss. (Ryan *et al.*, 1974) developed a correction to the wind function, accounting for free convection. The derivation of evaporation by just free convection is based on the analogy of heat and mass transfer.

Free convection is only included in temperature model 5 and the latent heat flux due to free convection reads:

$$Q_{ev,\text{free}} = k_s L_V \bar{\rho}_a (q_s - q_a), \quad (9.245)$$

with the average air density:

$$\bar{\rho}_a = \frac{\rho_{a0} + \rho_{a10}}{2}, \quad (9.246)$$

and with the heat transfer coefficient defined as:

$$k_s = \begin{cases} 0 & \text{if } \rho_{a10} - \rho_{a0} \leq 0 \\ c_{fr.\text{conv}} \left\{ \frac{g\alpha^2}{\nu_{air}\bar{\rho}_a} (\rho_{a10} - \rho_{a0}) \right\}^{1/3} & \text{if } \rho_{a10} - \rho_{a0} > 0 \end{cases} \quad (9.247)$$

where the coefficient of free convection $c_{fr.\text{conv}}$ was calibrated to be 0.14, see (Ryan *et al.*, 1974). The viscosity of air ν_{air} is assumed to have the constant value $16.0 \times 10^{-6} \text{ m}^2/\text{s}$. The molecular diffusivity of air $\alpha \text{ m}^2/\text{s}$ is defined as

$$\alpha = \frac{\nu_{air}}{\sigma}, \quad (9.248)$$

with $\sigma = 0.7$ (for air) the Prandtl number. In Equation 9.245, the saturated air density is given by:

$$\rho_{a0} = \frac{\frac{100P_{atm}-100e_s}{R_{dry}} + \frac{100e_s}{R_{vap}}}{T_s + 273.15}, \quad (9.249)$$

the remote air density (10 m above the water level):

$$\rho_{a10} = \frac{\frac{100P_{atm}-100e_a}{R_{dry}} + \frac{100e_a}{R_{vap}}}{T_{air} + 273.15}, \quad (9.250)$$

where R_{dry} is the gas constant for dry air: 287.05 J/(kg K) and R_{vap} is the gas constant for water vapour: 461.495 J/(kg K). The specific humidity of respectively saturated air and remote air (10 m above the water level), q_s and q_a are given by Eqs. 9.242 and 9.243. The saturated and remote vapour pressure e_s and e_a are defined in Eqs. 9.236 and 9.237.

9.8.7 Convective heat flux

Convective heat flux for model 1, 2 and 4

Assuming that the turbulent exchange of heat at the air-water interface equals the turbulent exchange of mass, the convective heat flux can be related to the evaporative mass flux by the Bowen ratio:

$$Q_{co} = R_b Q_{ev}, \quad (9.251)$$

$$R_b = \gamma \frac{T_s - T_a}{e_s - e_a}, \quad (9.252)$$

with γ the so-called Bowen's constant:

$\gamma = 0.61$ for heat flux models 1 and 2

$\gamma = 0.66$ for heat flux model 4.

The saturated and actual vapour pressures are calculated by Eqs. 9.234 and 9.235 for heat flux models 1, 2 and 4 and by Eqs. 9.236 and 9.237 for heat flux model 5.

Convective heat flux for model 5

In heat flux model 5 the convective heat flux is split into two parts, just as the evaporative heat flux. The convective heat flux is divided into a contribution by forced convection and a contribution by free convection.

The total heat flux due to convection then results from adding the forced convection of sensible heat in [Equation 9.254](#) and the free convection of sensible heat in [Equation 9.256](#):

$$Q_{co} = Q_{co,\text{forced}} + Q_{co,\text{free}}. \quad (9.253)$$

Forced convection of sensible heat

The sensible heat flux due to forced convection is computed by:

$$Q_{co,\text{forced}} = \rho_a c_p g (U_{10}) (T_s - T_a), \quad (9.254)$$

with c_p the specific heat of air. It is considered constant and taken to be 1 004.0 J/(kg K). The wind-speed function $g (U_{10})$ is defined following [Gill \(1982\)](#):

$$g (U_{10}) = c_H U_{10}, \quad (9.255)$$

with c_H the so-called Stanton number. Without the influence of free convection, the Stanton number was calibrated for the North Sea to be $c_H = 0.00145$.

Free convection of sensible heat

The heat transfer for free convection is defined as:

$$Q_{co,\text{free}} = k_s \bar{\rho}_a c_p (T_s - T_a), \quad (9.256)$$

with the heat transfer coefficient k_s given by [Equation 9.247](#).

9.8.8 Overview of heat flux models

Heat flux model 1

For heat flux model 1 the relations used for the computation of the heat balance, [Equation 9.213](#), are given in Table 9.7.

Table 9.7: Terms of the heat balance used in heat model 1

Quantity	Description	According to
Q_{sc}	short wave solar radiation for clear sky	measured
Q_{sn}	net solar radiation	Equation 9.222
Q_{an}	net atmospheric radiation	Eqs. 9.224 and 9.225
Q_{br}	back radiation	Equation 9.227
Q_{ev}	heat loss due to evaporation	Eqs. 9.231 to 9.238
Q_{co}	heat loss due to convection	Eqs. 9.251 and 9.252

Heat flux model 2

Heat flux model 2 is based on formulations given by [Octavio et al. \(1977\)](#). The source of data for the combined net incoming radiation terms ($Q_{sn} + Q_{an}$) must come from direct measurements. The remaining terms (back radiation, evaporation, convection) are computed similarly as for heat flux model option 1, Table [9.8](#).

Table 9.8: Terms of the heat balance used in heat model 2

Quantity	Description	According to
$Q_{sn} + Q_{an}$	net solar and net atmospheric radiation	measured
Q_{br}	back radiation	Equation 9.227
Q_{ev}	heat loss due to evaporation	Eqs. 9.231 to 9.238
Q_{co}	heat loss due to convection	Eqs. 9.251 and 9.252

Excess temperature model - heat flux model 3

The excess temperature model 3 is based on [Sweers \(1976\)](#), the heat exchange flux is represented by a bulk exchange formula:

$$Q_{tot} = -\lambda (T_s - T_{back}), \quad (9.257)$$

with

T_s the water temperature at the free surface, [°C], and
 T_{back} the natural background temperature, [°C].

The heat exchange coefficient λ is a function of the surface temperature T_s and the wind speed U_{10} . It is derived by linearization of the exchange fluxes for back radiation, evaporation and convection. The following relation was derived by [Sweers \(1976\)](#):

$$\lambda = 4.48 + 0.049T_s + f(U_{10}) (1.12 + 0.018T_s + 0.00158T_s^2). \quad (9.258)$$

For the wind function $f(U_{10})$ [Equation 9.238](#) is applied.

Murakami model - heat flux model 4

The heat flux model 4 was calibrated for the Japanese waters ([Murakami et al., 1985](#)). The following relations are used for the computation of the heat terms, [Table 9.9](#).

Table 9.9: Terms of the heat balance used in heat model 4

Quantity	Description	According to
Q_{sn}	net solar radiation	measured
Q_{eb}	effective back radiation	Equation 9.229
Q_{ev}	heat loss due to evaporation	Eqs. 9.231 to 9.235
Q_{co}	heat loss due to convection	Eqs. 9.251 and 9.252

Ocean model - heat flux model 5

The heat flux model 5 following [Gill \(1982\)](#) and [Lane \(1989\)](#) was calibrated for the North Sea and successfully applied for great lakes. The following relations are used for the computation of the heat terms, [Table 9.10](#).

Table 9.10: Terms of the heat balance used in heat model 5

Quantity	Description	According to
Q_{sn}	net solar radiation	Eqs. 9.219 and 9.222
Q_{eb}	effective back radiation	Equation 9.230
Q_{ev}	heat loss due to evaporation	Eqs. 9.241 to 9.240
Q_{co}	heat loss due to convection	Eqs. 9.254 to 9.253

Summary of required input data for the heat flux models

The required time dependent input data for each of the heat flux models is summarised in [Table 9.11](#).

Table 9.11: Summary of time dependent input data of the heat flux models

Quantity	model 1	model 2	Excess temperature	Murakami	Ocean
Q_{sc}	✓				
Q_{sn}				✓	✓ (optional)
$Q_{sn} + Q_{an}$		✓		✓ (optional)	
T_{back}			✓		
T_a	✓	✓		✓	✓
r_{hum}	✓	✓		✓	✓
F_c					✓

9.9 Tide generating forces

Numerical models of tidal motion in coastal seas generally do not account for the direct local influence of the tide generating forces. The amount of water mass in these models is relatively small and the effect of these forces on the flow can be neglected. For coastal areas, the prescription of tidal forcing along open boundaries is sufficient in generating the appropriate and accurate tidal motion.

The need to model larger seashore areas with sections of the deep ocean or large closed basins has increased. In the numerical models of these areas the contribution of the gravitational forces on the water motion increases considerably and can no longer be neglected, see [Figure 9.13](#), to generate an accurate tidal motion.

The tide generating forces originate from the Newtonian gravitational forces of the terrestrial system (Sun, Moon and Earth) on the water mass. Based on their origins the tide generated by these forces can be distinguished into:

- ◊ Equilibrium tide.
Assuming that the earth is covered entirely with water and neglecting any secondary effects (see next section), then the ocean response is called the equilibrium tide, ζ (idealised situation).
- ◊ Earth Tide.
Displacement the earth's crust due to the elasticity of the earth.

9.9.1 Tidal potential of Equilibrium tide

The equilibrium tide is modelled by including the tide generating potential terms for equilibrium tide in the momentum equations. The following terms are added to the right-hand side of the momentum equations Eqs. [9.7](#) and [9.8](#), respectively:

$$\frac{g}{\sqrt{G_{\xi\xi}}} \frac{\partial \varphi}{\partial \xi}, \quad \text{and} \quad \frac{g}{\sqrt{G_{\eta\eta}}} \frac{\partial \varphi}{\partial \eta}. \quad (9.259)$$

Following [Schwiderski \(1980\)](#), the tidal potential φ for equilibrium tide is decomposed into the

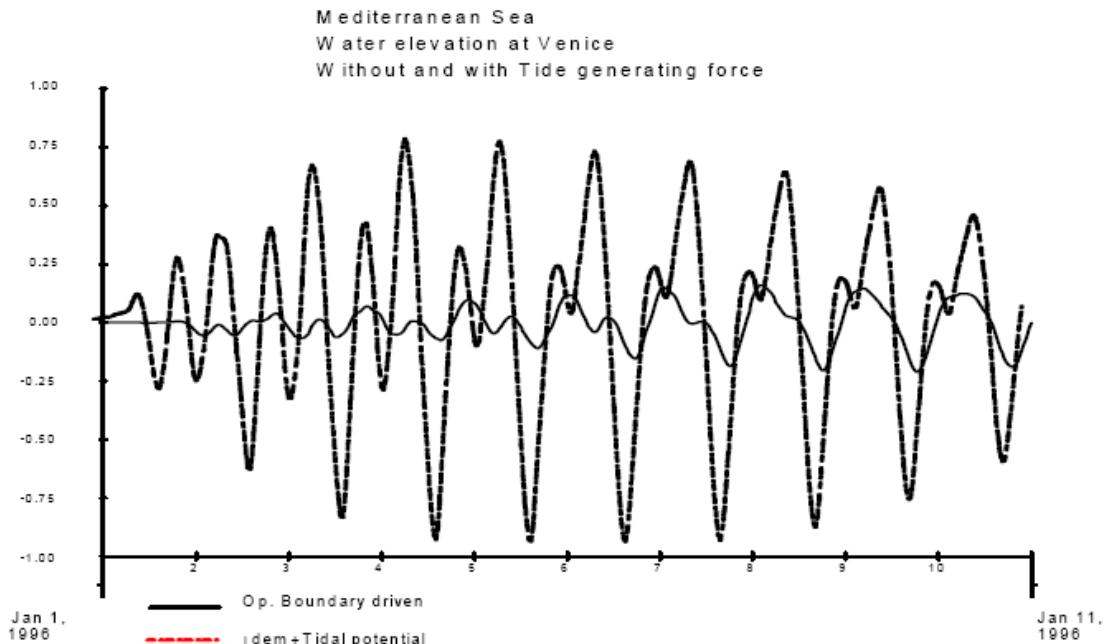


Figure 9.13: Effect of tide generating force on the computed water elevation at Venice

series:

$$\varphi = \sum_{\nu=0,1,2} \varphi_\nu (\lambda, \phi, t), \quad (9.260)$$

consisting of the following major tidal species:

$$\text{Long-Period species } \nu = 0 : \varphi_0 = K_i (1.5 \cos^2 \phi - 1) \cos (\omega_i t + 2\lambda + \chi_i), \quad (9.261)$$

$$\text{Diurnal species } \nu = 1 : \varphi_1 = K_i \sin (2\phi) \cos (\omega_i t + \lambda + \chi_i), \quad (9.262)$$

$$\text{Semi-diurnal species } \nu = 2 : \varphi_2 = K_i \cos^2 (\phi) \cos (\omega_i t + 2\lambda + \chi_i), \quad (9.263)$$

with:

λ, ϕ	geographical co-ordinates
K_i	amplitude of equilibrium tide (component dependent)
ω_i	frequency of equilibrium tide (component dependent)
χ_i	astronomical argument of equilibrium tide (component dependent) relative to UTC midnight
t	universal standard time - UTC

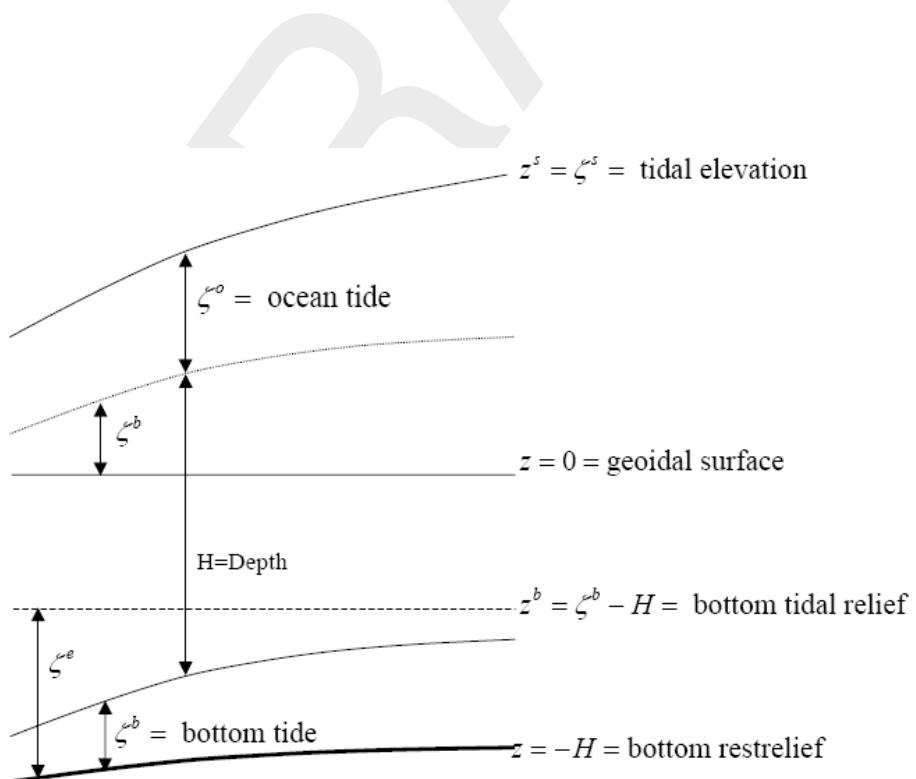
Table 9.12 provides an overview of frequencies ω_i and constants K_i for each of the 11 tidal constituents considered in the species Eqs. 9.261 to 9.263.

Following Schwiderski (1980), all other constituents fall below 4% of the dominating semi-diurnal principal moon (M2) tide and need not be considered in global ocean tide models.

Since tidal models can be used in predictive mode (predicting the actual water elevations and currents) the long term variations (18.6 years cycle) in the tidal components are accounted for by including nodal corrections (see Delft3D-TIDE, User Manual; TIDE UM (2013)). In Delft3D-FLOW the correction is carried out only once at the initialisation of the simulation. Consequently, with regard to the accuracy of the generated tide, this imposes a restriction on the simulation period to approximately one year.

Table 9.12: Constants of major tidal modes

Tidal constituent	Frequency ω_i (degrees/hr)	Constant K_i
Semi-diurnal Species		
M2 principal lunar	28.984104237	0.242334
S2 principal solar	30.000000000	0.113033
N2 elliptical lunar	28.439729554	0.046398
K2 declination lunar-solar	30.082137278	0.030704
Diurnal Species		
K1 declination lunar-solar	15.041068639	0.141565
O1 principal lunar	13.943035598	0.100514
P1 principal solar	14.958931361	0.046843
Q1 elliptical solar	13.398660914	0.019256
Long-period Species		
Mf fortnightly lunar	1.098033041	0.041742
Mm monthly lunar	0.544374684	0.022026
Ssa semi-annual solar	0.082137278	0.019446

**Figure 9.14: Earth ocean tidal interaction (after Schwiderski (1980))**

For the determination of the astronomical arguments χ in Eqs. 9.261 to 9.263 a module of Delft3D-TIDE is applied. If the computations are to be carried out in local time, then the time difference between the local time and UTC must be specified. This difference is referred to as the local time zone (LTZ) in the input.

9.9.2 Tidal potential of Earth tide

The equilibrium tide approach concentrates on the influence of gravitational forces on ocean tides. It neglects the effect of these forces on terrestrial tide (i.e. it assumes that earth is a stiff body) and/or the interaction between the two. Following Schwiderski, the following two interaction effects should be considered:

- 1 Earth tide (ζ^e) and tidal loading ($\zeta^{eo} = \zeta^e - \zeta^b$), are the earth tidal responses to the equilibrium tide φ and oceanic tidal load ζ .
- 2 φ^e , φ^{eo} and φ^o are perturbations of the equilibrium tide φ in response to ζ^e , ζ^{eo} and ζ , respectively.

These perturbations are subsequently regrouped in the final equations according to their causes and are approximated as follows:

$$\varphi^e - \zeta^e = 0.31\varphi, \quad (9.264)$$

and:

$$\varphi^o + \zeta^{eo} - \varphi^{eo} = 0.10\zeta. \quad (9.265)$$

However the improvement due to the second correction term is considerably less significant (10% improvement in the difference between observed and simulated phases), than the inclusion of proper eddy viscosity and bottom friction coefficients for the hydrodynamics of an ocean basin, see [Schwiderski \(1980\)](#). Therefore, in Delft3D-FLOW only the first correction term has been included.

Consequently, in the momentum equations, Eqs. 9.7 and 9.8, the following net tidal potential terms are applied, yielding a 31 % reduction of the originally derived equilibrium tide potential terms):

$$0.69 \frac{g}{\sqrt{G_{\xi\xi}}} \frac{\partial \varphi}{\partial \xi}, \quad \text{and} \quad 0.69 \frac{g}{\sqrt{G_{\eta\eta}}} \frac{\partial \varphi}{\partial \eta}. \quad (9.266)$$

9.10 Hydraulic structures

Obstacles in the flow may generate sudden transitions from flow contraction to flow expansion. The grid resolution is often low compared to the gradients of the water level, the velocity and the bathymetry. The hydrostatic pressure assumption may locally be invalid. Examples of these obstacles in civil engineering are: gates, barriers, sluices, groynes, weirs, bridge piers and porous plates. The obstacles generate energy losses and may change the direction of the flow.

The forces due to obstacles in the flow which are not resolved (sub-grid) on the horizontal grid, should be parameterised. The obstacles are denoted in Delft3D-FLOW as hydraulic structures. In the numerical model the hydraulic structures should be located at velocity points of the staggered grid, see [section 10.1](#). The direction of the forces: U or V should be specified at input. To model the force on the flow generated by a hydraulic structure, the flow in a

computational layer is blocked or a quadratic energy or linear loss term is added to the momentum equation. The mathematical formulations are given in this section. For the numerical implementation we refer to Section 10.9.

9.10.1 3D gate

In 3D simulations, constructions which partially block the horizontal flow can be modelled as so-called “gates”. Its horizontal and vertical position can be specified. The horizontal velocities of the computational layers at position of the gate are set to zero. This generates a vertical structure of the horizontal flow around a hydraulic structure which is more realistic. There is no transport of salt or sediment through the computational layers of a gate. The width of a gate is assumed to be zero, so it has no influence on the water volume. Upstream of the gate the flow is accelerated due to contraction and downstream the flow is decelerated due to expansion.

Furthermore, a free slip boundary conditions has been implemented at the transition from an open layer to a closed layer (representing a gate):

$$\nu_V \frac{\partial U}{\partial z} \Big|_{z=a} = 0, \quad (9.267)$$

with $z = a$ the top or the bottom of the gate.

9.10.2 Quadratic friction

The flow rate Q through a hydraulic structure can often be described by a so-called Q - H relation. They relate the flow rate to the difference between the upstream and downstream water levels:

$$Q = \mu A \sqrt{2g|\zeta_u - \zeta_d|}, \quad (9.268)$$

with μ the contraction coefficient ($0 < \mu \leq 1$), the wet flow-through area and the upstream and downstream water level, respectively. The contraction coefficient in the so-called Q-H relation is dependent on the kind of hydraulic structure. The contraction coefficient μ can be used to determine the additional quadratic friction term in the momentum equation.

$$M_\xi = -\frac{c_{loss-u}}{\Delta x} u \sqrt{u^2 + v^2}, \quad (9.269)$$

$$M_\eta = -\frac{c_{loss-v}}{\Delta y} v \sqrt{u^2 + v^2}, \quad (9.270)$$

with c_{loss} the energy loss coefficient. For the following types of hydraulic structures, a quadratic formulation for the energy loss is implemented in Delft3D-FLOW:

- ◊ barrier
- ◊ bridge pier
- ◊ Current Deflecting Wall (CDW)
- ◊ local weir or 2D weir
- ◊ porous plate
- ◊ culvert

9.10.3 Linear friction

To reduce the flow rate through a certain cross-section Delft3D-FLOW offers also the functionality of a linear friction term. This kind of hydraulic structure is denoted as a *rigid sheet*:

$$M_\xi = -\frac{c_{loss-u} u}{\Delta x}, \quad (9.271)$$

$$M_\eta = -\frac{c_{loss-v} v}{\Delta y}, \quad (9.272)$$

with c_{loss} the energy loss coefficient, which should be specified by you at input.

9.11 Flow resistance: bedforms and vegetation

The terrain and vegetation exert shear stresses on the passing flow. The magnitude of the shear stress of the bed is often characterised by means of roughness coefficient of type Chézy, Manning or White-Colebrook as discussed in [subsection 9.4.1](#). Within the main stream flow the shear stresses are largely determined by the local conditions of the alluvial bed (bed composition and bedform characteristics). In other areas, such the floodplains of rivers and in the intertidal areas of estuaries, the flow resistance is determined by a combination of vegetation and an alluvial bedforms or even a non-alluvial bed. To accurately represent such conditions in the numerical model, Delft3D-FLOW has been extended with a couple features, this includes bedform roughness predictors and vegetation models. These types of flow resistance may be resolved in a 2D numerical model using the trachytope approach (see [subsection 9.11.2](#)) and in a 3D model by a combination of bed resistance formulations. First, we start with a description of the bedform roughness height predictor.

9.11.1 Bedform heights

There are various types of bedforms; each type influences the hydro- and morphodynamics in its own way. Following [Van Rijn \(2007\)](#) we distinguish between roughness heights for ripples, mega-ripples and dunes. Delft3D contains a module to predict dune characteristics (height/length) which may be used in determining the threshold level for dredging (see keyword `UseDunes` in [subsection A.2.23](#)) or to compute a dune roughness height which can be used as the bed roughness in the flow computation via trachytope formulae 105 ([Equation 9.312](#)) or 106 ([Equation 9.313](#)) as described in [subsection 9.11.2](#). One can also determine the dune, mega-ripple, and ripple roughness heights directly according [Van Rijn \(2007\)](#). Both approaches are described in the two sections below. The corresponding keywords are described in [Appendix B.16](#).

9.11.1.1 Dune height predictor

Four dune height predictors have been implemented: [Van Rijn \(1984c\)](#), [Fredsøe \(1982\)](#) based on [Meyer-Peter and Müller \(1948\)](#) or [Engelund and Hansen \(1967\)](#), and a power relation. One of these predictors can be selected using the keyword `BdfH`. In case of [Van Rijn \(1984c\)](#) the dune height h_d is computed as

$$h_d = 0.11\varepsilon H \left(\frac{D_{50}}{H} \right)^{0.3} (1 - e^{-T/2})(25 - T) \quad (9.273)$$

where ε is a calibration coefficient and T is the dimensionless bed shear parameter. [Fredsøe \(1982\)](#) formulated an approach to obtain a dune height predictor for a general sediment transport formula $s = f(U, H, k)$ depending on depth-averaged flow magnitude U , water depth H and Nikuradse bed roughness k as

$$h_d = \varepsilon s H / \left(U \frac{\partial s}{\partial U} - H \frac{\partial s}{\partial H} - k \frac{\partial s}{\partial k} \right) \quad (9.274)$$

If one applies this approach to the transport formula by [Meyer-Peter and Müller \(1948\)](#) one obtains

$$h_d = \frac{24}{63} \varepsilon H \max\left(1 - \frac{\theta_{cr}}{\mu\theta}, 0\right) \quad (9.275)$$

where μ is the efficiency factor $(C/C_{g,90})^{1.5}$ based on a grain related bed roughness of $3D_{90}$ and θ_{cr} is the critical mobility parameter based on the Shields curve or given by the user. If one applies the formulation of [Fredsøe \(1982\)](#) to the transport formula by [Engelund and Hansen \(1967\)](#) one obtains

$$h_d = \frac{1}{5} \varepsilon H \quad (9.276)$$

The latter two formulations have been implemented in Delft3D. Finally, the power relation dune height predictor reads

$$h_d = a_H H^{b_H} \quad (9.277)$$

The dune height formulations given by equations 9.273–9.277 are approximations for the dune height under equilibrium conditions. Under time and spatially varying conditions, the dune height will need time to adjust; this effect has been implemented by means of an advection relaxation equation.

$$\frac{\partial h_d}{\partial t} + u_H \frac{\partial h_d}{\partial x} + v_H \frac{\partial h_d}{\partial y} = \frac{h_{d,eq} - h_d}{T_H} \quad (9.278)$$

By default these effects are neglected, such that $h_d = h_{d,eq}$. By means of the keyword `BdfRlx` one can select four options for computing the variables u_H , v_H and T_H : (i) constant relaxation time scale T_H , (ii) constant relaxation distance L_H , (iii) relaxation distance proportional to dune length L_d , and (iv) relaxation distance dependent on dune length L_d and water depth H . In the first three advection cases, the direction of the bedform migration is assumed to be equal to the direction of the depth averaged velocity in all cases, i.e.

$$u_H = c_H \frac{u}{U} \quad (9.279)$$

$$v_H = c_H \frac{v}{U} \quad (9.280)$$

where u and v are the x and y components of the depth averaged velocity, U the flow velocity magnitude $\sqrt{u^2 + v^2}$, and c_H is the bedform celerity (magnitude) given by a power relation derived for a generic transport formula approximation $s = a_s U^{b_s}$ as

$$c_H = \frac{b_s s}{H} = \frac{a_s b_s U^{b_s}}{H} = \frac{a_C U^{b_C}}{H} \quad (9.281)$$

while the relaxation time scale T_H is either given, or computed from $T_H = L_H/c_H$ where the relaxation distance L_H is either given or proportional to the dune length $L_H = f_{LH} L_d$. In the fourth advection case T_H is computed again using $T_H = L_H/c_H$ but in that case the relaxation distance L_H is given by

$$L_H = \min \left[\left(\frac{H_{\max}}{H} \right)^{b_{LH}}, \phi_{\max} \right] L_d \quad (9.282)$$

and the bedform celerity is given by

$$c_H = a_{cH} \max \left[\left(\frac{H}{H_{\max}} \right)^{b_{cH}}, \gamma_{\min} \right] \frac{s_b}{H(1 - Fr^2)} \quad (9.283)$$

where s_b is the total bedload transport rate, Fr^2 is the squared Froude number U^2/gH and H_{\max} , b_{LH} , ϕ_{\max} , a_{cH} , b_{cH} and γ_{\min} are constants that can be set. In this case the bedform migration direction is assumed to be equal to the direction of the bedload transport vector \vec{s}_b .

For the dune length L_d used in some of the formulations above, you can choose either one of the following two options using keyword `BdfL`. If you choose to compute the dune height according [Van Rijn \(1984c\)](#) then it will be computed as

$$L_d = 7.3H \quad (9.284)$$

otherwise it will be computed using the power relation

$$L_d = a_L H^{b_L} \quad (9.285)$$

Note that only the dune height is subject to the advective and relaxation behaviour formulated in [Equation 9.278](#); for the dune length always these equilibrium formulations are used.

Finally, you may select one of three methods for computing the bedform roughness height by means of the keyword `BdfRou`; these methods are based on the formulations of [Van Rijn \(1984c\)](#), [Van Rijn \(2007\)](#) or a power relation. This value will only be used if you use the trachytope option for specifying bed roughness and include either one of alluvial roughness formulae 105 and 106. In case of [Van Rijn \(1984c\)](#) the roughness height related to dunes is computed as

$$k_{s,d} = 1.1h_d (1 - e^{-25h_d/L_d}) \quad (9.286)$$

and in case of the power relation this quantity is computed as

$$k_{s,d} = a_R H_d^{b_R} \quad (9.287)$$

The roughness height related to ripples is set in both cases equal to $3D_{90}$. In case of [Van Rijn \(2007\)](#) the roughness heights neither depend on the computed dune height nor on the dune length; in that case the equations given in the following section are used.

9.11.1.2 [Van Rijn \(2007\)](#) bedform roughness height predictor

As proposed by Eq. (5a)–(5d) in [Van Rijn \(2007\)](#) the roughness height predictor for ripples is given by

$$k_{s,r} = \alpha_r \begin{cases} 20D_{silt} & \text{if } D_{50} < D_{silt} \\ k_{s,r}^* & \text{otherwise} \end{cases} \quad (9.288)$$

where α_r is a calibration factor, $D_{silt} = 32 \mu\text{m}$ and $k_{s,r}^*$ is given by

$$k_{s,r}^* = \begin{cases} 150f_{cs}D_{50} & \text{if } \psi \leq 50 \\ (182.5 - 0.65\psi)f_{cs}D_{50} & \text{if } 50 < \psi \leq 250 \\ 20f_{cs}D_{50} & \text{if } \psi > 250 \end{cases} \quad (9.289)$$

in which ψ is the current-wave mobility parameter defined as $U_{wc}^2/[(s - 1)gD_{50}]$ using $U_{wc}^2 = U_w^2 + u_c^2$ and relative density $s = \rho_s/\rho_w$, peak orbital velocity near bed $U_w =$

$\pi H_s/[T_r \sinh(2kh)]$, depth-averaged current velocity u_c , significant wave height H_s , wave number $k = 2\pi/L$, wave length L derived from $(L/T_p \pm u_c)^2 = gL \tanh(2\pi h/L)/(2\pi)$, relative wave period T_r , peak wave period T_p , water depth H and factor $f_{cs} = (0.25D_{\text{gravel}}/D_{50})^{1.5}$ in which $D_{\text{gravel}} = 0.002$ m. The value of $k_{s,r}$ is limited to values ranging from D_{90} to 2% of the water depth.

Following Eq. (6a)–(6d) by [Van Rijn \(2007\)](#) the roughness height predictor for mega-ripples is given by

$$k_{s,mr} = \alpha_{mr} \begin{cases} 0 & \text{if } D_{50} < D_{\text{silt}} \\ k_{s,mr}^* & \text{otherwise} \end{cases} \quad (9.290)$$

where α_{mr} is a calibration factor and $k_{s,mr}^*$ is given by

$$k_{s,mr}^* = \begin{cases} 0.0002f_{fs}\psi H & \text{if } \psi \leq 50 \\ (0.011 - 0.00002\psi)f_{fs}H & \text{if } 50 < \psi \leq 550 \\ 0.02 & \text{if } \psi > 550 \text{ and } D_{50} \geq D_{\text{sand}} \\ 200D_{50} & \text{if } \psi > 550 \text{ and } D_{\text{silt}} \leq D_{50} < D_{\text{sand}} \end{cases} \quad (9.291)$$

in which $f_{fs} = (D_{50}/1.5D_{\text{sand}})$ and $D_{\text{sand}} = 0.000062$ m. The value of $k_{s,mr}$ is limited to values smaller than 0.2 m.

Following Eq. (7a)–(7d) by [Van Rijn \(2007\)](#) the roughness height predictor for dunes is given by

$$k_{s,d} = \alpha_d \begin{cases} 0 & \text{if } D_{50} < D_{\text{silt}} \\ k_{s,d}^* & \text{otherwise} \end{cases} \quad (9.292)$$

where α_d is a calibration factor and $k_{s,d}^*$ is given by

$$k_{s,d}^* = \begin{cases} 0.0004f_{fs}\psi H & \text{if } \psi \leq 100 \\ (0.048 - 0.00008\psi)f_{fs}H & \text{if } 100 < \psi \leq 600 \\ 0 & \text{if } \psi > 600 \end{cases} \quad (9.293)$$

Since these formulations do not depend on the equations for the dune characteristics given in the previous section, the advection and relaxation effects represented by [Equation 9.278](#) cannot be included in these [Van Rijn \(2007\)](#) formulations. Therefore, a separate relaxation option has been implemented for all three bedform roughness heights according [Van Rijn \(2007\)](#) formulations as given by

$$\begin{aligned} k_{s,r} &= (1 - \alpha_r)k_{s,r,new} + \alpha_r k_{s,r,old} & \text{where } \alpha_r = e^{-\Delta t/T_r} \\ k_{s,mr} &= (1 - \alpha_{mr})k_{s,mr,new} + \alpha_{mr} k_{s,mr,old} & \text{where } \alpha_{mr} = e^{-\Delta t/T_{mr}} \\ k_{s,d} &= (1 - \alpha_d)k_{s,d,new} + \alpha_d k_{s,d,old} & \text{where } \alpha_d = e^{-\Delta t/T_d} \end{aligned} \quad (9.294)$$

where Δt is the hydrodynamic time step of bed roughness updating, and T_r , T_{mr} and T_d are different relaxation time scales for the three bedform types. Note that if a relaxation time scale T_* becomes zero, the corresponding relaxation factor α_* will be zero and the relaxation effect disappears.

Warning:

- ◊ Until Delft3D release 3.28.10, the relaxation factors α_* are not defined as indicated above but as $\max[1 - (5\Delta t/T_*), 0]$.



**Remarks:**

- ◊ In case of a simulation with multiple sediment fractions, the D_{50} of the grain mixture in the surface layer is used.
- ◊ Non-erodible layers and partial transport effects are not taken into account to reduce the roughness height.

9.11.2 Trachytopes

This functionality allows you to specify the bed roughness and flow resistance on a sub-grid level by defining and using various land use or roughness/resistance classes, further referred to as trachytopes after the Greek word $\tauραχύτης$ for roughness. The input parameters and files to use the trachytopes functionality are described in [section B.17](#).

At every time step (or less frequent as requested by the user) the trachytopes are converted into a representative bed roughness C , k or n (as described in [subsection 9.4.1](#)) and optional linear flow resistance coefficient $\lambda_{m,n}$ per velocity point.

$$M_\xi = -\frac{1}{2} \lambda_{m,n} u \sqrt{u^2 + v^2} \quad (9.295)$$

$$M_\eta = -\frac{1}{2} \lambda_{m,n} v \sqrt{u^2 + v^2} \quad (9.296)$$

To save computational time the user may choose to update the computed bed roughness and resistance coefficients less frequently than every time step. See [section B.17](#) for a description of the keywords and input files associated with this feature.

The following two sections describe the various classes of trachytopes distinguished and the way in which they are combined, respectively.

9.11.2.1 Trachytope classes

Three base classes of trachytopes are distinguished: area classes, line classes and point classes. The area classes (type range 51–200) basically cover the whole area, therefore, they are generally the dominant roughness factor. The line classes (type range 201–250) may be used to represent hedges and similar flow resistance elements; it will add anisotropy to the roughness field. The point class (type range 251–300) represents a set of point flow resistance elements. The next six sections provide an overview of the various trachytope formulae implemented.

Special classes (1–50)

In addition to the three base classes two special trachytype classes have been defined: a flood protected area and a composite trachytype class. The first class represents a sub-grid area that is protected from flooding and thus does not contribute to the bed roughness; however, the effect on the flow resistance should be taken into account. The second class can be used to make derived trachytype classes that are a combination of two other trachytypes: an area fraction α of trachytype type T_1 and an area fraction β (often equal to $1 - \alpha$) of trachytype type T_2 .

FormNr	Name	Formula
Special classes (1–50)		
1	flood protected area	area fraction shows up as f_b in Eqs. 9.348 and 9.351
2	composite trachytype	fraction α of type T_1 and fraction β (generally $\beta = 1 - \alpha$) of type T_2

Area trachytype classes (51–200)

The class of area trachytypes is subdivided into three types: simple (51–100), alluvial (101–150) and vegetation (151–200). Four simple area trachytypes have been implemented representing the four standard roughness types of flow module.

FormNr	Name	Formula
51	White-Colebrook value	k
52	Chézy value	C
53	Manning value	$C = \sqrt[6]{h}/n$
54	z_0 value	$k = 30z_0$

Six alluvial trachytypes have been implemented.

FormNr	Name	Formula
101	simplified Van Rijn power relation	Equation 9.297
102		Equation 9.298
103	Van Rijn (1984c)	Equations 9.299 to 9.307
104	Struiksma	Equations 9.308 to 9.311
105	bedforms quadratic	Equation 9.312
106	bedforms linear	Equation 9.313

The first alluvial roughness formula is a simplified version of the [Van Rijn \(1984c\)](#) alluvial roughness predictor

$$k = Ah^{0.7} \left[1 - e^{-Bh^{-0.3}} \right] \quad (9.297)$$

it is obtained from [Equation 9.299](#) by noting that $h_b \propto h^{0.7}$ and $L_b \propto h$ and ignoring the grain related roughness. The parameters A and B can be calibrated by the user. The second

formula implemented is a straightforward general power law

$$C = Ah^B \quad (9.298)$$

where A and B are calibration coefficients. The [Van Rijn \(1984c\)](#) alluvial roughness predictor reads

$$k = k_{90} + 1.1h_b \left(1 - e^{-25h_b/L_b}\right) \quad (9.299)$$

where the bedform height h_b and length L_b are given by

$$h_b = 0.11h \left(\frac{D_{50}}{h}\right)^{0.3} (1 - e^{-T/2}) (25 - T) \quad (9.300)$$

$$L_b = 7.3h \quad (9.301)$$

where h is the local water depth and the transport stage parameter T is given by

$$T = \frac{u'^2 - u_{*,cr}^2}{u_{*,cr}^2} \quad (9.302)$$

where u' is the bed shear velocity given by

$$u'^2 = gu^2/C_{g,90}^2 \quad (9.303)$$

where

$$C_{g,90} = 18^{10} \log(12h/k_{90}) \text{ and } k_{90} = 3D_{90} \quad (9.304)$$

and $u_{*,cr}$ is the critical bed shear velocity according Shields given by

$$u_{*,cr}^2 = g\Delta D_{50}\theta_c \quad (9.305)$$

given

$$\theta_c = \begin{cases} 0.24/D_* & \text{if } D_* \leq 4 \\ 0.14D_*^{-0.64} & \text{if } 4 < D_* \leq 10 \\ 0.04D_*^{-0.10} & \text{if } 10 < D_* \leq 20 \\ 0.013D_*^{0.29} & \text{if } 20 < D_* \leq 150 \\ 0.055 & \text{if } 150 < D_* \end{cases} \quad (9.306)$$

where

$$D_* = D_{50} \left(\frac{g\Delta}{\nu^2}\right)^{1/3} \quad (9.307)$$

This predictor does not contain any calibration coefficients but requires D_{50} and D_{90} data from the morphology module. It does not include the advective and relaxation behaviour that is available by explicitly simulating the dune height as described in [Section 9.11.1](#) combined with trachytope number 106.

The second alluvial roughness predictor proposed by (Struiksma, pers. comm.) allows for a lot of adjustments, it reads

$$\frac{1}{C^2} = (1 - \xi) \frac{1}{C_{90}^2} + \xi \frac{1}{C_{min}^2} \quad (9.308)$$

where

$$C_{90} = A_1^{10} \log(A_2 h / D_{90}) \quad (9.309)$$

and

$$\xi = \frac{\max(0, \theta_g - \theta_c)}{\theta_m - \theta_c} \frac{\theta_m^2 - \theta_c \theta_g}{(\theta_m - \theta_c) \theta_g} \quad (9.310)$$

which varies from 0 at $\theta_g \leq \theta_c$ to 1 at $\theta_g = \theta_m$ where

$$\theta_g = \frac{u^2}{C_{90}^2 \Delta D_{50}} \quad (9.311)$$

and $A_1, A_2, \theta_c, \theta_m, C_{\min}$ are coefficients that the user needs to specify. This formula requires also D_{50} and D_{90} data from the morphology module. The fifth formula is based on [Van Rijn \(2007\)](#) and reads

$$k = \min(\sqrt{k_{s,r}^2 + k_{s,mr}^2 + k_{s,d}^2}, \frac{h}{2}) \quad (9.312)$$

It uses the roughness heights of ripples k_r , mega-ripples k_{mr} and dunes k_d . These roughness heights are based on [Van Rijn \(2007\)](#) formulae as described in [subsection 9.11.1](#); these formulae depend on D_{50} and D_{90} data which may be either specified as part of the roughness type or obtained from the morphology module. The sixth formula is similar, but uses a linear addition

$$k = \min(k_{s,r} + k_{s,mr} + k_{s,d}, \frac{h}{2}) \quad (9.313)$$

Four vegetation based area trachytopes have been implemented. Two formulae (referred to as ‘Barneveld’) are based on the work by [Klopstra et al. \(1996, 1997\)](#) and two on the work by [Baptist \(2005\)](#).

FormNr	Name	Formula
151	Barneveld 1	Eqs. 9.314 – 9.323, $C_D = 1.65$
152	Barneveld 2	Eqs. 9.314 – 9.320, 9.324 – 9.326
153	Baptist 1	Eqs. 9.327 and 9.328
154	Baptist 2	Eqs. 9.329, 9.331 and 9.332

The formula by [Klopstra et al. \(1997\)](#) reads

$$C = \frac{1}{h^{3/2}} \left\{ \begin{array}{l} \frac{2}{\sqrt{2A}} \left(\sqrt{C_3 e^{h_v \sqrt{2A}} + u_{v0}^2} \right) + \\ \frac{u_{v0}}{\sqrt{2A}} \ln \left(\frac{(\sqrt{C_3 e^{h_v \sqrt{2A}} + u_{v0}^2} - u_{v0})(\sqrt{C_3 + u_{v0}^2} + u_{v0})}{(\sqrt{C_3 e^{2\sqrt{2A}} + u_{v0}^2} + u_{v0})(\sqrt{C_3 + u_{v0}^2} - u_{v0})} \right) + \\ \frac{\sqrt{g(h - (h_v - a))}}{\kappa} \left((h - (h_v - a)) \ln \left(\frac{h - (h_v - a)}{z_0} \right) - a \ln \left(\frac{a}{z_0} \right) - (h - h_v) \right) \end{array} \right\} \quad (9.314)$$

where

$$A = \frac{n C_D}{2\alpha} \quad (9.315)$$

$$C_3 = \frac{2g(h - h_v)}{\alpha\sqrt{2A}(e^{h_v\sqrt{2A}} + e^{-h_v\sqrt{2A}})} \quad (9.316)$$

$$a = \frac{1 + \sqrt{1 + \frac{4E_1^2\kappa^2(h - h_v)}{g}}}{\frac{2E_1^2\kappa^2}{g}} \quad (9.317)$$

and

$$z_0 = ae^{-F} \quad (9.318)$$

where

$$E_1 = \frac{\sqrt{2AC_3}e^{h_v\sqrt{2A}}}{2\sqrt{C_3e^{h_v\sqrt{2A}} + u_{v0}^2}} \quad (9.319)$$

and

$$F = \frac{\kappa\sqrt{C_3e^{h_v\sqrt{2A}} + u_{v0}^2}}{\sqrt{g(h - (h_v - a))}} \quad (9.320)$$

Here, h is the water depth, h_v is the vegetation height, and $n = mD$ where m is the number of stems per square metre and D is the stem diameter. For the first implementation the parameter α in [Equation 9.316](#) is given by

$$\alpha = \max(0.001, 0.01\sqrt{hh_v}) \quad (9.321)$$

and the velocity within the vegetation is approximated by $u_{v0}\sqrt{i}$ where

$$u_{v0}^2 = \frac{2g}{C_D n} \quad (9.322)$$

and i is the water level gradient. For emerged vegetation the first implementation reads

$$\frac{1}{C^2} = \frac{C_D nh}{2g} \quad (9.323)$$

The second implementation based on [Klopstra et al. \(1996\)](#) is identical except for the following modifications to Eqs. [9.321 – 9.323](#). The main difference between the two implementations is the inclusion of the roughness C_b of the bed itself (without vegetation). The parameter α in [Equation 9.316](#) is now given by

$$\alpha = 0.0227h_v^{0.7} \quad (9.324)$$

and the velocity within the vegetation is approximated by $u_{v0}\sqrt{i}$ where

$$u_{v0}^2 = \frac{h_v}{\frac{C_D h_v n}{2g} + \frac{1}{C_b^2}} \quad (9.325)$$

and i is the water level gradient. For emerged vegetation the second implementation reads

$$\frac{1}{C^2} = \frac{C_D nh}{2g} + \frac{1}{C_b^2} \quad (9.326)$$

For large values of C_b the latter two equations simplify to the corresponding equations of the first implementation. The first implementation requires vegetation height h_v and density n as input parameters (the drag coefficient C_D is equal to 1.65); for second implementation you'll also need to specify the drag coefficient C_D and the alluvial bed roughness k_b (C_b in [Equation 9.326](#) is computed as $18^{10} \log(12h/k_b)$).

The first implementation of the roughness predictor by Baptist ([Baptist, 2005](#)) reads for the case of submerged vegetation

$$C = \frac{1}{\sqrt{\frac{1}{C_b^2} + \frac{C_D nh_v}{2g}}} + \frac{\sqrt{g}}{\kappa} \ln\left(\frac{h}{h_v}\right) \quad (9.327)$$

where n is the vegetation density ($n = mD$ where m is the number of stems per square metre and D is the stem diameter). The second term goes to zero at the transition from submerged to emerged vegetation. At that transition the formula changes into the formula for non-submerged vegetation which reads

$$C = \frac{1}{\sqrt{\frac{1}{C_b^2} + \frac{C_D nh}{2g}}} \quad (9.328)$$

which is identical to the non-submerged case of the second implementation of the work by [Klopstra et al. \(1996\)](#) (see [Equation 9.326](#)).

The drawback of the three vegetation based formulations above is that they parameterize the flow resistance by means of the bed roughness. Consequently, the presence of vegetation will lead to a higher bed roughness and thus to a higher bed shear stress and larger sediment transport rates in case of morphological computations. Therefore, we have included a $-\frac{\lambda}{2} u^2$ term in the momentum equation where λ represents the flow resistance of the vegetation. For the case of non-submerged vegetation $h < h_v$ the flow resistance and bed roughness are strictly separated

$$C = C_b \quad \text{and} \quad \lambda = C_D n \quad (9.329)$$

In the case of submerged vegetation $h > h_v$ the two terms can't be split in an equally clean manner. However, we can split the terms such that the bed shear stress computed using the depth averaged velocity u and the net bed roughness C equals the bed shear stress computed using the velocity u_v within the vegetation layer and the real bed roughness C_b .

$$\frac{u^2}{C^2} = \frac{u_v^2}{C_b^2} \quad (9.330)$$

With this additional requirement we can rewrite [Equation 9.327](#) as

$$C = C_b + \frac{\sqrt{g}}{\kappa} \ln\left(\frac{h}{h_v}\right) \sqrt{1 + \frac{C_D nh_v C_b^2}{2g}} \quad (9.331)$$

and

$$\lambda = C_D n \frac{h_v}{h} \frac{C_b^2}{C^2} \quad (9.332)$$

which simplify to [Equation 9.329](#) for $h = h_v$. Both formulae by Baptist require vegetation height h_v , density n , drag coefficient C_D and alluvial bed roughness C_b as input parameters.

Linear trachytype classes (201–250)

Two formulae have been implemented for linear trachytypes such as hedges or bridge piers.

FormNr	Name	Formula
201	hedges 1	Eqs. 9.333 to 9.335
202	hedges 2	Eqs. 9.336 to 9.338

The first implementation reads

$$\frac{1}{C^2} = \frac{h}{2g} \frac{L_{hedge}}{W_{cell} L_{cell}} \frac{1 - \mu^2}{\mu^2} \quad (9.333)$$

where L_{hedge} is the projected length of the hedge, W_{cell} and L_{cell} are the width and length of the grid cell. The ratio L_{hedge}/W_{cell} may be interpreted as the number of hedges that the flow encounters per unit width. The second ratio is thus the inverse of the average distance between these hedges within the grid cell. The last term may be loosely referred to as the drag of the hedge, which is determined by the hedge pass factor μ given by

$$\mu = 1 + 0.175n \left(\frac{h}{h_v} - 2 \right) \quad (9.334)$$

if the hedge extends above the water level ($h_v > h$) and is given by

$$\mu = 1 - 0.175n \left(\frac{h}{h_v} \right) \quad (9.335)$$

if the hedge is fully submerged ($h > h_v$) where n is a dimensionless hedge density. The second implementation reads

$$\frac{1}{C^2} = \frac{C_D n L_{hedge} h}{2g L_{cell} W_{cell}} \quad (9.336)$$

or equivalently

$$C = \sqrt{\frac{2g L_{cell} W_{cell}}{h L_{hedge}}} \left(\sqrt{\frac{1}{C_D n}} \right) \quad (9.337)$$

for non-submerged conditions and

$$C = \sqrt{\frac{2g L_{cell} W_{cell}}{h L_{hedge}}} \left(\frac{h_v}{h} \sqrt{\frac{1}{C_D n}} + m_0 \sqrt{\frac{\left(\frac{h-h_v}{h}\right)^2}{1 - \left(\frac{h-h_v}{h}\right)^2}} \right) \quad (9.338)$$

for submerged conditions. We recognize the same ratio $L_{cell} W_{cell} / L_{hedge}$ that represents the average distance between hedges. [Equation 9.336](#) can be directly compared to similar equations for area trachytypes ([Equation 9.323](#)), point trachytypes ([Equation 9.339](#)) and bridge resistance ([Equation 10.76](#)). Note that the formula for computing the loss coefficient for a bridge explicitly includes the reduction in the flow area and the resulting increase in the effective flow velocity, whereas the above mentioned trachytype formulae don't.

Point trachytype classes: various (251–300)

One formula for point trachytypes has been implemented. It may be used to represent groups of individual trees or on a smaller scale plants.

FormNr	Name	Formula
251	trees	Eqn. 9.339

The implemented formula reads

$$C = \sqrt{\frac{2g}{C_D n \min(h_v, h)}} \quad (9.339)$$

where $n = mD$ with m the number of trees per unit area and D the characteristic tree diameter, h_v is the vegetation height and h is the local water depth. The formula is identical to [Equation 9.328](#) except for the fact that the point trachytype formula has no bed roughness and area associated with it. The generalization of [Equation 9.339](#) to the submerged case ($h > h_v$) lacks the extra term in [Equation 9.327](#).

9.11.2.2 Averaging and accumulation of trachytypes

Point and linear roughnesses are accumulated by summing the inverse of the squared Chézy values C_i .

$$\frac{1}{C_{pnt}^2} = \sum_i \frac{1}{C_{pnt,i}^2} \quad (9.340)$$

$$\frac{1}{C_{lin}^2} = \sum_i \frac{1}{C_{lin,i}^2} \quad (9.341)$$

The area roughnesses are accumulated weighted by the surface area fraction f_i . These roughnesses are accumulated as White-Colebrook roughness values and as Chézy values; for the latter values both the linear sum (“parallel”) and the sum of inverse of squared values (“serial”) are computed. Roughness values are converted into each other as needed based on the local water depth using Eqs. 9.54–9.56.

$$k_{area} = \sum_i f_i k_i \quad (9.342)$$

$$\frac{1}{C_{area,s}^2} = \sum_i f_i \frac{1}{C_i^2} \quad (9.343)$$

$$C_{area,p} = \sum_i f_i C_i \quad (9.344)$$

For the fraction of the grid cell area for which no roughness class is specified the default roughness is used.

The flow resistance coefficients are also accumulated proportionally to the surface area fraction f_i associated with the trachytype considered. For the fraction of the grid cell area for which no flow resistance is specified, obviously none is used.

$$\lambda = \sum_i f_i \lambda_i \quad (9.345)$$

The final effective bed roughness of the grid cell may be computed by either one of the following two methods.

Method 1

The total mean roughness is computed by summing the White-Colebrook values for the areas and line and point resistance features.

$$k_m = k_{area} + k_{lin} + k_{pnt} \quad (9.346)$$

where $k_{lin} = 12h10^{-C_{lin}/18}$ and $k_{pnt} = 12h10^{-C_{pnt}/18}$. The effect of the water free area fraction f_b is taken into account by means of the following empirical relation in which $C_m = 18^{10}\log(12h/k_m)$ is the mean Chézy value corresponding to the total mean White-Colebrook roughness value obtained from [Equation 9.346](#).

$$f_b = \max(\min(0.843, f_b), 0.014) \quad (9.347)$$

$$C_{total} = C_m \left(1.12 - 0.25f_b - 0.99\sqrt{f_b} \right) \quad (9.348)$$

The resulting C_{total} value is used in the computation. This method together with trachytype classes 1, 51, 101, 151 and 201 corresponds to the NIKURADSE option of the WAQUA/TRIWAQ flow solver.

Method 2

The total roughness is computed by first averaging over the serial and parallel averages of the Chézy values according

$$C_{area} = \alpha_s C_{area,s} + (1 - \alpha_s) C_{area,p} \quad (9.349)$$

where $\alpha_s = 0.6$ by default. Subsequently the effect of the water free area fraction f_b is taken into account by means of the following empirical relation (identical to [Equation 9.348](#) of method 1).

$$f_b = \max(\min(0.843, f_b), 0.014) \quad (9.350)$$

$$C_{area,corr} = C_{area} \left(1.12 - 0.25f_b - 0.99\sqrt{f_b} \right) \quad (9.351)$$

Finally the Chézy value representing the total bed roughness is computed by accumulating the inverses of the squared Chézy values.

$$\frac{1}{C_{total}^2} = \frac{1}{C_{area,corr}^2} + \frac{1}{C_{lin}^2} + \frac{1}{C_{pnt}^2} \quad (9.352)$$

The resulting C_{total} value is used in the computation. This method together with trachytype classes 1, 51, 52, 53, 101, 152, 202 and 251 corresponds to the ROUGHCOMBINATION option of the WAQUA/TRIWAQ flow solver.

9.11.3 (Rigid) 3D Vegetation model

In a detailed numerical model one may want to represent the vertical variations in the vegetation characteristics and study the effect of vegetation on the 3D flow and turbulence.

The interaction of submersed vegetation upon hydrodynamics and fysical processes has received increasing attention because of its influence on water quality in lakes and on morphological developments in salt marshes ([Houwing et al., 2000](#); [WL | Delft Hydraulics, 1998](#)).

[Uittenbogaard \(dec. 2000\)](#) formulated theory for incorporating the effects of vegetation upon momentum and turbulence equations and implemented this in the so called '(Rigid) 3D Vegetation model', see [Winterwerp and Uittenbogaard \(1997\)](#), that has extensively been tested and compared to experiments ([Meijer, 1998](#)) by [Oberez \(2001\)](#).

The main input parameter for this formulation is the plant geometry, see [subsection A.2.31](#). The implementation of vegetation resistance can also be applied for 2DH computations.

Theoretical background

The basic input parameters are the number of stems per unit area as function of height $n(z)$, and the stem width as function of height $\phi(z)$.

The influence of the vegetation upon the momentum equations is given by the vertical distribution of the friction force as caused by cylindrical elements in oblique flow:

$$F(z) = \frac{1}{2} \rho_0 C_D \phi(z) n(z) |u(z)| u(z) \quad [\text{N/m}^3] \quad (9.353)$$

with $u(z)$ the horizontal flow velocity profile and C_D the cylindrical resistance coefficient (default value 1.0).

The horizontal cross sectional plant area is given by:

$$A_p(z) = \frac{\pi}{4} \phi^2(z) n(z) \quad (9.354)$$

The influence of the vegetation upon vertical mixing is reflected in an extra source term T in the kinetic turbulent energy equation:

$$\frac{\partial k}{\partial t} = \frac{1}{1 - A_p} \frac{\partial}{\partial z} \left\{ (1 - A_p) \left(\nu + \frac{\nu_t}{\sigma_k} \right) \frac{\partial k}{\partial z} \right\} + T + P_k - B_k - \varepsilon \quad (9.355)$$

with $T(z)$ the work spent by upon the fluid:

$$T(z) = F(z) u(z) \quad (9.356)$$

and an extra source term $T\tau^{-1}$ in the epsilon equation,

$$\frac{\partial \varepsilon}{\partial t} = \frac{1}{1 - A_p} \frac{\partial}{\partial z} \left\{ (1 - A_p) \left(\nu + \frac{\nu_t}{\sigma_k} \right) \frac{\partial \varepsilon}{\partial z} \right\} + T\tau^{-1} + P_\varepsilon - B_\varepsilon - \varepsilon_\varepsilon \quad (9.357)$$

with τ the minimum of:

$$\tau = \min(\tau_{\text{free}}, \tau_{\text{veg}}) \quad (9.358)$$

with the dissipation time scale of free turbulence

$$\tau_{\text{free}} = \frac{1}{c_{2\varepsilon}} \frac{k}{\varepsilon} \quad (9.359)$$

and the dissipation time scale of eddies in between the plants

$$\tau_{\text{veg}} = \frac{1}{c_{2\varepsilon} \sqrt{c_\mu}} \sqrt[3]{\frac{L^2}{T}} \quad (9.360)$$

that have a typical size limited by the smallest distance in between the stems:

$$L(z) = C_l \sqrt{\frac{1 - A_p(z)}{n(z)}} \quad (9.361)$$

C_l is a coefficient reducing the geometrical length scale to the typical volume averaged turbulence length scale. Uittenbogaard presents a closure and finds that a C_l value of 0.07 is applicable for grid generated turbulence. For vegetation this coefficient may be tuned, values of 0.8 are found applicable, see [Uittenbogaard \(dec. 2000\)](#).

10 Numerical aspects of Delft3D-FLOW

The mathematical formulations of the 3D shallow water flow and transport model Delft3D-FLOW were presented in [chapter 9](#). To solve the partial differential equations the equations should be transformed to the discrete space. In this chapter the space discretisation, time integration and numerical solution methods applied in Delft3D-FLOW will be described.

The description provided here applies in most cases for both the vertical σ co-ordinate system (σ -model) and the vertical z co-ordinate system (Z -model). Numerical aspects that are specifically valid only for the Z -model system will be described separately in [chapter 12](#).

Delft3D-FLOW is applied for modelling a wide range of different flow conditions: e.g. turbulent flows in laboratory flumes, tidal flow in estuaries and seas, rapidly varying flows in rivers, density driven flows by thermal discharges, wind driven flows in lakes and ocean dynamics. For all these applications (with complete different length scales) Delft3D-FLOW should give a solution. For the choice of the numerical methods, robustness had high priority.

10.1 Staggered grid

The numerical method of Delft3D-FLOW is based on finite differences. To discretise the 3D shallow water equations in space, the model area is covered by a curvilinear grid. It is assumed that the grid is orthogonal and well-structured, see [Figure 10.1](#).

The grid co-ordinates can be defined either in a Cartesian or in a spherical co-ordinate system. In both cases a curvilinear grid, a file with curvilinear grid co-ordinates in the physical space, has to be provided. Such a file may be generated by a grid generator, see the User Manual of RGFGRID.

The numerical grid transformation is implicitly known by the mapping of the co-ordinates of the grid vertices from the physical to the computational space. The geometrical quantities $\sqrt{G_{\xi\xi}}$ and $\sqrt{G_{\eta\eta}}$ introduced in the transformed equations, see Eqs. 9.3 to 9.8, have to be discretised on the computational grid, see [Figure 10.2](#).

The primitive variables water level and velocity (u, v, w) describe the flow. To discretise the 3D shallow water equations, the variables are arranged in a special way on the grid, see [Figure 10.2](#) and [Figure 10.3](#). The pattern is called a staggered grid. This particular arrangement of the variables is called the Arakawa C-grid. The water level points (pressure points) are defined in the centre of a (continuity) cell. The velocity components are perpendicular to the grid cell faces where they are situated.

Staggered grids have several advantages such as:

- ◊ Boundary conditions can be implemented in a rather simple way.
- ◊ It is possible to use a smaller number of discrete state variables in comparison with discretizations on non-staggered grids, to obtain the same accuracy.
- ◊ Staggered grids for shallow water solvers prevent spatial oscillations in the water levels; see e.g. [Stelling \(1984\)](#).

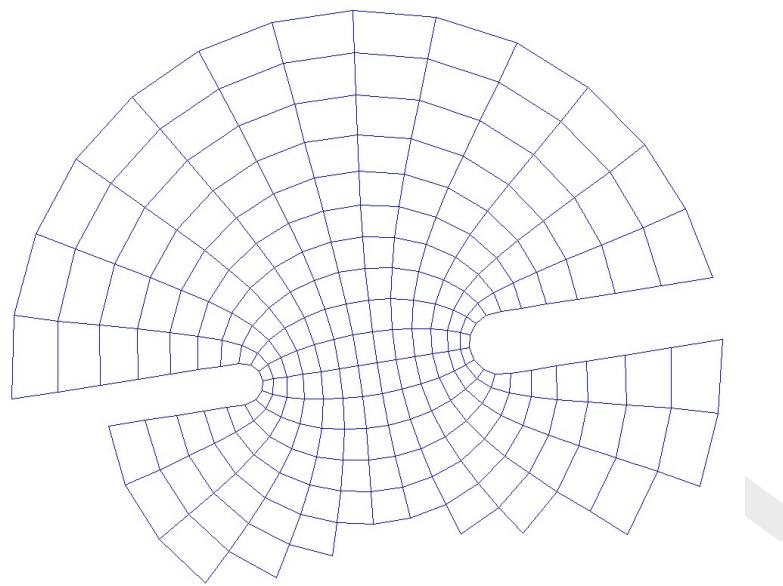


Figure 10.1: Example of a grid in Delft3D-FLOW

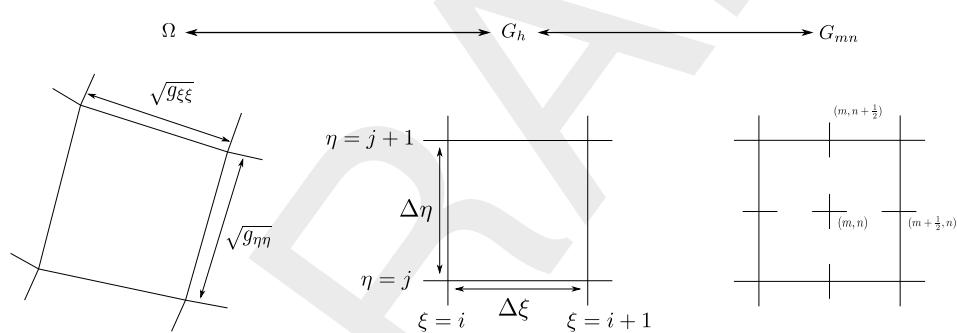
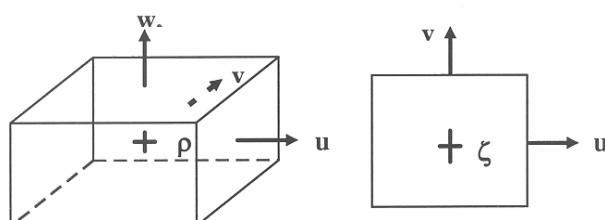


Figure 10.2: Mapping of physical space to computational space



Legend:

- + water level (ζ) / density (ρ) point
- velocity point (u, v or w)

Figure 10.3: Grid staggering, 3D view (left) and top view (right)

10.2 σ -grid and Z -grid

In the vertical direction, two types of grid are available:

- ◊ a boundary-fitted σ co-ordinates and
- ◊ a grid that is strictly horizontal also called the Z -grid.

For the σ co-ordinate grid, the number of layers over the entire horizontal computational area is constant, irrespective of the local water depth. The distribution of the relative layer thickness is usually non-uniform. This allows for more resolution in the zones of interest such as the near surface area (important for e.g. wind-driven flows, heat exchange with the atmosphere) and the near bed area (sediment transport).

The σ -grid is commonly used in Delft3D-FLOW. However, occasionally this grid may not be sufficient to solve problems where stratified flow can occur in combination with steep topography. The σ -grid, though boundary fitted, will generally not have enough resolution around the pycnocline which is strictly horizontal in the physical space. High resolution around pycnocline is in this situation preferred. Therefore, recently a second vertical grid co-ordinate system based on Cartesian co-ordinates (Z -grid) was introduced in Delft3D-FLOW for 3D simulations of weakly forced stratified water systems.

The Z -grid has horizontal co-ordinate lines that are (nearly) parallel with density interfaces (isopycnals) in regions with steep bottom slopes. This is important to reduce artificial mixing of scalar properties such as salinity and temperature. The Z -grid is not boundary-fitted in the vertical. The bottom (and free surface) is usually not a co-ordinate line and is represented as a staircase (see previous chapter).

10.3 Definition of model boundaries

The horizontal model area is defined by specifying the so-called computational grid enclosure (automatically generated by RGFGRID). The computational grid enclosure consists of one or more closed polygons that specify the boundaries of the model area. There are two types of boundaries: closed boundaries along “land-water” lines (coastlines, riverbanks) and open boundaries across the flow field. The open boundaries are artificial and chosen to limit the computational area. The polygons consist of line pieces connecting water level points on the numerical grid, with a direction parallel to the grid lines or diagonal (45 degrees) through the grid. The computational cells on the grid enclosure are land points (permanent dry) or open boundary points. An island may be removed from the computational domain by specifying a closed polygon (“land-water line”) as part of the grid enclosure. If a computational grid enclosure is not specified, then a default rectangular computational grid is assumed. A default enclosure is spanned by the lines connecting the water level points (1, 1), (Mmax, 1), (Mmax, Nmax) and (1, Nmax).

[Figure 10.4](#) shows for a horizontal model area the grid staggering, the grid enclosure and the position of open and closed boundaries. [Figure 10.5](#) shows the corresponding grid of control volumes which may be generated by the grid generator RGFGRID. This example will now be explained in detail.

The computational grid enclosure for the grid of control volumes shown in [Figure 10.4](#) is defined by the following polygon of line pieces:

```
(1,1) - (5,1)
(5,1) - (7,3)
(7,3) - (8,3)
```

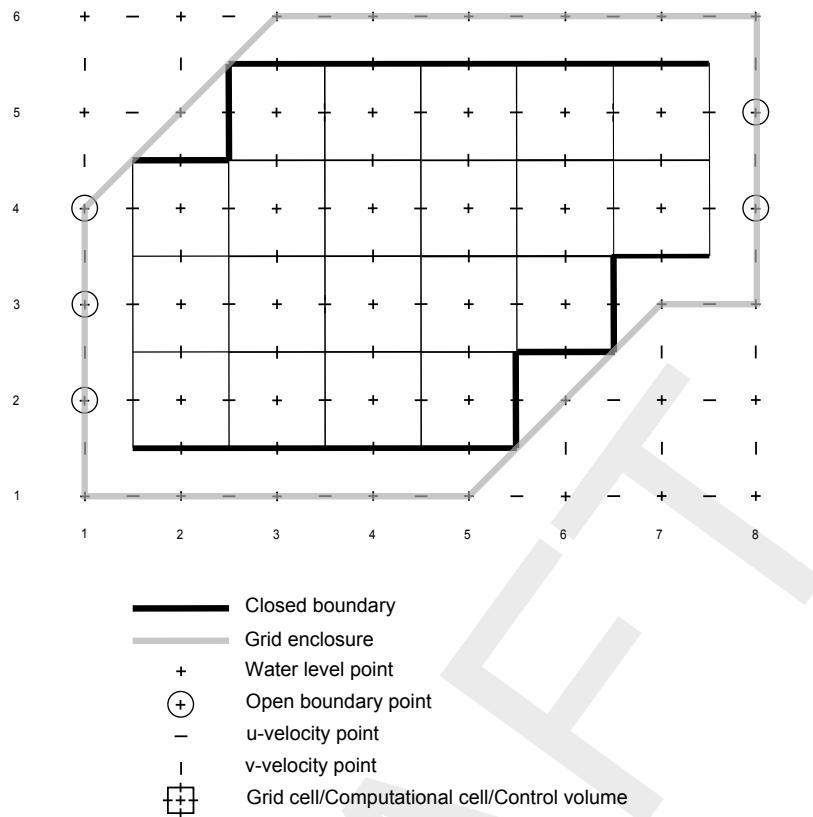


Figure 10.4: Example of Delft3D-FLOW model area

$(8, 3) - (8, 6)$
 $(8, 6) - (3, 6)$
 $(3, 6) - (1, 4)$
 $(1, 4) - (1, 1)$

The array dimensions $Mmax=8$ and $Nmax=6$, with $Mmax-2$ and $Nmax-2$ the number of grid cells in ξ - and η -direction, respectively. Open boundaries are located at the line pieces $(m,n)=(1,2)$ to $(1,3)$ and at $(m,n)=(8,4)$ to $(8,5)$. For this example, the grid lines and corresponding control volumes drawn by the grid generator RGFGRID are shown in [Figure 10.5](#). We remark that the dimensions of the grid in RGFGRID are $Mmax-1$ and $Nmax-1$, counting the grid corners. *Thus, in both directions the dimension of the grid in RGFGRID is one less than the dimension specified for the model area in the MDF-file.* This has to do with the staggered grid and is often a source of misunderstanding. We remark that the depth files (files with extension `<dep>`) should have entries for all $Mmax$ by $Nmax$ values, see `<*.dep>` files and [subsection A.2.4](#). For a detailed discussion of the grid, the grid enclosure and the location of boundary points see [Appendix E](#).

After generating a curvilinear grid with the grid generator RGFGRID , as output file there is also a computational grid enclosure file, which can be used by Delft3D-FLOW.

10.4 Time integration of the 3D shallow water equations

Following [Stelling \(1984\)](#), a robust solver for the shallow water equations has to satisfy the following demands:

- ◊ Robustness (unconditionally stable).
- ◊ Accuracy (at least second order consistency).

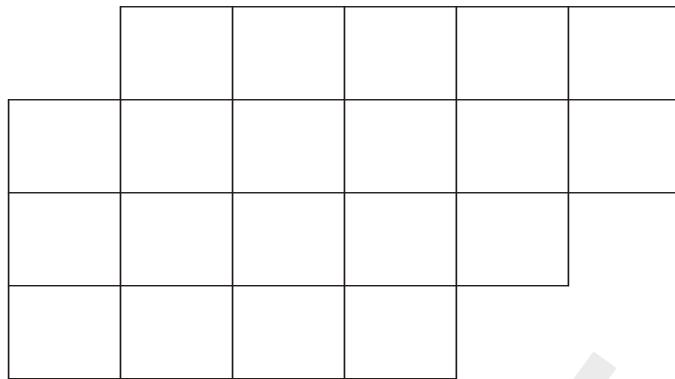


Figure 10.5: Example of Delft3D-FLOW grid

- ◊ Suitable for both time-dependent and steady state problems.
- ◊ Computationally efficient.

An explicit time integration of the shallow water equations on a rectangular grid is subject to a time step condition based on the Courant number for wave propagation:

$$CFL_{wave} = 2\Delta t \sqrt{gH} \sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}} < 1, \quad (10.1)$$

where Δt is the time step, g is the acceleration of gravity, H is the total water depth and $\Delta x = \sqrt{G_{\xi\xi}}$ and $\Delta y = \sqrt{G_{\eta\eta}}$ are the smallest grid spaces in ξ - and η -direction of the physical space. For many practical applications, this requires a time step of only a few seconds to simulate tidal propagation. Exceeding the time step would generate instability and from the view of robustness, this is not acceptable. Therefore, an implicit method is needed. However, an implicit scheme can be uneconomic in computer time and storage if the inversion of a large matrix is required. A straightforward implicit finite difference approximation is the Crank-Nicholson method. For the linearised depth-averaged shallow water equations without advection and bottom friction, the Crank-Nicholson method in vector form reads:

$$\frac{\vec{U}^{\ell+1} - \vec{U}^\ell}{\Delta t} + \frac{1}{2} A \vec{U}^\ell + \frac{1}{2} A \vec{U}^{\ell+1} = \vec{0}, \quad (10.2)$$

with:

$$\vec{U} = (u, v, \zeta)^T, \quad (10.3)$$

$$A = \begin{pmatrix} 0 & -f & g \frac{\partial}{\partial x} \\ f & 0 & g \frac{\partial}{\partial y} \\ H \frac{\partial}{\partial x} & H \frac{\partial}{\partial y} & 0 \end{pmatrix}, \quad (10.4)$$

and:

$$H = d + \zeta. \quad (10.5)$$

The solution after one time step is given by

$$\vec{U}^{\ell+1} = \left(I + \frac{1}{2} \Delta t A \right)^{-1} \times \left(I - \frac{1}{2} \Delta t A \right) \vec{U}^\ell. \quad (10.6)$$

After discretization, the spatial differential operators in both grid directions generate a system of difference equations with a band matrix. In each row there are at least five non-zero entries, corresponding to the grid cells surrounding the water level point. The band width is relatively large and all grid points are implicitly coupled. Solving the equations would require a large computational effort. In the next section a computationally efficient time integration method will be described, which is applied for the σ -grid in Delft3D-FLOW.

10.4.1 ADI time integration method

[Leendertse \(1967\)](#); [Leendertse and Gritton \(1971\)](#); [Leendertse et al. \(1973\)](#) introduced an *Alternating Direction Implicit* (ADI) method for the shallow water equations. The ADI-method splits one time step into two stages. Each stage consists of half a time step. In both stages, all the terms of the model equations are solved in a consistent way with at least second order accuracy in space.

For the spatial discretization of the horizontal advection terms, three options are available in Delft3D-FLOW. The first and second option use higher-order dissipative approximations of the advection terms. The time integration is based on the ADI-method. The first scheme is denoted as the *WAQUA*-scheme, see [Stelling \(1984\)](#) and [Stelling and Leendertse \(1992\)](#). The second scheme is denoted as the *Cyclic* method; see [Stelling and Leendertse \(1992\)](#). The *WAQUA*-scheme and the *Cyclic* method do not impose a time step restriction. The third scheme can be applied for problems that include rapidly varying flows for instance in hydraulic jumps and bores ([Stelling and Duinmeijer, 2003](#)). The scheme is denoted as the *Flooding* scheme and was developed for 2D simulation with a rectilinear grid of the inundation of dry land with obstacles such as road banks and dikes. The integration of the advection term is explicit and the time step is restricted by the Courant number for advection. We discuss the time integration for scheme *WAQUA* and method *Cyclic*. For the *Flooding* scheme, the advection term is taken at the previous time level and moved to the right-hand side.

For the water level gradient and the advection terms, the time levels are alternating; if in one stage a term is taken implicitly in time, this term will be taken explicitly in time in the other stage. For the complete time step, each separate term is still integrated second-order accurate in time. The advantage of the ADI-method is that the implicitly integrated water levels and velocities are coupled along grid lines leading to systems of equations with a small bandwidth. Substitution of the discrete momentum equations in the continuity equation leads to a tri-diagonal system of equations for the water levels. After computing the water levels, back substitution of the water levels in the discrete momentum equations gives the velocities; see [Stelling \(1984\)](#) for full details.

In vector form (for the 2D case) the ADI-method is given by:

Step 1:

$$\frac{\vec{U}^{\ell+\frac{1}{2}} - \vec{U}^\ell}{\frac{1}{2}\Delta t} + \frac{1}{2}A_x \vec{U}^{\ell+\frac{1}{2}} + \frac{1}{2}A_y \vec{U}^\ell + B\vec{U}^{\ell+\frac{1}{2}} = \vec{d}, \quad (10.7)$$

Step 2:

$$\frac{\vec{U}^{\ell+1} - \vec{U}^{\ell+\frac{1}{2}}}{\frac{1}{2}\Delta t} + \frac{1}{2}A_x \vec{U}^{\ell+\frac{1}{2}} + \frac{1}{2}A_y \vec{U}^{\ell+1} + B\vec{U}^{\ell+1} = \vec{d}, \quad (10.8)$$

with:

$$A_x = \begin{pmatrix} 0 & -f & g \frac{\partial}{\partial x} \\ 0 & u \frac{\partial}{\partial x} + v \frac{\partial}{\partial y} & 0 \\ H \frac{\partial}{\partial x} & 0 & u \frac{\partial}{\partial x} \end{pmatrix}, \quad (10.9)$$

$$A_y = \begin{pmatrix} u \frac{\partial}{\partial x} + v \frac{\partial}{\partial y} & 0 & 0 \\ f & 0 & g \frac{\partial}{\partial y} \\ 0 & H \frac{\partial}{\partial y} & v \frac{\partial}{\partial y} \end{pmatrix}, \quad (10.10)$$

and:

$$B = \begin{pmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad (10.11)$$

with λ the linearized bottom friction coefficient. To improve stability the bottom friction is integrated implicitly for each stage. \vec{d} is the right-hand side containing external forcings like wind and atmospheric pressure. The time integration of the horizontal viscosity terms is discussed in [subsection 10.4.4](#) and is dependent on the formulation.

In the first stage the time level proceeds from ℓ to $\ell + \frac{1}{2}$ and the simulation time from $t = \ell \Delta t$ to $t = (\ell + \frac{1}{2}) \Delta t$. In this stage first the V -momentum equation, [Equation 9.8](#), is solved, followed by the U -momentum equation, [Equation 9.7](#), which is implicitly coupled with the continuity equation, [Equation 9.3](#), by the free surface gradient. In the second stage the time level proceeds from $\ell + \frac{1}{2}$ to $\ell + 1$. In this stage first the U -momentum equation is solved, followed by the V -momentum equation which is implicitly coupled with the continuity equation by the free surface gradient. In the stage, in which the barotropic pressure term (i.e. water level gradient) is integrated implicitly, the advection terms and viscosity terms are integrated explicitly. Similarly in the stage, in which the barotropic pressure term (i.e. water level gradient) is integrated explicitly, the advection terms and viscosity terms are integrated implicitly.

The second stage in the ADI-method is almost similar to the first stage. The grid coefficients, direction dependent roughness coefficients and the u -velocity and v -velocity are interchanged. The only principal difference between the u - and v -momentum equations is the sign of the Coriolis term. In Delft3D-FLOW the similarity of the difference equations is used in the implementation. The same subroutine is used for the momentum equation in both directions, only the sign of the Coriolis term is dependent on the computational direction.

For the 3D shallow water equations the horizontal velocity components are coupled in the vertical direction by the vertical advection and viscosity term. An explicit time integration of the vertical exchange terms on a σ -co-ordinate or a Z -co-ordinate grid would in shallow areas lead to very severe time step limitations:

$$\Delta t \leq \frac{(\Delta \sigma H)^2}{2\nu_V} \text{ or } \Delta t \leq \frac{(\Delta z)^2}{2\nu_V}, \quad (10.12)$$

$$\Delta t \leq \frac{\Delta \sigma H}{\omega} \text{ or } \Delta t \leq \frac{\Delta z}{w}, \quad (10.13)$$

where $\Delta \sigma H$ and ΔZ are the vertical grid size and ω is the transformed vertical velocity and w the vertical velocity. Therefore, in the vertical direction a fully implicit time integration method is applied, which is first-order accurate in time and leads to tri-diagonal systems of equations. The vertical coupling of the discretized momentum equations is eliminated by a double sweep algorithm.

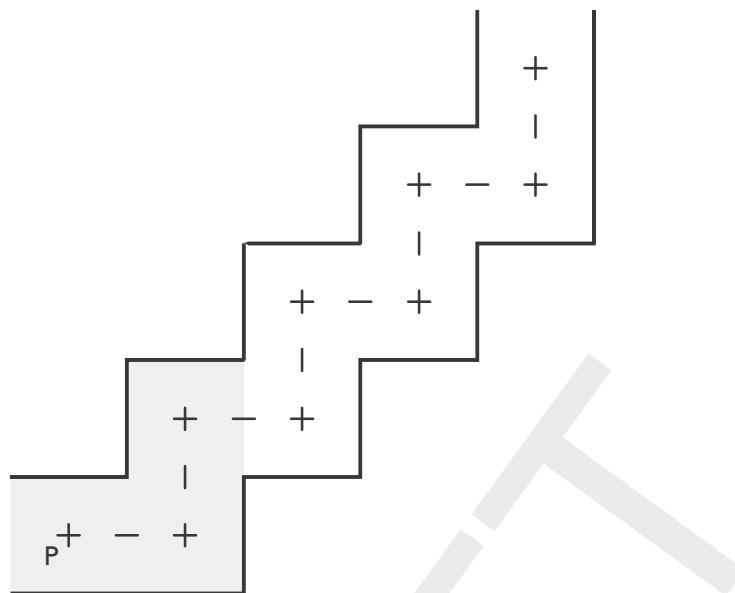


Figure 10.6: Numerical region of influence for one time step, “Zig-zag channel”

10.4.2 Accuracy of wave propagation

The standard time integration method in Delft3D-FLOW for the σ -model is the ADI-method described in the previous section. Water levels and velocities are implicitly solved along grid lines. The wave propagation is related to the Courant number, see [Equation 10.1](#). It is known, see e.g. [Benqué et al. \(1982\)](#) and [Weare \(1979\)](#), that for flow along irregular (staircase) closed boundaries, flow around islands, flow over tidal flats, and through “zig-zag”, see [Figure 10.6](#), channels where the grid lines do not smoothly follow the geometry, the ADI-method is inaccurate for Courant numbers larger than $4\sqrt{2}$. The numerical region of influence does not correspond to the mathematical region of influence following the theory of characteristics for hyperbolic systems of equations. The ADI-method may lead to inaccurately predicted flow patterns, see e.g. [Stelling \(1984\)](#). This inaccuracy is called the ADI-effect and is introduced by the splitting of the spatial operator in two directions. A free surface wave cannot travel through more than two bends of 90 degrees in one complete ADI time step ([Stelling, 1984](#)), see [Figure 10.6](#).

The upper bound for the Courant number of $4\sqrt{2}$ occurs in the most critical situation, namely in case of a narrow channel (width of few grid sizes) that makes an angle of 45 degrees with the computational grid. In practical situations the Courant number should not exceed a value of 10. However, this is a rough estimate. You are advised to carry out sensitivity tests in order to determine the largest time step for which the ADI-method still yields accurate results.

10.4.3 Iterative procedure continuity equation

The non-linearity in the coupled continuity equation, [Equation 9.3](#), and momentum equation, Eqs. [9.7](#) and [9.8](#) requires an iterative procedure. For the U -momentum equation:

$$\frac{\partial}{\partial \xi} \left(\sqrt{G_{\eta\eta}} HU \right)^{\ell+\frac{1}{2}} \approx \frac{\left(\sqrt{G_{\eta\eta}}^{\xi} HU \right)_{m+\frac{1}{2},n}^{\ell+\frac{1}{2}} - \left(\sqrt{G_{\eta\eta}}^{\xi} HU \right)_{m-\frac{1}{2},n}^{\ell+\frac{1}{2}}}{\Delta\xi} \quad (10.14)$$

$$\approx \frac{\left(\sqrt{G_{\eta\eta}}^{\xi} H_{m+\frac{1}{2},n}^{[q]} U_{m+\frac{1}{2},n}^{[q]} \right) - \left(\sqrt{G_{\eta\eta}}^{\xi} H_{m-\frac{1}{2},n}^{[q]} U_{m-\frac{1}{2},n}^{[q]} \right)}{\Delta\xi}. \quad (10.15)$$

The iterative procedure in Delft3D-FLOW removes the non-linear term by multiplying the free surface gradient in the discretized momentum equation by the factor:

$$r^{[q]} = \frac{H_{m,n}^{[q-1]}}{H_{m,n}^{[q]}}. \quad (10.16)$$

This factor converges to one. The iterative procedure in Delft3D-FLOW is mass conservative after each iteration. Two iterations are enough both for accuracy and stability. The number of iterations is set in the MDF-file by the keyword `Iter`.

For rapidly varying flows, the water depth at velocity points may be discontinuous. To obtain accurate approximations of the local water depth at discontinuities slope limiters are used, see [subsection 10.5.1](#). In this way positive water depths are guaranteed. This requires local linearization of the continuity equation. The same approach is used for the Z -model:

$$\frac{\partial}{\partial \xi} \left(\sqrt{G_{\eta\eta}} HU \right)^{\ell+\frac{1}{2}} \approx \frac{\left(\sqrt{G_{\eta\eta}}^{\xi} H_{m+\frac{1}{2},n}^{\ell} U_{m+\frac{1}{2},n}^{\ell+\frac{1}{2}} \right) - \left(\sqrt{G_{\eta\eta}}^{\xi} H_{m-\frac{1}{2},n}^{\ell} U_{m-\frac{1}{2},n}^{\ell+\frac{1}{2}} \right)}{\Delta\xi} \quad (10.17)$$

This linearization of the continuity equation is used in combination with the *Flooding* scheme for advection and the Z -model. This approach is less stable in shallow areas.

10.4.4 Horizontal viscosity terms

In [subsection 9.3.1](#) the horizontal turbulent fluxes of momentum are described. If the horizontal viscosity terms are simplified, Eqs. [9.26](#) and [9.27](#), there results a Laplace operator along grid lines. The u -momentum equation involves only second-order derivatives of the u -velocity. In this case, this term is integrated fully implicitly by using operator splitting. This procedure is unconditionally stable.

In the momentum equations, the complete Reynolds stress tensor is used, Eqs. [9.23–9.25](#), for the following cases:

- ◊ partial slip at closed boundaries,
- ◊ no slip at the closed boundaries,
- ◊ HLES-model for sub-grid viscosity.

Table 10.1: Time step limitations shallow water solver Delft3D-FLOW

Points per wave period T	$\Delta t \leq \frac{T}{40}$
Accuracy ADI for barotropic mode for complex geometries	$C_f = 2\Delta t \sqrt{gH \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right)} < 4\sqrt{2}$
Explicit advection scheme “Flooding scheme” and for the Z -model.	$\frac{\Delta t u }{\Delta x} < 2$
Stability baroclinic mode internal wave propagation (Z -model only)	$\Delta t \sqrt{\frac{(\rho_{bottom} - \rho_{top})}{\rho_{top}} g \frac{H}{4} \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right)} < 1$
Stability horizontal viscosity term (HLES, partial slip, no slip)	$2\Delta t \nu_H \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) < 1$
Secondary flow (with β defined in Equation 9.155)	$\frac{2\Delta t \beta u }{\min(\Delta x, \Delta y)} \leq 1$

For the complete Reynolds stress tensor the shear stress $\tau_{\xi\eta}$ in the u -momentum equation contains derivatives of the v -velocity. The Reynolds stresses are integrated explicitly, which leads to the following additional stability condition:

$$\Delta t \leq \frac{1}{2\nu_H} \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right)^{-1}. \quad (10.18)$$

In case of a curvilinear grid $\Delta x = \sqrt{G_{\xi\xi}}$ and $\Delta y = \sqrt{G_{\eta\eta}}$.

The stresses due to secondary flow $T_{\xi\xi}, T_{\xi\eta}$ in the depth averaged U -momentum equation [Equation 9.147](#) contain the depth averaged U - and V -velocity components. These stresses are integrated explicitly, which leads to the following stability condition:

$$\frac{2\Delta t \beta |u|}{\min(\Delta x, \Delta y)} \leq 1 \quad (10.19)$$

with β defined in [Equation 9.155](#).

10.4.5 Overview time step limitations

In this section we give an overview of the time step limitations due to stability and accuracy for the time integration of the shallow water equations in Delft3D-FLOW. Let Δx and Δy be horizontal grid sizes. In case of a curvilinear grid $\Delta x = \sqrt{G_{\xi\xi}}$ and $\Delta y = \sqrt{G_{\eta\eta}}$. In [Table 10.1](#) the time step limitations are given for the shallow water code Delft3D-FLOW. Which of the limitations is most restrictive is dependent on the kind of application: length scale, velocity scale, with or without density-coupling etc.

10.5 Spatial discretizations of 3D shallow water equations

10.5.1 Horizontal advection terms

The choice of the spatial discretization of the advective terms has great influence on the accuracy, monotony and efficiency of the computational method. Central differences are often second order accurate, but may give rise to non-physical spurious oscillations, the so-called

“wiggles” ([Gresho and Lee, 1981](#)) in the solution. These wiggles arise in the vicinity of steep gradients of the quantity to be resolved. In shallow water flow these wiggles may also be introduced near closed boundaries and thin dams. On the other hand, first order upwinding is unconditionally wiggle-free or monotone, thus promoting the stability of the solution process, but introduces a truncation error, which has the form of a second-order artificial viscosity term ([Vreugdenhil, 1994](#)). In advection-dominated flows, this artificial viscosity dominates the physical viscosity and the computed solution is much smoother than the correct one. Higher order upwinding is not free from numerical oscillations and introduces fourth-order artificial viscosity. This higher order viscosity suppresses the wiggles without smoothing the solution too much.

In Delft3D-FLOW the following advection schemes are implemented:

- 1 WAQUA-scheme
- 2 Cyclic method
- 3 Flooding-scheme
- 4 Multi directional upwind (Z -model only)

WAQUA-scheme

The first scheme (*WAQUA*) based on [Stelling \(1984\)](#) is an extension of the ADI-method of Leendertse with a special approach for the horizontal advection terms. The normal advection term $u \frac{\partial u}{\partial \xi}$ is discretised with central differences and the cross advection term $v \frac{\partial u}{\partial \eta}$, based on the dissipative reduced phase error scheme. It is a splitting of a third order upwind finite difference scheme for the first derivative into two second order consistent discretizations: a central discretization [Equation 10.20](#) and [Equation 10.21](#), which are successively used in both stages of the ADI-scheme. For the cross advection term the spatial discretization is given by:

Stage 1:

$$\left. \frac{v}{\sqrt{G_{\eta\eta}}} \frac{\partial u}{\partial \eta} \right|_{m,n,k} = \frac{\bar{v}_{m,n,k}^{\xi\eta}}{(\sqrt{G_{\eta\eta}})_{m,n}} \left(\frac{u_{m,n+1,k} - u_{m,n-1,k}}{2\Delta\eta} \right), \quad (10.20)$$

and

Stage 2:

$$\left. \frac{v}{\sqrt{G_{\eta\eta}}} \frac{\partial u}{\partial \eta} \right|_{m,n,k} = \begin{cases} \frac{\bar{v}_{m,n,k}^{\xi\eta}}{(\sqrt{G_{\eta\eta}})_{m,n}} \left(\frac{3u_{m,n,k} - 4u_{m,n-1,k} + u_{m,n-2,k}}{2\Delta\eta} \right), & \bar{v}_{m,n,k}^{\xi\eta} \geq 0 \\ \frac{\bar{v}_{m,n,k}^{\xi\eta}}{(\sqrt{G_{\eta\eta}})_{m,n}} \left(\frac{-3u_{m,n,k} + 4u_{m,n+1,k} - u_{m,n+2,k}}{2\Delta\eta} \right), & \bar{v}_{m,n,k}^{\xi\eta} < 0 \end{cases} \quad (10.21)$$

The advective terms for the v -momentum equation are discretized in a similar way. Spatial oscillations in the velocities are suppressed by means of fourth-order dissipation, ([Stelling, 1984](#); [Stelling and Leendertse, 1992](#)). The *WAQUA* scheme has little dissipation and is used for the accurate prediction of water levels along the Dutch rivers.

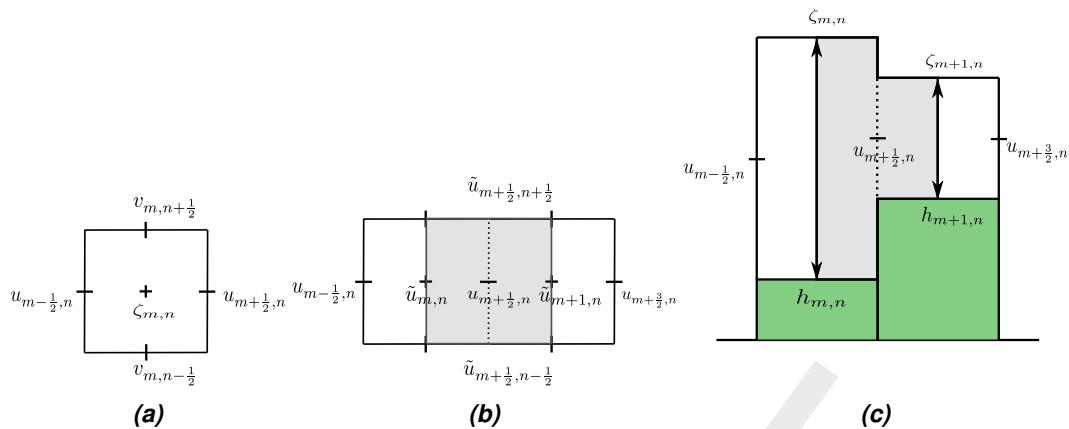


Figure 10.7: (a) Control Volume for mass for the Flooding scheme,
 (b) Control Volume for momentum in horizontal and
 (c) vertical direction for the Flooding scheme

Cyclic-scheme

The second method, (*Cyclic*), which is the default method, is based on the dissipative reduced phase error scheme Eqs. 10.20 and 10.21 for both the normal advection term $u \frac{\partial u}{\partial \xi}$ and the cross advection term $v \frac{\partial u}{\partial \eta}$ (Stelling and Leendertse, 1992).

For WAQUA and the *Cyclic* method, the advection terms are integrated implicitly in the stage of the ADI-method in which the free surface gradient is at the old time level. In the stage in which the free surface gradient is integrated implicitly the advection terms are at the old time level. For stability the vertical terms are integrated implicitly in both stages. The upwind discretization is used in the stage in which both the horizontal advection and vertical viscosity term are integrated implicitly. The resulting linear system of equations has eleven diagonals. The system is solved efficiently by a Red Black Jacobi iterative scheme in the horizontal direction and a double sweep in the vertical direction. For the *Cyclic* method the matrix is diagonally dominant and the iterative scheme converges well. For the WAQUA scheme, due to the central difference scheme for the normal advection, the matrix is *not* diagonally dominant for large time steps (Courant number advection more than one) and this may lead to bad convergence behaviour.

Flooding-scheme

The third scheme, based on Stelling and Duinmeijer (2003) can be applied to rapidly varying depth-averaged flows for instance the inundation of dry land or flow transitions due to large gradients of the bathymetry (obstacles). The scheme can also be used in combination with obstacles represented by only one point on coarse grids. In the *Flooding* scheme the bottom is approximated by a staircase of tiles (DPUOPT=MIN), centred around the water level points, see Figure 10.7a. In combination with the local invalidity of the hydrostatic pressure assumption, conservation properties become crucial. In flow expansions a numerical approximation is applied that is consistent with conservation of momentum and in flow contractions a numerical approximation is applied that is consistent with the Bernoulli equation. For sufficiently smooth conditions, and a fine grid size, both approximations converge to the same solution. The local order of consistency depends on the solution. The approximations are second-order, but the accuracy reduces to first order near extreme values by the use of the so-called *Minmod slope limiter* (Stelling and Duinmeijer, 2003). The limiter prevents the generation of wiggles. *The conservation of momentum in the present implementation has been derived only for a Cartesian rectangular grid and depth averaged velocities.*

We describe the discretizations for positive flow direction. For negative flow direction the discretizations are defined accordingly. The momentum conservative approximation for the normal advection term $u \frac{\partial u}{\partial \xi}$ derived for a Control Volume around a velocity point is given by (using $q = Hu$):

$$\frac{u}{\sqrt{G_{\xi\xi}}} \frac{\partial u}{\partial \xi} \Big|_{m+\frac{1}{2},n} = \frac{1}{H \sqrt{G_{\xi\xi}}} \left(\frac{\partial qu}{\partial \xi} - u \frac{\partial q}{\partial \xi} \right) \Big|_{m+\frac{1}{2},n} \quad (10.22)$$

$$= \frac{2}{(H_{m,n} + H_{m+1,n}) (\sqrt{G_{\xi\xi}})_{m+\frac{1}{2},n}} \times \\ \left\{ \left(\frac{\bar{q}_{m+1,n}^\xi \tilde{u}_{m+1,n} - \bar{q}_{m,n}^\xi \tilde{u}_{m,n}}{\Delta \xi} \right) - u_{m+\frac{1}{2},n} \left(\frac{\bar{q}_{m+1,n}^\xi - \bar{q}_{m,n}^\xi}{\Delta \xi} \right) \right\} \quad (10.23)$$

$$\bar{q}_{m,n}^\xi = \frac{q_{m+\frac{1}{2},n} + q_{m-\frac{1}{2},n}}{2} \quad (10.24)$$

$$\tilde{u}_{m,n} = \begin{cases} u_{m-\frac{1}{2},n} + \frac{1}{2}\psi(r_u) (u_{m-\frac{1}{2},n} - u_{m-\frac{3}{2},n}) & \bar{q}_{m,n}^\xi \geq 0 \\ u_{m+\frac{1}{2},n} + \frac{1}{2}\psi(r_u) (u_{m+\frac{1}{2},n} - u_{m+\frac{3}{2},n}) & \bar{q}_{m,n}^\xi < 0 \end{cases} \quad (10.25)$$

$$\psi(r_u) = \max(0, \min(r_u, 1)) \quad (10.26)$$

$$r_u = \frac{u_{m+\frac{1}{2},n} - u_{m-\frac{1}{2},n}}{u_{m-\frac{1}{2},n} - u_{m-\frac{3}{2},n}} \quad (10.27)$$

For negative flow direction, the limiter is defined accordingly.

$$q|_{m+\frac{1}{2},n} = u|_{m+\frac{1}{2},n} \frac{H_{m,n} + H_{m+1,n}}{2} \quad (10.28)$$

The momentum conservative approximation for the cross advection term $v \frac{\partial u}{\partial \eta}$ is given by (using $p = Hv$):

$$\frac{v}{\sqrt{G_{\eta\eta}}} \frac{\partial u}{\partial \eta} \Big|_{m+\frac{1}{2},n} = \frac{1}{H \sqrt{G_{\eta\eta}}} \left(\frac{\partial pu}{\partial \eta} - u \frac{\partial p}{\partial \eta} \right) \Big|_{m+\frac{1}{2},n} \quad (10.29)$$

$$= \frac{2}{(H_{m,n} + H_{m+1,n}) (\sqrt{G_{\eta\eta}})_{m+\frac{1}{2},n}} \times \\ \left\{ \left(\frac{\bar{p}_{m+\frac{1}{2},n+\frac{1}{2}}^\xi \tilde{u}_{m+\frac{1}{2},n+\frac{1}{2}} - \bar{p}_{m+\frac{1}{2},n-\frac{1}{2}}^\xi \tilde{u}_{m+\frac{1}{2},n-\frac{1}{2}}}{\Delta \eta} \right) - \right. \\ \left. u_{m+\frac{1}{2},n} \left(\frac{\bar{p}_{m+\frac{1}{2},n+\frac{1}{2}}^\xi - \bar{p}_{m+\frac{1}{2},n-\frac{1}{2}}^\xi}{\Delta \eta} \right) \right\} \quad (10.30)$$

$$\bar{p}_{m+\frac{1}{2},n-\frac{1}{2}}^{\xi} = \frac{p_{m,n-\frac{1}{2}} + p_{m+1,n-\frac{1}{2}}}{2} \quad (10.31)$$

$$\tilde{u}_{m+\frac{1}{2},n-\frac{1}{2}} = \begin{cases} u_{m+\frac{1}{2},n-1} + \frac{1}{2}\psi(r_u) \left(u_{m+\frac{1}{2},n-1} - u_{m+\frac{1}{2},n-2} \right) & \bar{p}_{m+\frac{1}{2},n-\frac{1}{2}}^{\xi} \geq 0 \\ u_{m+\frac{1}{2},n+1} + \frac{1}{2}\psi(r_u) \left(u_{m+\frac{1}{2},n+1} - u_{m+\frac{1}{2},n+2} \right) & \bar{p}_{m+\frac{1}{2},n-\frac{1}{2}}^{\xi} < 0 \end{cases} \quad (10.32)$$

$$\psi(r_u) = \max(0, \min(r_u, 1)) \quad (10.33)$$

$$r_u = \frac{u_{m+\frac{1}{2},n} - u_{m+\frac{1}{2},n-1}}{u_{m+\frac{1}{2},n-1} - u_{m+\frac{1}{2},n-2}} \quad (10.34)$$

For negative flow direction, the limiter is defined accordingly.

$$p|_{m+\frac{1}{2},n} = \bar{v}^{\xi\eta}|_{m+\frac{1}{2},n} \frac{H_{m,n} + H_{m+1,n}}{2} \quad (10.35)$$

From the momentum conservative formulation a so-called energy head conservative discretization for the same Control Volume, see Figures 10.7b and 10.7c, can be derived under steady state conditions (constant discharge q) in 1D (along a streamline in 2D), see [Equation 10.36](#).

$$\begin{aligned} \frac{q^2}{2\sqrt{G_{\xi\xi}}} \left[\frac{1}{(H_{m+1,n})^2} - \frac{1}{(H_{m,n})^2} \right] &= \\ \lambda \frac{2q^2}{(H_{m,n} + H_{m+1,n}) \sqrt{G_{\xi\xi}}} \left[\frac{1}{H_{m+1,n}} - \frac{1}{H_{m,n}} \right] & \text{Momentum Conservation} \end{aligned} \quad (10.36)$$

$$\lambda = \begin{cases} 1 & q > 0 \wedge H_{m,n} < H_{m+1,n} \\ \frac{(H_{m,n} + H_{m+1,n})^2}{4H_{m,n}H_{m+1,n}} & q > 0 \wedge H_{m,n} > H_{m+1,n} \end{cases} \quad (10.37)$$

The energy conservative discretization is applied for contractions in both directions. For 2D flow the direction of the grid lines do not always coincide with streamlines and this will generate small head losses.

Near the boundaries the higher order discretization stencils for the advection terms contain grid points on or across the boundary. To avoid an artificial boundary layer or instabilities, the discretizations are reduced to lower order discretizations with smaller stencils. [Stelling \(1984\)](#) developed the numerical boundary treatment implemented in Delft3D-FLOW.

Multi directional upwind (Z -model only)

The fourth scheme, which is only available in the Z -model is a *multi-directional upwind scheme*. It is an extension to two dimensions of the first-order upwind method. Both an explicit and an implicit variant of the scheme is available. It is a positive and monotone scheme. For the explicit variant there is the Courant number stability constraint. The method is described by [Bijvelds \(2001\)](#).

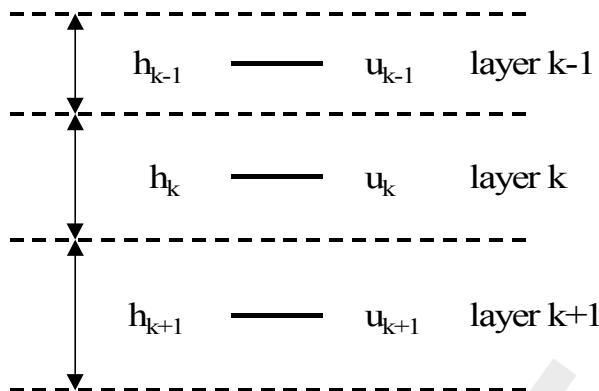


Figure 10.8: Layer numbering in σ -model

10.5.2 Vertical advection term

The horizontal velocities of adjacent vertical layers are coupled by the vertical advection and the vertical viscosity term. The σ co-ordinate system can lead to very thin layers in shallow areas. To prevent instabilities, a fully implicit time integration is used for the vertical exchange terms. This results in tridiagonal systems of equations in the vertical.

In a shallow water model the horizontal length scale is much larger than the vertical length scale. In the vertical direction the eddy viscosity term dominates the advection term. Only for stratified flows where the turbulent exchange is reduced, advection may be dominant. For the space discretization of the vertical advection term, a second order central difference is used:

$$\frac{\omega}{H} \frac{\partial u}{\partial \sigma} \Big|_{m,n,k} = \bar{\omega}_{m,n,k}^{\xi\sigma} \left(\frac{u_{m,n,k-1} - u_{m,n,k+1}}{\frac{1}{2}h_{m,n,k-1} + h_{m,n,k} + \frac{1}{2}h_{m,n,k+1}} \right), \quad (10.38)$$

where $h_{m,n,k}$ denotes the thickness of the computational layer with index k defined by $h_{m,n,k} = \Delta\sigma_k H_{m,n}$ and H the total water depth or $h_{m,n,k} = \Delta z_{m,n,k}$ in the Z -model.

10.5.3 Viscosity terms

The approximation of the vertical viscosity terms are based on central differences. The vertical viscosity term in the u -equations is discretized as:

$$\begin{aligned} \frac{1}{H^2} \frac{\partial}{\partial \sigma} \left(\nu_V \frac{\partial u}{\partial \sigma} \right) \Big|_{m,n,k} &= \frac{\nu_V|_{m,n,k-1}}{h_{m,n,k}} \left(\frac{u_{m,n,k-1} - u_{m,n,k}}{\frac{1}{2}(h_{m,n,k-1} + h_{m,n,k})} \right) + \\ &\quad - \frac{\nu_V|_{m,n,k}}{h_{m,n,k}} \left(\frac{u_{m,n,k} - u_{m,n,k+1}}{\frac{1}{2}(h_{m,n,k} + h_{m,n,k+1})} \right). \end{aligned} \quad (10.39)$$

The vertical eddy viscosity is computed at the layer interface, with $h_{m,n,k} = \Delta\sigma_k H_{m,n}$ or $h_{m,n,k} = \Delta z_{m,n,k}$ in the Z -model.

10.6 Solution method for the transport equation

A robust and accurate solver for scalar transport has to satisfy the following demands:

- ◊ Mass conservation.
- ◊ Monotony (positive solution).
- ◊ Accuracy (at least second order consistency).
- ◊ Suitable for both time-dependent and steady state problems.

- ◊ Computationally efficient.

An explicit time integration of the scalar advection-diffusion equation on a rectangular grid has a time step limitation based on the Courant number for advection:

$$C_{adv} = \max \left(\frac{u\Delta t}{\Delta x}, \frac{v\Delta t}{\Delta y} \right) \leq 1, \quad (10.40)$$

with Δx and Δy the grid spaces in the physical space.

Explicit integration of the horizontal diffusion term yields an upper limit of:

$$\Delta t \leq \frac{1}{D_H} \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right)^{-1}. \quad (10.41)$$

For the 3D transport equation the scalar concentrations are coupled in the vertical direction by the vertical advection and diffusion term. An explicit time integration of the vertical exchange terms on the σ -co-ordinate grid would lead to very severe time step limitations:

$$\Delta t \leq \frac{(\Delta \sigma H)^2}{2D_V}, \quad (10.42)$$

$$\Delta t \leq \frac{\Delta \sigma H}{\omega}. \quad (10.43)$$

Therefore in the vertical direction a fully implicit time integration method is applied, which is first order in time and leads to tridiagonal systems of equations. The vertical coupling of the discretized transport equations is removed by a double sweep algorithm.

To ensure that the total mass is conserved the transport equation in Delft3D-FLOW is discretized with a mass conserving Finite Volume approach (flux form). For the spatial discretization of the horizontal advection terms, two options are available in Delft3D-FLOW. The first (and default) option is a finite difference scheme that conserves large gradients without generating spurious oscillations and is based on the ADI-method. This scheme is denoted as the *Cyclic* method, see [Stelling and Leendertse \(1992\)](#). The *Cyclic* method of Stelling and Leendertse is based on an implicit time integration of both advection and diffusion and does not impose a time step restriction.

The second option is an explicit scheme that belongs to the class of monotonic schemes: the so-called *Van Leer-2* scheme ([Van Leer, 1974](#)). The *Van Leer-2* scheme is slightly less accurate than the scheme of Stelling and Leendertse. It combines two numerical schemes, namely a first order upwind scheme and the second order upwind scheme developed by Fromm. In case of a local minimum or maximum the first order upwind scheme is applied, whereas the upwind scheme of Fromm is used in case of a smooth numerical solution. The time integration of the *Van Leer-2* scheme is explicit and therefore a CFL condition for advection and diffusion must be fulfilled. Owing to the explicit time integration the *Van Leer-2* scheme requires per time step less computation time than the *Cyclic* method of Stelling and Leendertse. However, the *Van Leer-2* scheme produces a more diffusive numerical solution, because of the fact that a first order upwind discretization is applied in case of a local maximum or minimum. The transport scheme for the *Z*-model is described by [Bijvelds \(2001\)](#).

The transport equation is coupled with the momentum equations by the baroclinic pressure term, see Eqs. [9.15](#) and [9.16](#) and subsection [9.3.4](#). The temporal variations in salinity are

slow compared to the variations in the flow and therefore the baroclinic term in the momentum equations is treated explicitly, introducing a stability condition for internal gravity waves (baroclinic mode), see subsection 10.4.5. The coupling with the flow is weak and in Delft3D-FLOW the transport equation is solved independently of the flow for each half time step.

For the time integration of the horizontal diffusion term along σ -planes the Crank-Nicholson method is applied. If the spatial discretization of the horizontal diffusion term is based on a Cartesian grid using the Finite Volume approach of Stelling and Van Kester (1994) the integration is explicit.

Source terms are integrated explicitly. In order to avoid negative concentrations and instabilities, sink terms are integrated fully implicit.

10.6.1 Cyclic method

To keep the numerical diffusion as small as possible the horizontal advection terms in the scalar transport equation are approximated by the sum of a third-order upwind scheme and a second-order central scheme. A second order central scheme is applied for the approximation of the vertical advection term.

For the *Cyclic* method the time integration follows the ADI-method for the continuity equation. In the first stage all space derivatives with respect to ξ are taken implicitly and all derivatives in the η -direction are taken explicitly. In the second stage the directions for explicit and implicit integration are interchanged. If the upwind discretization is used in the stage in which both the horizontal advection and vertical viscosity term are integrated implicitly, the resulting linear system of equations has thirteen diagonals but the matrix is diagonally dominant. Thus, the system can be solved effectively by a Red Black Jacobi iterative scheme in the horizontal direction and a double sweep in the vertical direction.

For the *Cyclic* method the upwind discretization of the horizontal advective fluxes in ξ -direction is described by:

$$\frac{1}{\sqrt{G_{\xi\xi}}} \frac{\partial (huc\sqrt{G_{\eta\eta}})}{\partial \xi} \Big|_{m,n,k} = \frac{1}{\sqrt{G_{\xi\xi}}|_{m,n}} \xi \eta \left(\frac{F_{m+\frac{1}{2},n,k} - F_{m-\frac{1}{2},n,k}}{\Delta \xi} \right). \quad (10.44)$$

For the scalar flux $F_{m+\frac{1}{2},n,k}$ at the U -velocity point the interpolation is given by:

$$F_{m+\frac{1}{2},n,k} = u_{m+\frac{1}{2},n,k} h_{m+\frac{1}{2},n,k} \sqrt{G_{\eta\eta}} \Big|_{m+\frac{1}{2},n} \begin{cases} \frac{10c_{m,n,k} - 5c_{m-1,n,k} + c_{m-2,n,k}}{6\Delta\xi}, & u_{m+\frac{1}{2},n,k} \geq 0 \\ \frac{10c_{m+1,n,k} - 5c_{m+2,n,k} + c_{m+3,n,k}}{6\Delta\xi}, & u_{m+\frac{1}{2},n,k} < 0 \end{cases} \quad (10.45)$$

In the first stage in η -direction a central scheme is applied:

$$\frac{1}{\sqrt{G_{\eta\eta}}} \frac{\partial (hvc\sqrt{G_{\xi\xi}})}{\partial \eta} \Big|_{m,n,k} = \frac{1}{\sqrt{G_{\eta\eta}}|_{m,n}} \xi \eta \left(\frac{G_{m,n+\frac{1}{2},k} - G_{m,n-\frac{1}{2},k}}{\Delta \eta} \right), \quad (10.46)$$

with the scalar flux $G_{m+\frac{1}{2},n,k}$ at the V -velocity point determined by:

$$G_{m+\frac{1}{2},n,k} = v_{m+\frac{1}{2},n,k} h_{m+\frac{1}{2},n,k} \sqrt{G_{\xi\xi}} \Big|_{m,n+\frac{1}{2}} \frac{c_{m,n,k} + c_{m,n+1,k}}{2}. \quad (10.47)$$

Near open and closed boundaries the approximations for the fluxes are reduced to lower order.

10.6.2 Van Leer-2 scheme

For the second option in Delft3D-FLOW, namely the *Van Leer-2* scheme, the interpolation formula for the horizontal fluxes is given by:

$$F_{m+\frac{1}{2},n,k} = u_{m+\frac{1}{2},n,k} h_{m+\frac{1}{2},n,k} \sqrt{G_{\eta\eta}} \Big|_{m+\frac{1}{2},n} \times \begin{cases} c_{m,n,k} + \alpha (1 - CFL_{adv-u}) (c_{m,n,k} - c_{m-1,n,k}) \frac{c_{m+1,n,k} - c_{m,n,k}}{c_{m+1,n,k} - c_{m-1,n,k}}, \\ \text{when } u_{m+\frac{1}{2},n,k} \geq 0, \\ c_{m+1,n,k} + \alpha (1 + CFL_{adv-u}) (c_{m+1,n,k} - c_{m+2,n,k}) \frac{c_{m,n,k} - c_{m+1,n,k}}{c_{m,n,k} - c_{m+2,n,k}}, \\ \text{when } u_{m+\frac{1}{2},n,k} < 0, \end{cases} \quad (10.48)$$

with:

$$CFL_{adv-u} = \frac{\Delta t |u|}{\Delta x} \quad (10.49)$$

and:

$$\alpha = \begin{cases} 0, & \left| \frac{c_{m+1,n,k} - 2c_{m,n,k} + c_{m-1,n,k}}{c_{m+1,n,k} - c_{m-1,n,k}} \right| > 1, \text{ (local max. or min.)}, \\ 1, & \left| \frac{c_{m+1,n,k} - 2c_{m,n,k} + c_{m-1,n,k}}{c_{m+1,n,k} - c_{m-1,n,k}} \right| \leq 1, \text{ (monotone)}. \end{cases} \quad (10.50)$$

In η -direction a similar discretization is applied. Eqs. 10.48 to 10.50 consist of a diffusive first-order upwind term and a higher order anti-diffusive term.

The time integration of the *Van Leer-2* scheme is explicit. The Courant number for advection should be smaller than 1.

10.6.3 Vertical advection

In the vertical direction the fluxes are discretized with a central scheme:

$$\frac{1}{H} \frac{\partial \omega c}{\partial \sigma} \Big|_{m,n,k} = \frac{1}{H_{m,n}} \left(\frac{F_{m,n,k-1} - F_{m,n,k}}{\Delta \sigma} \right), \quad (10.51)$$

with the flux $F_{m,n,k}$ determined by:

$$F_{m,n,k} = \omega_{m,n,k} \frac{c_{m,n,k} + c_{m,n,k-1}}{2}. \quad (10.52)$$

The time integration in the vertical direction is fully implicit. The vertical advection leads to a tridiagonal system in the vertical. If the flow in the vertical is advection dominated, due to vertical stratification in combination with upwelling or downwelling near a closed boundary or a sill, a discharge of buoyant water, the central differences in the vertical may give rise to non-physical spurious oscillations. The solution has an unphysical maximum or minimum scalar concentration (overshoot or undershoot).

10.6.4 Forester filter

It is well-known that second or higher order advective difference methods (such as the *Cyclic* method) on coarse grids may exhibit non-physical oscillations near regions of steep gradients. The difference operators do not guarantee positive solutions and consequently negative concentrations may occur. For depth-averaged simulations the *Van Leer-2* scheme is strictly positive. In 3D, for both transport schemes, the central differences in the vertical may give rise to non-physical spurious oscillations, so-called “wiggles” ([Gresho and Lee, 1981](#)) in the solution when the vertical grid Péclet number $\text{Pe}_{\Delta z} = \frac{|w|\Delta z}{D_V} \leq 2$. These wiggles may arise in the vicinity of steep gradients of the quantity to be resolved. In shallow water scalar transport these wiggles may be introduced in stratified areas near closed boundaries and steep bottom slopes. Positive solutions are not guaranteed. In case of negative concentrations an iterative filter procedure based on local diffusion along σ -lines followed by a vertical filter is started in order to remove the negative values. The filtering technique in this procedure is the so-called Forester filter ([Forester, 1979](#)), a non-linear approach which removes the computational noise without inflicting significant amplitude losses in sharply peaked solutions.

If concentration $c_{m,n,k}$ is negative, then the iterative, mass conservative filtering process is described (for the sake of simplicity only in one direction, namely the ξ -direction) by:

$$\begin{aligned} c_{m,n,k}^{p+1} = c_{m,n,k}^p + & \frac{c_{m+1,n,k}^p - c_{m,n,k}^p}{4} \min \left(1, \frac{V_{m+1,n,k}}{V_{m,n,k}} \right) + \\ & + \frac{c_{m-1,n,k}^p - c_{m,n,k}^p}{4} \min \left(1, \frac{V_{m-1,n,k}}{V_{m,n,k}} \right), \end{aligned} \quad (10.53)$$

with $V_{m,n,k}$ denoting the volume of cell (m, n, k) .

This filter is applied only in grid cells with a negative concentration. The superscript p denotes the iteration number. The filter smoothes the solution and reduces the local minima (negative concentrations). [Equation 10.53](#), can be interpreted as an approximation of the following advection-diffusion equation:

$$\frac{\partial c}{\partial t} = \frac{\alpha - \beta}{4} \frac{\Delta x}{\Delta t} \frac{\partial c}{\partial x} + \frac{\alpha + \beta}{4} \frac{\Delta x^2}{\Delta t} \frac{\partial^2 c}{\partial x^2} + (\text{higher order terms}), \quad (10.54)$$

with:

$$\alpha = \min \left(1, \frac{V_{m+1,n,k}}{V_{m,n,k}} \right), \quad (10.55)$$

and:

$$\beta = \min \left(1, \frac{V_{m-1,n,k}}{V_{m,n,k}} \right). \quad (10.56)$$

The Forester filter introduces an artificial advection and diffusion. The numerical diffusion coefficient of the horizontal filter is:

$$D_{num} = \frac{\alpha + \beta}{4} \frac{\Delta x^2}{\Delta t} \leq \frac{\Delta x^2}{2\Delta t}. \quad (10.57)$$

Thus the filter introduces numerical diffusion but only locally. Maximal 100 iterations are carried out. If there is still a grid cell with a negative concentration after 100 iterations, then a

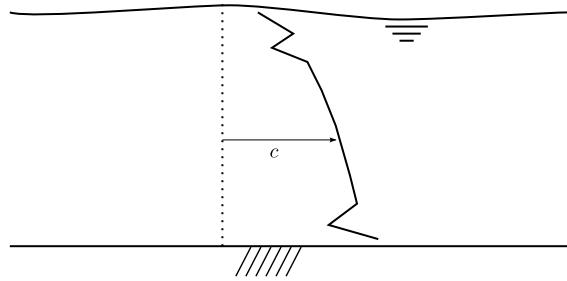


Figure 10.9: Illustration of wiggles in vertical direction

warning is generated. To further understand the influence of the Forester filter, we rewrite [Equation 10.53](#) as:

$$c_{m,n,k}^{p+1} = \left(1 - \frac{\alpha + \beta}{4}\right) c_{m,n,k}^p + \frac{\alpha}{4} c_{m+1,n,k}^p + \frac{\beta}{4} c_{m-1,n,k}^p. \quad (10.58)$$

As both $\alpha \leq 1$ and $\beta \leq 1$, all coefficients of [Equation 10.58](#) are positive. Consequently, a positive concentration will remain positive, i.e. it will not introduce negative concentrations irrespective the steepness of the concentration gradients. A negative concentration surrounded by positive concentrations, usually the result of ill represented steep gradients (wiggles), will be less negative after one iteration and is effectively removed after several iterations by adding enough (local) diffusion to force the concentration to become positive.

In the vertical, these wiggles may lead to unrealistic vertical profiles of temperature and/or salinity, see [Figure 10.9](#).

Local maxima and minima in temperature or salinity in the vertical direction, *generated by the computational method* may give physically unstable density profiles and can also better be removed by a numerical filter than by turbulent vertical mixing. We should be sure that it is a wiggle generated by the numerical method and not by physical processes like heating or cooling through the free surface or the discharge of water somewhere in the vertical. A similar filtering technique as in the horizontal direction is applied for points with a local maximum or minimum in the vertical:

local maximum:

$$c_{m,n,k} > \max(c_{m,n,k+1}, c_{m,n,k-1}) + \varepsilon, \quad \text{and} \quad \text{Pe}_{\Delta z} = \frac{|w| \Delta z}{D_V} \leq 2$$

local minimum:

$$c_{m,n,k} < \min(c_{m,n,k+1}, c_{m,n,k-1}) + \varepsilon, \quad \text{and} \quad \text{Pe}_{\Delta z} = \frac{|w| \Delta z}{D_V} \leq 2$$

the filter is applied, with $\varepsilon = 10^{-3}$. The numerical diffusion coefficient of the vertical filter is:

$$D_{num} = \frac{\Delta z^2}{2\Delta t}. \quad (10.59)$$

Smooth but unstable vertical density profiles of salinity and temperature in the vertical direction, can sometimes also better be vertically mixed by a numerical filter technique than by

the turbulence model. E.g. for temperature, the algorithm is given by:

$$\begin{aligned}
 & \text{If } T_{m,n,k} > T_{m,n,k-1} + \varepsilon \text{ Then} \\
 & \quad T_{m,n,k} = T_{m,n,k} - \min(\Delta z_k, \Delta z_{k-1}) \frac{(T_{m,n,k} - T_{m,n,k-1})}{2\Delta z_k} \\
 & \quad T_{m,n,k-1} = T_{m,n,k-1} + \min(\Delta z_k, \Delta z_{k-1}) \frac{(T_{m,n,k} - T_{m,n,k-1})}{2\Delta z_{k-1}} \\
 & \text{Endif}
 \end{aligned} \tag{10.60}$$

with $\varepsilon = 10^{-6}$.

If both the horizontal and vertical filter are switched on, then first the filter in the horizontal direction is carried out. The maximum number of filter steps in the horizontal direction is 100. This is followed by applying the filter in the vertical direction and hereby minimising the additional vertical mixing. The maximum number of filter steps in the vertical direction is 1000. If the maximum number of filter steps is exceeded, a warning is written in the diagnostic file.

Remark:

- ◊ The vertical Forester filter does not affect sediments and other constituents. When activated, the filter only smooths salinity and temperature.



10.7 Numerical implementation of the turbulence models

The turbulence closure models in Delft3D-FLOW are all based on the eddy viscosity concept; see [section 9.5](#). The eddy viscosity is always based on information of the previous half time step. The transport equations of turbulent kinetic energy k , [Equation 9.127](#), and dissipation rate ε , [Equation 9.128](#) are solved in a *non-conservative* form. For turbulent boundary flows local production, dissipation, and vertical diffusion are the dominant processes. On the staggered grid, the turbulent quantities k , ε and the eddy viscosity ν_V are positioned at the layer interfaces in the centre of the computational cell. This choice makes it possible to discretize the vertical gradients in the production term and buoyancy term accurately and to implement the vertical boundary conditions at the bed and the free surface. First order upwind differencing for the advection provides positive solutions. For more details we refer to [Uittenbogaard et al. \(1992\)](#) and [Van Kester \(1994\)](#).

10.8 Drying and flooding

Estuaries and coastal embayments contain large, shallow, and relatively flat areas separated and interleaved by deeper channels and creeks. When water levels are high, the entire area is water covered but as tide falls, the shallow areas are exposed, and ultimately the flow is confined only to the deeper channels. The dry tidal flats may occupy a substantial fraction of the total surface area. The accurate reproduction of covering or uncovering of the tidal flats is an important feature of numerical flow models based on the shallow water equations.

Many rivers have compound channels, consisting of a main channel which always carries flow (the summer-bed) and one or two flood plains which only carry flow during extreme river discharges (the winter-bed). The summer bed is surrounded by low dikes, which will overtop when the river discharge increases. The winter-bed is surrounded by much higher dikes, which are designed to protect the polders against extreme river discharges. The flooding of the flood plains increases the drainage capacity of the river and reduces the local water level gradients.

In a numerical model, the process of drying and flooding is represented by removing grid points from the flow domain that become “dry” when the tide falls and by adding grid points

that become “wet” when the tide rises. Drying and flooding is constrained to follow the sides of grid cells. In this section, we specify the algorithms which have been used to determine the moment when a grid cell (water level point) or cell boundary (velocity point) becomes dry or wet. Drying and flooding gives a discontinuous movement of the closed boundaries and may generate small oscillations in water levels and velocities. The oscillations introduced by the drying and flooding algorithm are small if the grid sizes are small and the bottom has smooth gradients.

The crucial items in a wetting and drying algorithm are:

- ◊ The way in which the bottom depth is defined at a water level point.
- ◊ The way in which the water level is defined at velocity points.
- ◊ Criteria for setting a velocity and/or water level points wet or dry.

In the following subsections, these three items will be discussed.

The standard drying and flooding algorithm in Delft3D-FLOW is efficient and accurate for coastal regions, tidal inlets, estuaries, and rivers. In combination with the *Flooding* scheme for advection in the momentum equation, the algorithm is also effective and accurate for rapidly varying flows with large water level gradients because of the presence of hydraulic jumps or the occurrence of bores as a result of dam breaks.

10.8.1 Bottom depth at water level points

Delft3D-FLOW uses a staggered grid; see [Figure 10.4](#). At input the bottom depth can be specified by you at the vertices of a computational cell, the so-called depth points, or in the cell centre, the so-called water level point (`DPSOPT=DP`). The `DPSOPT=DP`-option implies that the position of the depth points is shifted to the water level points. You should consider this interpretation when generating the depth values with e.g. `QUICKIN`.

To determine the total water depth at water level points, a bottom depth in the cell centre of the Control Volume is required. The bottom depth in a water level point $d_{m,n}^\zeta$ is not uniquely defined; see [Figure 10.10](#). The algorithm used to determine this depth value from the four surrounding depth points depends on the choice made by you. In older versions of Delft3D-FLOW, three options were available: MEAN, MAX and MIN (through the value of the parameter `DRYFLP`). Recently a new flag `DPSOPT` has been introduced with the following extended options: MEAN, MAX, MIN and DP. It replaces partly the function of the old input parameter `DRYFLP`.

The algorithms to determine the depth in a water level point from the four surrounding depth points are given by:

$$\text{MAX-option: } d_{m,n}^\zeta = \max(d_{m,n}, d_{m-1,n}, d_{m,n-1}, d_{m-1,n-1})$$

$$\text{MEAN-option: } d_{m,n}^\zeta = 0.25(d_{m,n} + d_{m-1,n} + d_{m,n-1} + d_{m-1,n-1})$$

$$\text{MIN-option: } d_{m,n}^\zeta = \min(d_{m,n}, d_{m-1,n}, d_{m,n-1}, d_{m-1,n-1})$$

$$\text{DP-option: } d_{m,n}^\zeta = d_{m,n}$$

With the introduction of `DPSOPT` the value of the keyword in the input file that is related to the selection of additional drying and flooding procedure at a water level point, `DRYFLP`, is now restricted to YES or NO. YES implies that an additional drying and flooding check is required

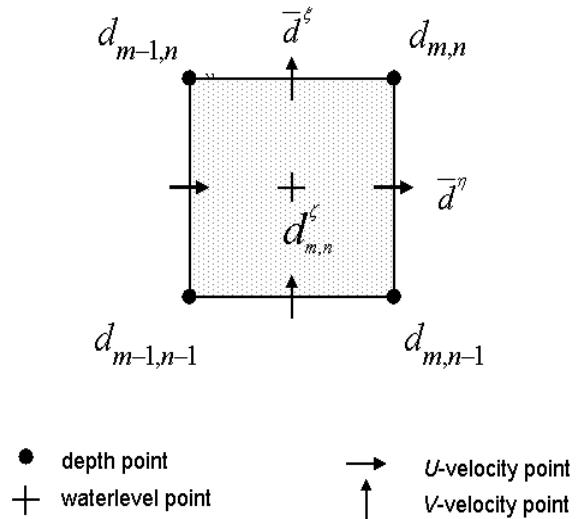


Figure 10.10: Definition bottom depth on FLOW grid

based on the evaluation of the value of the total water depth. If its value drops below the user specified threshold than all four cell interfaces are set to dry (velocities are set to zeroes). DRYFLP=NO implies that the drying and flooding check is only to be based on the individual test of the depth values at the cell interfaces.

For the time being, if unspecified, DPSOPT value is determined automatically from DRYFLP by the program (default value of DPSOPT is equal to DRYFLP and subsequently DRYFLP is then set according to its original value).

The retention volume of a dry cell is the cell area times the difference between the water level and the bottom depth at the cell centre. For the combination of flow computations with transport of dissolved substances, the control volume/retention volume should be positive. In Figure 10.11 an example is shown in which the water level is below the bottom in the cell centre ($d_{m,n}^\zeta$), determined on basis of the average depth, is below the bottom, while some of the adjacent velocity points still have a positive flow through height.

Therefore after solving the coupled system of continuity equation and momentum equation there is a drying check applied to the water level points, see subsection 10.8.3. If the total water depth in a water level point is negative:

$$H_{m,n}^\zeta = d_{m,n}^\zeta + \zeta_{m,n} \leq 0, \quad (10.61)$$

the continuity cell is taken out of the computation and the half time step is repeated. In case of steep bottom slopes, the MEAN option may lead to flooding of velocity points and afterwards drying due to a negative control volume, increasing the computational time. The use of MAX (default) is recommended. The algorithms for flooding and drying of tidal flats have been extensively described by Stelling *et al.* (1986). However, that article does not include the MAX option. It has been found that the MAX procedure is more favourable and will produce a more smooth solution than the options described by Stelling *et al.* (1986).

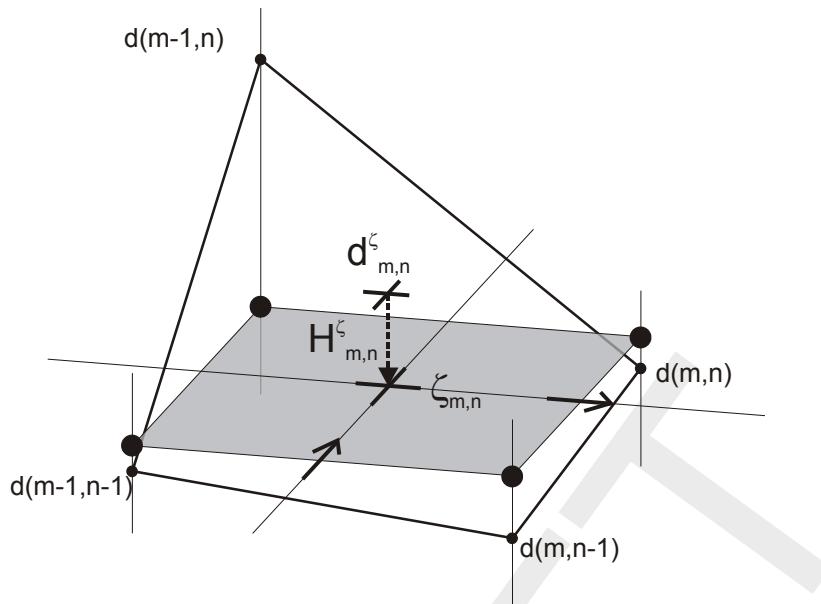


Figure 10.11: Negative control volume with two positive flow-through heights, MEAN-option

The initial water level at a dry cell is determined by the depth at a water level point:

$$\zeta_{m,n} = -d_{m,n}^\xi. \quad (10.62)$$

10.8.2 Total water depth at velocity points

Due to the staggered grid applied in Delft3D-FLOW, the total water depth at a velocity point for the computation of the discharge through a cell face is not uniquely defined. Usually (default option, marginal depth $DCO=-999$) it is determined by the arithmetic average of the depth specified in the vertices of the cell face (side) plus the average of the water levels computed in the cell centres at each side of that cell face:

$$H_{m,n}^U = \bar{d}^\eta + \bar{\zeta}^\xi, \quad (10.63)$$

For the depth values at the cell interfaces, \bar{d}^η , you can now choose from the following three options by setting an appropriate value for the input parameter $DPUOPT$:

$$DPUOPT = \text{MEAN (default option)}: \bar{d}^\eta = \frac{(d_{m,n} + d_{m,n-1})}{2},$$

$$DPUOPT = \text{MIN(imum)}: \bar{d}^\eta = \min(d_{m,n}^\xi, d_{m+1,n}^\xi),$$

$$DPUOPT = \text{UPW(ind)}: \bar{d}^\eta = \begin{cases} d_{m,n}^\xi & \text{if } U_{m,n} > 0 \\ d_{m+1,n}^\xi & \text{if } U_{m,n} < 0 \\ \min(d_{m,n}^\xi, d_{m+1,n}^\xi) & \text{if } U_{m,n} = 0 \end{cases}$$



Remark:

◊ DPSOPT is DP and DPUOPT is MEAN should not be used together.

For the water level at the cell interfaces $\bar{\zeta}^\xi$ as default the mean value is used: $\bar{\zeta}^\xi = \frac{(\zeta_{m,n} + \zeta_{m+1,n})}{2}$.

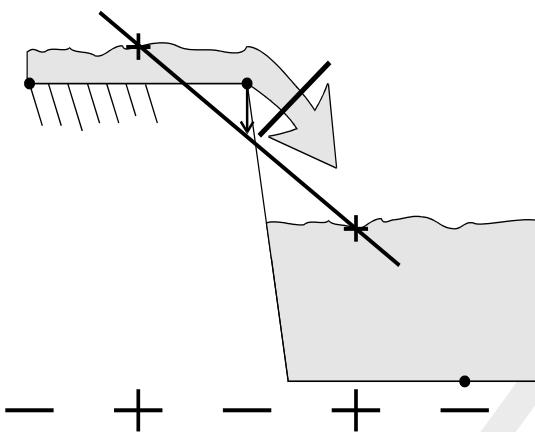


Figure 10.12: Drying of a tidal flat; averaging approach. The flow-through height is based on the average water level, see [Equation 10.63](#), the velocity point is set dry.

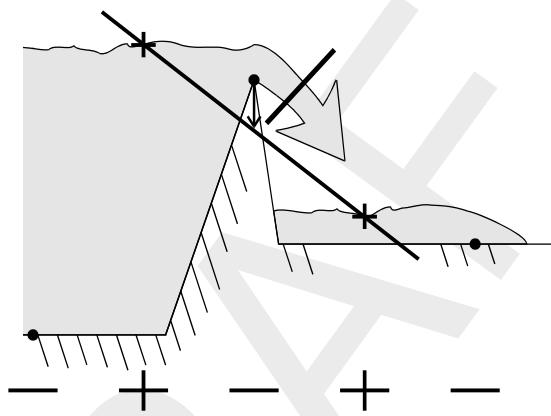


Figure 10.13: Overtopping of a river bank (weir); averaging approach. The flow-through height is based on the average water level, see [Equation 10.63](#), the velocity point is set dry.

In the neighbourhood of steep bottom gradients, use of the average water level to compute the total water depth at a velocity point, may lead to an inaccurate determination of the flow-through height. The velocity point is set dry too early, see [Figure 10.12](#) and [Figure 10.13](#). A large volume of water is left on the tidal flat, increasing artificially the storage capacity of the wet area.

It is also possible to determine the water levels at the cell faces with a so-called upwind approach. This approach was already suggested by [Stelling \(1984\)](#) for shallow regions. In Delft3D-FLOW the choice between upwind and the average approach for the water level in a U -point is controlled by flag `DPUOPT`. In the Z -model always upwind water levels and depth are used in velocity points ([Bijvelds, 2001](#)). The upwind flow through height in a U -velocity point based on an upwind water level is given by:

$$H_{m,n}^U = \bar{d}^\eta + \bar{\zeta}^\xi,$$

If $(\text{DPUOPT} = \text{UPW} \vee \text{hydraulic structure} \vee Z\text{-model})$ Then

$$H_{m,n}^U = \begin{cases} \bar{d}^\eta + \zeta_{m,n}, & U_{m,n} > 0, \\ \bar{d}^\eta + \zeta_{m+1,n}, & U_{m,n} < 0, \\ \bar{d}^\eta + \max(\zeta_{m,n}, \zeta_{m+1,n}), & U_{m,n} = 0, \end{cases}$$

Endif

(10.64)

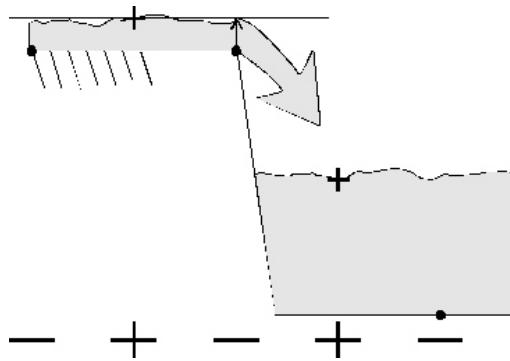


Figure 10.14: Drying of a tidal flat; upwind approach. The flow-through height is determined by flow direction, see [Equation 10.64](#), the velocity point remains wet.

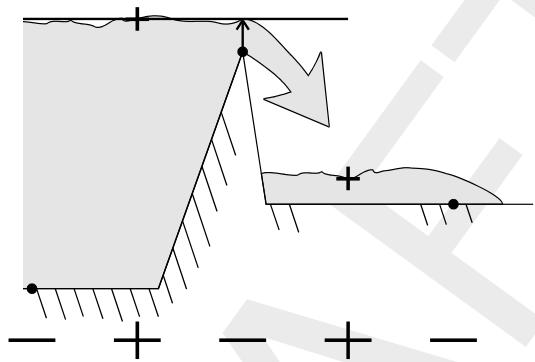


Figure 10.15: Overtopping of a river bank; upwind approach. The flow-through height is based on the maximum water level, see [Equation 10.64](#), the velocity point remains wet.

with $U_{m,n}$ representing the depth averaged velocity both for 2D and 3D. The computation of the upwind total water depth $H_{m,n}^V$ in a V -velocity point is similar. The upwind approach is *physically more realistic* for velocity points between cells with different bottom depth at low water and falling tide ([Figure 10.14](#)) or for weir like situations ([Figure 10.15](#)). Upwinding the water level in the determination of the total water depth (flow through height) enhances the discharge because the upwind water level is generally higher than the average water level, resulting in a larger flow area, which allows the water level gradient to drive a larger amount of water into the neighbouring cell during the next time step. Taking the maximum of the two surrounding water levels at a dry cell face prevents that a velocity point is artificially kept dry. The upwind approach is *physically less realistic* if the flow has the opposite direction as the water level gradient (wind driven flow).

[Figure 10.15](#) shows the situation of a river which overtops its bank. If we take the average water level to determine the total water depth at the crest, the velocity point remains dry. The water level will rise too much in the main channel of the river, leading to unrealistic water levels downstream. When the river run off increases, suddenly the flood plains are filled with water, generating a shock wave.

The *Flooding* scheme, see [subsection 10.4.4](#), is accurate for the approximation of the advection in rapidly varied flows due to sudden expansions or sudden contractions ([Stelling and Duinmeijer, 2003](#)). An example of such a flow problem is the simulation of the inundation of dry land with obstacles such as road banks and dikes. Due to a sudden contraction, the flow speed can become critical. The accuracy in the approximation of the critical discharge rate is dependent on the approximation (limiter) of the total water depth at a velocity point

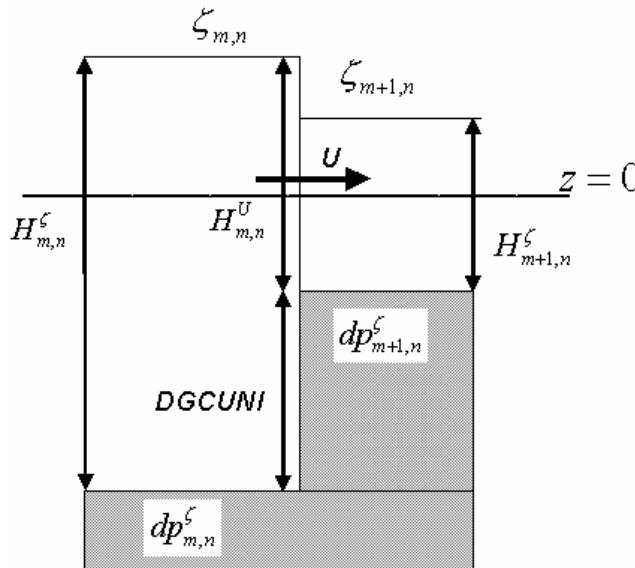


Figure 10.16: Special limiter for critical flow due to a sudden contraction (Flooding scheme and increase in bottom larger than DGCUNI)

downstream.

For the *Flooding* scheme the bottom is assumed to be represented as a staircase ($DPUOPT=MIN$) of tiles, centred around the water level points, see [Figure 10.7b](#) and [Figure 10.16](#). The flow through height in a U -velocity point is always based on an upwind water level, the user-defined marginal depth DCO is set at +999.

For the *Flooding* scheme the accuracy in the numerical approximation of the critical discharge rate for flow with steep bed slopes, can be increased by the use of a special approximation (slope limiter) of the total water depth at a velocity point downstream. The limiter function is controlled by the user-defined threshold depth for critical flow limiter $DGCUNI$, see [Figure 10.16](#) and:

$$\begin{aligned}
 &\text{If } (U_{m,n} > 0 \wedge d_{m,n}^\zeta > d_{m+1,n}^\zeta + DGCUNI) \text{ Then} \\
 &\quad H_{m,n}^U = \min \left[H_{m,n}^U, \max \left(\frac{2}{3} H_{m,n}^\zeta, H_{m+1,n}^\zeta \right) \right] \\
 &\text{Elseif } (U_{m,n} < 0 \wedge d_{m,n}^\zeta + DGCUNI < d_{m+1,n}^\zeta) \text{ Then} \\
 &\quad H_{m,n}^U = \min \left[H_{m,n}^U, \max \left(H_{m,n}^\zeta, \frac{2}{3} H_{m+1,n}^\zeta \right) \right] \\
 &\text{Endif}
 \end{aligned} \tag{10.65}$$

By the introduction of the user-defined threshold $DGCUNI$, see [Figure 10.16](#), the points for this special approach, are recognised automatically during the simulation, without having to specify all the points at input.

10.8.3 Drying and flooding criteria

As described in [section 10.4](#) an Alternating Direction Implicit (ADI) time integration method is used in Delft3D-FLOW. This method consists of two stages (half time steps). At both stages the same drying and flooding algorithm is applied. Therefore, we will only describe the drying and flooding algorithm for the first half time step.

The total water depth $H_{m,n}^U$ at a velocity point should at least be positive to guarantee a realistic discharge across a cell face. If the total water level drops below half of a user-specified threshold, then the velocity point is set dry. In 3D simulations the velocities are set to zero for all the computational layers. The computational cell is closed for the side normal to the velocity point. If the water level rises and the total water depth is larger than the threshold, the velocity point is set wet again. The drying threshold is given half the value of the wetting threshold (*hysteresis*) to inhibit changes of state in two consecutive time steps (“flip-flop”), due to oscillations introduced by the algorithm itself. When all four velocity points of a computational cell surrounding a water level point are dry then this computational cell will be set dry.

The total water depth $H_{m,n}^\zeta$ at a water level point should at least be positive to guarantee a positive control volume. If the total water level becomes negative, the four velocity points at the cell sides are set dry. In 3D simulations the velocities are set to zero for all the computational layers. If a negative control volume occurs, the half time step should be completely repeated and the computational time increases. Flooding is restricted to velocity points.

The thickness of the water layer of a dry cell (retention volume) is dependent on the threshold d specified by you. Therefore, the threshold value d must fulfil the following condition:

$$\delta \geq \frac{\partial \zeta}{\partial t} \frac{\Delta t}{2}. \quad (10.66)$$

In general, the magnitude of the disturbances generated by the drying and flooding algorithm will depend on the grid size, the bottom topography and the time step. The disturbances are small if the grid size is small and the bottom has smooth gradients. If the bottom has steep gradients across a large area on a tidal flat, a large area may be taken out of the flow domain in just one half integration time step. This will produce short oscillations. You can avoid this by smoothing the bottom gradients.

Flooding is an explicit process. The boundary of the wet area can only move one grid cell per time step. If the time step is too large an unphysical water level gradient at the wet-dry interface is built up, which will generate oscillations after flooding.

Attention should also be paid to the fact that depth values at points at closed boundaries are used in the total water depth of a velocity point parallel to the boundary and for the depth at a water level point for the MEAN-option. The depth at closed boundaries should be a bottom value near the coastline and not a land height.

In the first stage of the ADI-method, the drying and flooding algorithm in Delft3D-FLOW consists of the following four checks:

- 1 Drying check for velocity points in y -direction ($H_{m,n}^V < 0.5\delta$).
- 2 Drying check for velocity points in x -direction ($H_{m,n}^U < 0.5\delta$) and flooding check for velocity points in x -direction ($H_{m,n}^U > \delta$). These checks are based on the water level of the previous half time step.
- 3 Drying check for velocity points in x -direction ($H_{m,n}^U < 0.5\delta$) during iterative solution for new water level.
- 4 Drying check (negative volumes) for water level points ($H_{m,n}^\zeta < 0.0$).

In the second stage of the ADI-method, the directions are interchanged.

The threshold δ is specified by you at input. The total water depth at velocity points is compu-

ted according to either [Equation 10.63](#) or [Equation 10.64](#), corresponding to the average and the upwind approach, respectively. Default the average approximation is used in the σ -model and the upwind approach is always used in the Z -model.

Remark:

- ◊ Near hydraulic structures: (discharge points, weirs and barriers) always upwinding [Equation 10.64](#) is applied, independent of the marginal depth DCO.



The total water depth in water level points depends on the way in which the bottom depth is computed. Four options are available (see [subsection 10.8.1](#)).

In step four of the drying and flooding algorithm in Delft3D-FLOW, a check at each water level point is carried out to avoid negative volumes.

A smooth flooding behaviour is obtained if the simulation is initialised at high water, with the initial water level at the expected maximum level. In that case the water layer (retention volume) at dry points is initialised by the drying flooding algorithm. If a simulation is not started at high water, the water levels at dry points are initially set at the bottom depth. The thickness of the water layer is zero. If the computational cell is flooded, the water layer may be very thin and cause problems in combination with online salt transport or off-line water quality simulations. In Delft3D-FLOW the computational part is protected against “dividing by zero” by assuming that the total water depth is at least 1 centimetre.

You may define in velocity points so-called weirs or spillways. Weirs are hydraulic structures causing energy losses, see [section 10.9](#). For a *2D weir* the height of the crest, $HKRU$, is taken into account in the drying and flooding algorithm.

The drying check for a *2D weir* point at a U -point is given by:

$$H_{m,n}^U < \frac{1}{2}\delta \wedge \max(\zeta_{m-1,n}, \zeta_{m,n}) + HKRU_{m,n} < \frac{1}{2}\delta, \quad (10.67)$$

and flooding the flooding check:

$$H_{m,n}^U > \delta \wedge \max(\zeta_{m-1,n}, \zeta_{m,n}) + HKRU_{m,n} > \delta. \quad (10.68)$$

The weir acts as a thin dam for water levels lower than the crest height.

10.9 Hydraulic structures

In a Delft3D-FLOW model, so-called hydraulic structures can be defined to model the effect of obstructions in the flow which can not be resolved on the horizontal grid (sub-grid) or where the flow is locally non-hydrostatic. Examples of hydraulic structures in civil engineering are: gates, sills, sluices, barriers, porous plates, bridges, groynes, weirs. A hydraulic structure generates a loss of energy apart from the loss by bottom friction. At hydraulic structure points, an additional force term is added to the momentum equation, to parameterise the extra loss of energy. The term has the form of a friction term with a contraction or discharge coefficient.

In this section, the mathematical formulations and implementation of the hydraulic structures available in Delft3D-FLOW will be described in more detail. The hydraulic structures are divided into three basic types:

- ◊ hydraulic structures with quadratic friction,
- ◊ hydraulic structures with linear friction and
- ◊ floating structure.

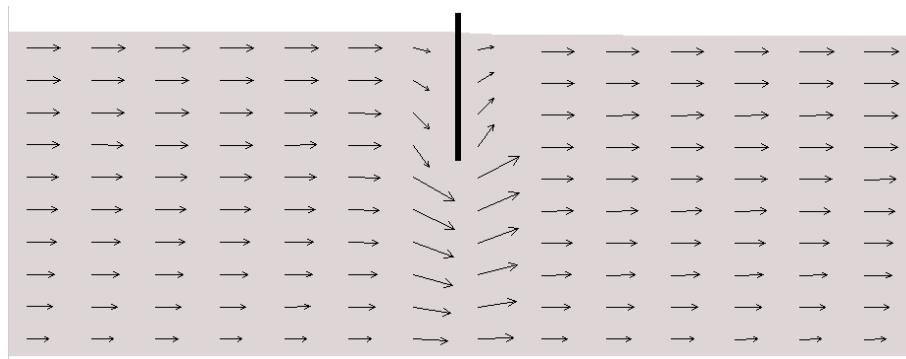


Figure 10.17: Example of a 3D Gate (vertical cross-section)

The flow condition at hydraulic structures may be supercritical. For supercritical flow, the downstream water level has no influence on the flow rate. The energy loss formulations presently available in Delft3D-FLOW assume subcritical flow. Only for the hydraulic structures of the types *2D weir* and *culvert* also the supercritical flow rate is computed accurately.

All hydraulic structures except for *culverts* are located on the interface between two computational cells. Around a hydraulic structure, there will be large horizontal gradients in the water level, the velocity field and in the concentrations. In order to prevent unphysical oscillations in the velocities and concentrations upstream of hydraulic structure points the user can switch (option) the discretisation of the advective terms at such points to an upwind approximation. For the momentum equations, the following energy preserving upwind discretization of advection is applied:

$$U \frac{\partial U}{\partial x} \Big|_{m,n,k} = \frac{1}{2} \frac{\partial U^2}{\partial x} \Big|_{m,n,k} = \begin{cases} \frac{U_{m,n,k}^2 - U_{m-1,n,k}^2}{2\Delta x}, & U_{m,n,k} > 0, \\ \frac{U_{m+1,n,k}^2 - U_{m,n,k}^2}{2\Delta x}, & U_{m,n,k} < 0. \end{cases} \quad (10.69)$$

For the transport equation, locally a first order upwind scheme is used by default.

Culverts have been implemented as two coupled discharge locations (one inlet and one outlet) which may be located in different parts of the computational grid. Culverts with a user defined discharge relation (type ‘u’) can be used to implement (non-local) discharge formulations representative of other kinds of structures, see [subsubsection B.3.4.6](#).

10.9.1 3D Gate

A 3D gate is in fact a thin dam with a limited height/depth (and positioned in the vertical). A gate may be used to model a vertical constriction of the horizontal flow such as barriers, sluices and Current deflection walls. The vertical constriction of the flow may vary in time by the lowering or raising of the gate.

A 3D gate is located at a velocity point and its width is assumed to be zero, so it has no influence on the water volume in the model area. The flow at all intermediate layers of the gate is set to zero. The layer near the top and the layer near the bottom of the gate may be partially blocked. Upstream of the structure the flow is accelerated due to contraction and downstream the flow is decelerated due to expansion.

For 3D gates not only the begin and end co-ordinates of its horizontal position, but also its vertical position can be specified. This means that some of the layers can be represented as vertical thin dams. In the σ -model the position of the 3D-gate can be:

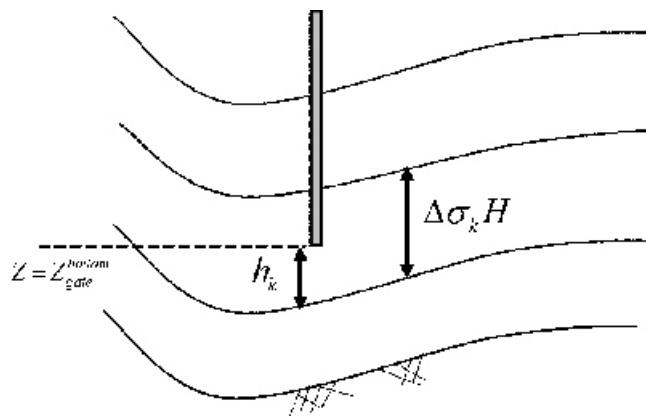


Figure 10.18: Computational layer partially blocked at bottom of gate

- ◊ “fixed” in the computational grid (and moving with the water level in the Cartesian co-ordinate system),
- ◊ “fixed” in the Cartesian co-ordinate system (and moving through the σ -grid),
- ◊ moving in the Cartesian co-ordinate system by lowering or raising of the gate (3D barrier).

For the latter two cases, a layer may be partially blocked near the top or the bottom of the gate, see [Figure 10.18](#). The wet cross section of that computational layer is reduced. The reduction factors are computed on basis of the vertical position of the gate and the time-dependent vertical σ -grid.

For more details on the data input requirements for the different type of 3D gates we refer to [subsection B.3.1](#).

10.9.2 Quadratic friction

The steady-state subcritical flow rate Q through a hydraulic structure is related to the difference between the upstream and downstream water levels:

$$Q = \mu A \sqrt{2g |\zeta_u - \zeta_d|}, \quad (10.70)$$

with μ the contraction coefficient ($0 < \mu \leq 1$), the wet flow-through area and the upstream and downstream water level, respectively. The contraction coefficient in the so-called Q - H relation is dependent on the kind of hydraulic structure. We assume that the hydraulic structure is “sub-grid”, and that there is a local equilibrium between the force on the flow due to the obstruction and the local water level gradient. The Q - H relation may be used to determine the coefficient c_{loss-U} in the quadratic friction term of the momentum equation, to model the effect of the hydraulic structure. For a Q - H relation at a U -velocity point:

$$g \frac{\zeta_u - \zeta_d}{\Delta x} = \frac{Q^2}{2\mu^2 A^2 \Delta x} = c_{loss-U} \frac{U_{m,n} |\vec{U}_{m,n}|}{\Delta x}, \quad (10.71)$$

The resistance coefficient c_{loss-U} need to be specified by the user at input.

For the following types of hydraulic structures, a quadratic formulation for the energy loss is implemented in Delft3D-FLOW:

- ◊ barrier
- ◊ bridge

- ◊ Current Deflection Wall
- ◊ 2D weir
- ◊ local weir
- ◊ porous plate
- ◊ culvert

10.9.2.1 Barrier

In Delft3D-FLOW, the hydraulic structure *barrier* is the combination of a movable gate and a quadratic friction term. It can be used in 2D and 3D models. In 3D models the model depth at the barrier point should be decreased to the sill depth for appropriate modelling of salt transport.

A barrier is located at a velocity point. For a 3D barrier, some of the layers are closed dependent on the gate height. The barrier is “sub-grid” and has no influence on the water volume in the model area. Upstream of a barrier the flow is accelerated due to contraction and downstream the flow is decelerated due to expansion. The expansion introduces a water level jump between the upstream and downstream water level, which is independent of the grid size. The energy loss for a barrier is taken into account by adding an extra quadratic friction term to the momentum equations [Equation 10.71](#). The appropriate energy loss coefficient should be specified for each barrier by the user, dependent on the local discharge relation, see [Equation 10.71](#). The quadratic friction is added to the momentum equations for all layers which are open.

The discharge relation presently available at barrier points in Delft3D-FLOW assumes *sub-critical free surface flow*. The depth-averaged flow-rate through a barrier is given by:

$$Q = \mu A \sqrt{2g |\zeta_u - \zeta_d|}, \quad (10.72)$$

with μ the barrier contraction coefficient ($0 < \mu \leq 1$). The contraction coefficient should be obtained from laboratory or field measurements. The contraction coefficient is used to determine the equivalent energy loss coefficient c_{loss-u} in [Equation 9.269](#) or c_{loss-v} in [Equation 9.270](#) depending on the U/V-direction of the barrier. Based on a depth-averaged analysis, the energy loss coefficient c_{loss} is related to the barrier contraction coefficient μ as:

$$c_{loss} = \frac{1}{2\mu^2}. \quad (10.73)$$

The user specifies either the energy loss coefficient c_{loss} directly (barrier loss type ‘a’) or the coefficients α and β (barrier loss type ‘b’) with which the energy loss coefficient is computed using:

$$c_{loss} = \frac{1}{2} \left(\frac{1}{\alpha + \beta p} - p \right)^2 \quad (10.74)$$

where p is the ratio of barrier opening height over total water depth at barrier ($p = 0$ if barrier is fully closed, $p = 1$ if barrier is completely open). The value of β is typically equal to $1 - \alpha$ such that energy loss coefficient c_{loss} becomes zero for a completely open gate. Energy losses due to barrier constrictions under fully open conditions may be schematised by means of a pair of porous plates immediately upstream and downstream of the barrier.

For a 3D barrier, part of the energy loss is computed directly by the discretisation of the convection terms in the momentum equations and the bottom friction term. The loss coefficient c_{loss} should be used for calibration. Between the gate and the sill we assume a uniform velocity profile. For a partial open layer k , see [Figure 10.18](#), with reduced thickness h_k the loss coefficient is increased proportionally to the square of the layer blockage factor.

Furthermore, free slip boundary conditions has been implemented at the transition from an open layer to a closed layer (representing the bottom side of the gate):

$$v_v \frac{\partial U}{\partial z} \Big|_{z=Z_{bottom}^{gate}} = 0 \quad (10.75)$$

with Z_{bottom}^{gate} the vertical position of the bottom of the gate.

10.9.2.2 Bridge

The flow resistance due to a jetty or a bridge is dependent on the blocking of the flow by the piers ([Farraday and Charlton, 1983](#)). For a row of piles perpendicular to the U -direction the energy loss coefficient c_{loss-u} perpendicular to the flow is given by (and has to be specified):

$$c_{loss-u} = \frac{NC_{drag}d_{pile}}{2\Delta y} \left(\frac{A_{tot}}{A_{eff}} \right)^2, \quad (10.76)$$

with:

A_{tot}	total cross sectional area ($A_{tot} = \zeta \Delta y$).
A_{eff}	effective wet cross sectional area (A_{tot} minus the area blocked by piles: $A_{eff} = A_{tot} - \zeta N d_{pile}$).
C_{drag}	the drag coefficient of a pier (pile) (1.0 for a smooth cylindrical pile).
d_{pile}	the diameter of a pile.
N	the number of piles in the grid cell.

At a bridge point the quadratic friction term is added to the momentum equation for all layers which are open. Currently, you should derive the value of the loss coefficient and specify at input (see the structure *Porous plate*).

10.9.2.3 Current Deflection Wall

In Delft3D-FLOW, the hydraulic structure *CDW* is the combination of a fixed gate and a quadratic friction term. It can be used in 2D and 3D models.

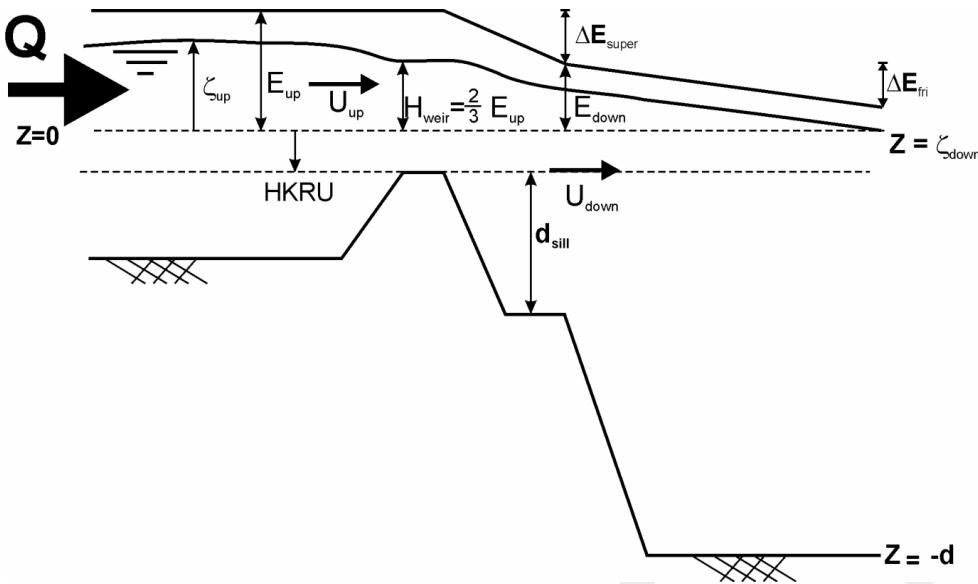
A *CDW* is located at a velocity point. In 3D some of the computational layers are closed dependent on the gate height and the upstream water level. The *CDW* is “sub-grid” and has no influence on the water volume in the model area. Upstream of a *CDW* the flow is accelerated due to contraction and downstream the flow is decelerated due to expansion. The expansion introduces a water level jump between the upstream and downstream water level, which is independent of the grid size. The energy loss for a *CDW* is taken into account by adding an extra quadratic friction term to the momentum equations [Equation 10.71](#). The appropriate energy loss coefficient should be specified by the user, dependent on the local discharge relation, see [Equation 10.71](#)) or dependent on the flow resistance that you want to induce; see [Equation 10.76](#). The quadratic friction is added to the momentum equations for all layers which are open.

10.9.2.4 Weir

In Delft3D-FLOW, the hydraulic structure *weir* is a fixed non-movable construction generating energy losses due to constriction of the flow. They are commonly used to model sudden changes in depth (roads, summer dikes) and groynes in simulation models of rivers.

2D Weir

The mathematical concept was developed by [Wijbenga \(1990\)](#). For a 2D *weir* the crest height



(HKRU) is taken into account only in the drying and flooding algorithm, to determine if a velocity point is wet or dry, see [subsection 10.8.3](#).

A 2D weir is assumed to be “sub-grid”, and the total water depth in the wet cross section to compute the discharge is still based on the bottom without crest. So the energy loss generated by the weir is not computed directly by the convective terms in the momentum equations. The energy loss is parameterised and added in the momentum equation as follows:

$$M_\xi = -\frac{gH\Delta E_{weir}}{\Delta x} \quad (10.77)$$

The flow condition at a local weir may be sub- or supercritical. For supercritical flow the discharge at the weir is completely determined by the energy head upstream and the discharge is limited by:

$$Q_{super} = \Delta y \frac{2}{3} E_{up} \sqrt{\frac{2}{3} g E_{up}} \quad (10.78)$$

with

$$E_{up} = \zeta_{up} + HKRU_{i+\frac{1}{2},j} + \frac{(U_{up})^2}{2g} = \zeta_{weir} + HKRU_{i+\frac{1}{2},j} + \frac{(U_{weir})^2}{2g} = \quad (10.79)$$

The flow condition depends on the water level downstream and the discharge rate.

The flow at the weir crest is supercritical if:

$$\zeta_{down} + HKRU \leq \frac{2}{3} E_{up} \quad \text{or} \quad Q \geq Q_{super} \quad (10.80)$$

For supercritical flow, the difference between the energy head upstream and downstream of the weir should be equal to the energy loss caused by the weir.

$$\Delta E_{weir} = E_{up} - E_{down} \quad (10.81)$$

with

$$E_{down} = \zeta_{down} + \frac{(U_{down})^2}{2g} \quad (10.82)$$

In the momentum equation, the energy loss term is used to control the discharge at the theoretical supercritical discharge rate [Equation 10.78](#).

For sub-critical flow conditions, the energy loss ΔE_{weir} is based on experimental data “Tabelenboek van Rijkswaterstaat” ([Vermaas, 1987](#)) ΔE_{table} and/or the formula of Carnot ΔE_{Car} . This depends on the flow velocity U_{weir} at the weir crest, This velocity is derived from the discharge upstream and the crest height, assuming conservation of energy for the flow contraction.

$$U_{weir} = \frac{Q_{weir}}{\Delta y (\zeta_{up} + HKRU)} \quad (10.83)$$

The energy loss following Carnot is given by:

$$\Delta E_{Car} = \frac{(U_{weir} - U_{down})^2}{2g} \quad (10.84)$$

with

$$U_{down} = \frac{Q_{weir}}{\Delta y (\zeta_{down} + HKRU + d_{sill})} \quad (10.85)$$

If the flow velocity at the weir crest is less than 0.25 m/s the energy loss is calculated according to ΔE_{Car} , when the velocity is between 0.25 m/s and 0.5 m/s, a weighted average is used between ΔE_{Car} and ΔE_{table} , if the velocity is more than 0.5 m/s, ΔE_{table} is used.

The energy loss is assumed to be normal to the obstruction and for a U -weir it is added to the momentum equation as follows

$$\frac{\partial U}{\partial t} + \dots = \dots - \frac{g \Delta E^n U_{i+1/2,j}^{n+1/2}}{\Delta x U_{i+1/2,j}^n} \quad (10.86)$$

$$\Delta E^n = (1 - \theta) \Delta E^n + \theta \Delta E^{n-1/2} \quad (10.87)$$

where the velocity in the denominator of the first term is replaced by the critical flow velocity over the weir in case of a perfect weir. To prevent oscillations when the flow over the weir is just below supercritical a relaxation parameter θ has been introduced which can be set to a value between 0 and 1 by means of the `ThetaW` keyword. The default value is 0 meaning no under-relaxation used for the energy loss ΔE . See [subsection B.3.2.4](#) for details on the user-input.

Local weir

In 3D models, the model depth at a weir point should be decreased for appropriate modelling of salt transport. The crest height of a *local weir* is not derived from the bottom but specified at input. The crest height (*HKRU*) is taken into account only in the drying and flooding algorithm, to determine if a velocity point is wet or dry, see [subsection 10.8.3](#). The total water depth

in the wet cross-section at a local weir point is based on the crest height. For a *local weir*, part of the energy loss is computed directly by the discretisation of the convection terms in the momentum equations and the bottom friction term. The loss coefficient c_{loss-U} should be used for calibration.

10.9.2.5 Porous plate

For a so-called *porous plate* the energy loss coefficient should be specified by the user at input.

$$M_\xi = -c_{loss-U} \frac{U_{m,n} |\vec{U}_{m,n}|}{\Delta x} \quad (10.88)$$

10.9.2.6 Culvert

In Delft3D-FLOW intake/outlet couplings have been implemented. A culvert is a special intake/outlet coupling in which the discharge rate depends on the flow regime. In case of a normal intake/outlet coupling, which is called a “power station”, the discharge rate is specified on input.

Definition of a culvert

In Delft3D-FLOW four types of culverts are distinguished:

Type	Description
‘c’	One-way culvert with a “simple” discharge formulation
‘d’	Two-way culvert with a “more generalized” discharge formulation
‘e’	One-way culvert with a “more generalized” discharge formulation
‘u’	Two-way culvert/structure with user-defined discharge formulation

Culvert of type ‘c’

For a *culvert of type ‘c’*, which corresponds to a completely submerged culvert, the discharge rate through the culvert (in [m³/s]) is computed by the Delft3D-FLOW program, according to

$$Q = \mu A \sqrt{2g \max(0, \zeta_{intake} - \zeta_{outlet})} \quad (10.89)$$

with μ the culvert loss coefficient (dimensionless), A the area (in m²) of the culvert opening and ζ_{intake} and ζ_{outlet} the water levels at the intake and outlet, respectively.

Culvert of type ‘d’ or ‘e’

It is known that [Equation 10.89](#) does not perform satisfactorily in all flow conditions. In particular, this is the case when the downstream water level is below the so-called critical depth. Therefore, a more general formulation for the computation of the discharge through the culvert is available. Six flow classifications are distinguished, with different discharge relations for each type. The classification was taken from [French et al. \(1987\)](#), (page 368). However, more generalized formulations were proposed by WL Borgerhout for the culvert loss coefficient and have been incorporated.

Let us first introduce some notation:

$$\zeta_u = \max(0, \zeta_{intake} - z_{culvert}) \quad \text{and} \quad \zeta_d = \max(0, \zeta_{outlet} - z_{culvert}) \quad (10.90)$$

in which $z_{culvert}$ is the vertical position of the bottom of the culvert relative to the reference level (positive upwards). Furthermore, the critical depth is defined by

$$H_c = \sqrt[3]{\frac{Q^2}{gW^2}} \quad (10.91)$$

with W the width (in [m]) of the culvert.

To consider six flow classifications and their discharge characteristics some variables are defined. L is the length of the culvert, n the Manning's coefficient (in [$m^{1/3}/s$]) representing the energy loss due to the culvert and a discharge coefficient that is prescribed by the user (c_D^1, c_D^2 and c_D^3), in which the superscript represents the type(s) of flow condition. Furthermore, S_0 is the slope of the culvert, S_c is the so-called critical slope, u is the velocity through the culvert and α is the so-called energy loss correction coefficient, which has to be specified by the user as well. This coefficient represents all energy losses due to a culvert that are not part of the other terms in the discharge relations.

The six flow classifications and their discharge characteristics read:

Type 1 (supercritical flow with critical depth at intake; steep culvert slope)

$$Q = \mu H_c W \sqrt{2g(\zeta_u - H_c)} \quad \text{with} \quad \mu = c_D^1 \quad (10.92)$$

Type 2 (supercritical flow with critical depth at outfall; mild culvert slope)

$$\begin{aligned} Q &= \mu(H^*) H_c W \sqrt{2g(\zeta_u - H_c)} \quad \text{with} \quad H^* = 0.5H_c + 0.5\zeta_u \\ \mu(H^*) &= c_D^1 / \sqrt{1 + \left\{ \frac{2gL n^2}{(R^*)^{4/3}} + \alpha \right\} (c_D^1)^2 \left(\frac{H_c}{H^*} \right)^2} \quad \text{and} \quad R^* = \frac{H^* W}{2H^* + W} \end{aligned} \quad (10.93)$$

Type 3 (tranquill flow):

$$\begin{aligned} Q &= \mu(H^*) \zeta_d W \sqrt{2g(\zeta_{intake} - \zeta_{outlet})} \quad \text{with} \quad H^* = 0.5\zeta_u + 0.5\zeta_d \\ \mu(H^*) &= c_D^1 / \sqrt{1 + \left\{ \frac{2gL n^2}{(R^*)^{4/3}} + \alpha \right\} (c_D^1)^2 \left(\frac{\zeta_d}{H^*} \right)^2} \quad \text{and} \quad R^* = \frac{H^* W}{2H^* + W} \end{aligned} \quad (10.94)$$

Type 4 (submerged flow)

$$\begin{aligned} Q &= \mu(H^*) H W \sqrt{2g(\zeta_{intake} - \zeta_{outlet})} \quad \text{with} \quad H^* = H \\ \mu(H^*) &= c_D^2 / \sqrt{1 + \left\{ \frac{2gL n^2}{(R^*)^{4/3}} + \alpha \right\} (c_D^2)^2} \quad \text{and} \quad R^* = \frac{H^* W}{2H^* + 2W} \end{aligned} \quad (10.95)$$

Type 5 (rapid flow at inlet)

$$Q = \mu(H^*) H W \sqrt{2g\zeta_u} \quad \text{with} \quad \mu(H^*) = c_D^3 \quad \text{and} \quad H^* = H \quad (10.96)$$

Type 6 (full flow free outlet)

$$Q = \mu(H^*)HW\sqrt{2g(\zeta_u - H)} \quad \text{with} \quad H^* = H$$

$$\mu(H^*) = c_D^2 / \sqrt{1 + \left\{ \frac{2gL_n^2}{(R^*)^{4/3}} + \alpha \right\} (c_D^2)^2} \quad \text{and} \quad R^* = \frac{HW}{2H + 2W} \quad (10.97)$$

From Eqs. 10.92 to 10.97 it can be seen that three different discharge coefficients are used, namely

- c_D^1 : flow classification type 1, 2 and 3
- c_D^2 : flow classification types 4 and 6
- c_D^3 : flow classification type 5

In the table below the conditions are listed for these six flow regimes.

Type	Flow regime	Conditions			
		ζ_u	ζ_d	ζ_d	other
1	supercritical flow with critical depth at intake	$< 1.5H$	$\leq H$	$\leq H_c$	$S_o > S_c$
2	supercritical flow with critical depth at outlet	$< 1.5H$	$\leq H$	$\leq H_c$	$S_o \leq S_c$
3	tranquil flow	$< 1.5H$	$\leq H$	$> H_c$	
4	submerged flow	$> H$	$> H$		
5	rapid flow at inlet	$\geq 1.5H$	$\leq H$	$\leq H_c$	
6	full flow free outlet	$\geq 1.5H$	$\leq H$	$\geq H_c$	

**Remarks:**

- ◊ From this table it can be verified that all possible flow conditions can occur. Either, (types 5 or 6), or (types 1, 2 or 3). If none of these flow conditions are satisfied, then (type 4).
- ◊ Flow type 1 corresponds to culverts with a steep slope, whereas flow type 2 represents a mild culvert slope. Since the culvert implementation applied in Delft3D-FLOW is based on strict horizontal culverts, *only flow type 2 can occur and not flow type 1*.
- ◊ The formulations are more or less identical to the ones applied in SOBEK-RURAL of Deltares. The differences are in the computation of the culvert loss coefficient μ .

Difference between culvert type 'd' and 'e'

In the previous text it is described how the discharge through a culvert is computed. For a one-way culvert (type 'e') the discharge is always from the intake location to the outlet location. If the water level at the outlet location is higher than at the intake, then there is no flow through the culvert. In case of a two-way culvert (type 'd') the location with the highest water level (thus, either intake or outlet location) is considered as the intake location. This means that for a two-way culvert flow through the culvert in two-directions is possible.

We note that culvert types 'c' and 'e' are comparable. Only the computation of the discharge through the culvert differs, see [Equation 10.89](#) versus Eqs. 10.92 to 10.97.

Culvert of type ‘u’

The discharge Q through a culvert of type ‘u’ is determined by a subroutine that you provide yourself. The discharge can be based on the water levels at both ends of the culvert. You can use this routine to reproduce the formulations for culverts of types ‘c’, ‘d’ and ‘e’ and variations thereof, or you can implement a discharge functions that is valid for some completely different structure. This culvert type can be used to implement a variety of structures resulting in point-to-point fluxes (either uniform over the water depth or at specific heights). An example routine is given in [subsection B.3.8](#).

10.9.3 Linear friction

Rigid sheet

For a so-called *rigid sheet*, the resistance force is assumed to be linearly dependent on the flow. The resistance coefficient, c_{loss-U} , can be specified by the user at input.

$$M_\xi = -c_{loss-U} \frac{U_{m,n}}{\Delta x} \quad (10.98)$$

10.9.4 Floating structure

Floating structures can be modelled. Via an external file the depth of the floating structure has to be specified. A floating structure is simulated by assuming a local pressure that pushes the water to a depth specified by the key word `Filfls`. See [subsection B.3.6](#) for details of the data input requirements.

In combination with a floating structure, an artificial compression coefficient may be specified. Then, the time derivative in the continuity equation [Equation 9.9](#) is multiplied by the artificial compression coefficient α , (keyword `Riglid` in the input) yielding:

$$\begin{aligned} \alpha \frac{\partial \zeta}{\partial t} + \frac{1}{\sqrt{G_{\xi\xi}} \sqrt{G_{\eta\eta}}} \frac{\partial [(d + \zeta) u \sqrt{G_{\eta\eta}}]}{\partial \xi} + \frac{1}{\sqrt{G_{\xi\xi}} \sqrt{G_{\eta\eta}}} \frac{\partial [(d + \zeta) v \sqrt{G_{\xi\xi}}]}{\partial \eta} + \\ + \frac{\partial \omega}{\partial \sigma} = H (q_{in} - q_{out}) . \end{aligned} \quad (10.99)$$

10.10 Artificial vertical mixing due to σ co-ordinates

The σ -transformation is boundary-fitted in the vertical. The bottom boundary and free surface are represented smoothly. The water column is divided into the same number of layers independent of the water depth. In a σ -model, the vertical resolution increases automatically in shallow areas.

For steep bottom slopes combined with vertical stratification, σ -transformed grids introduce numerical problems for the accurate approximation of horizontal gradients both in the baroclinic pressure term and in the horizontal diffusion term. Due to truncation errors artificial vertical mixing and artificial flow may occur, [Leendertse \(1990\)](#) and [Stelling and Van Kester \(1994\)](#). This artificial vertical transport is sometimes called “creep”.

Let ζ be the position of the free surface, d the depth measured downward positive and H the total water depth. If we consider the transformation from Cartesian co-ordinates to σ co-ordinates, defined by:

$$x = x^*, y = y^*, \sigma = \frac{z - \zeta}{H}, \quad (10.100)$$

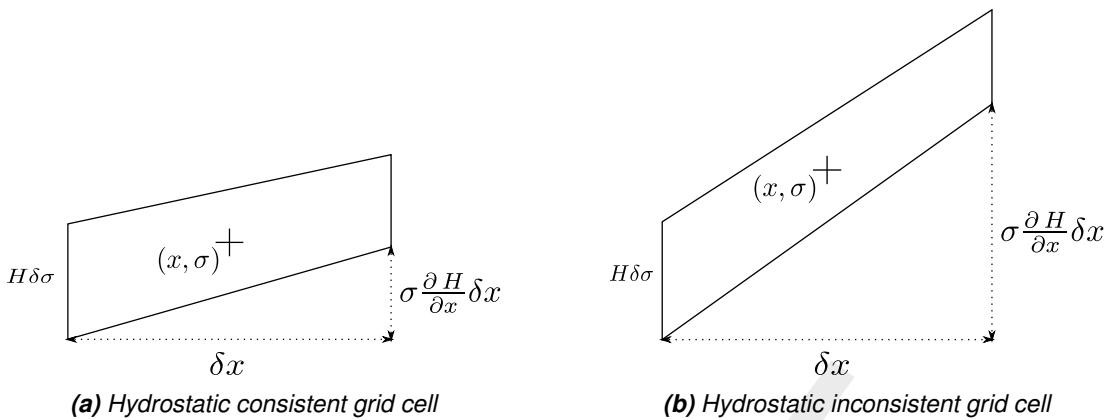


Figure 10.19: Example of a hydrostatic consistent and inconsistent grid;
(a) $H\delta\sigma > \sigma \frac{\partial H}{\partial x} \delta x$, **(b)** $H\delta\sigma < \sigma \frac{\partial H}{\partial x} \delta x$

the horizontal pressure gradient reads:

$$\frac{\partial p}{\partial x} = \frac{\partial p^*}{\partial x^*} \frac{\partial x^*}{\partial x} + \frac{\partial p^*}{\partial \sigma} \frac{\partial \sigma}{\partial x} = \frac{\partial p^*}{\partial x^*} - \frac{1}{H} \left[\frac{\partial \zeta}{\partial x} + \sigma \frac{\partial H}{\partial x} \right] \frac{\partial p^*}{\partial \sigma}. \quad (10.101)$$

In case of vertical stratification near steep bottom slopes, small pressure gradients at the left-hand side may be the sum of relatively large terms with opposite sign at the right-hand side. Small truncation errors in the approximation of both terms result in a relatively large error in the pressure gradient. This artificial forcing produces artificial flow. The truncation errors depend on the grid sizes Δx and Δz . Observations of this kind has led to the notion of “hydrostatic consistency”, see also [Figure 10.19](#). In the notation used by [Haney \(1991\)](#) this consistency relation is given by:

$$\left| \frac{\sigma}{H} \frac{\partial H}{\partial x} \right| < \left| \frac{\partial \sigma}{\partial x} \right|. \quad (10.102)$$

From this equation, it can be seen that by increasing the number of σ -levels the consistency condition will eventually be violated.

Similarly, for the horizontal diffusion term, the transformation from Cartesian co-ordinates to σ co-ordinates leads to various cross derivatives. For example, the transformation of a simple second order derivative leads to:

$$\frac{\partial^2 c}{\partial x^2} = \frac{\partial^2 c^*}{\partial x^{*2}} + \left(\frac{\partial \sigma}{\partial x} \right)^2 - \frac{\partial^2 c^*}{\partial \sigma^2} + 2 \frac{\partial \sigma}{\partial x} - \frac{\partial^2 c^*}{\partial x^* \partial \sigma} + \frac{\partial^2 \sigma}{\partial x^2} - \frac{\partial c^*}{\partial \sigma}. \quad (10.103)$$

For such a combination of terms it is difficult to find a numerical approximation that is stable and positive, see [Huang and Spaulding \(1996\)](#). Near steep bottom slopes or near tidal flats where the total depth becomes very small, truncations errors in the approximation of the horizontal diffusive fluxes in σ -co-ordinates are likely to become very large, similar to the horizontal pressure gradient.

In Delft3D-FLOW the stress tensor is redefined in the σ co-ordinate system assuming that the horizontal length scale is much larger than the water depth ([Blumberg and Mellor, 1985](#)) and that the flow is of boundary-layer type. The horizontal gradients are taken along σ -planes.

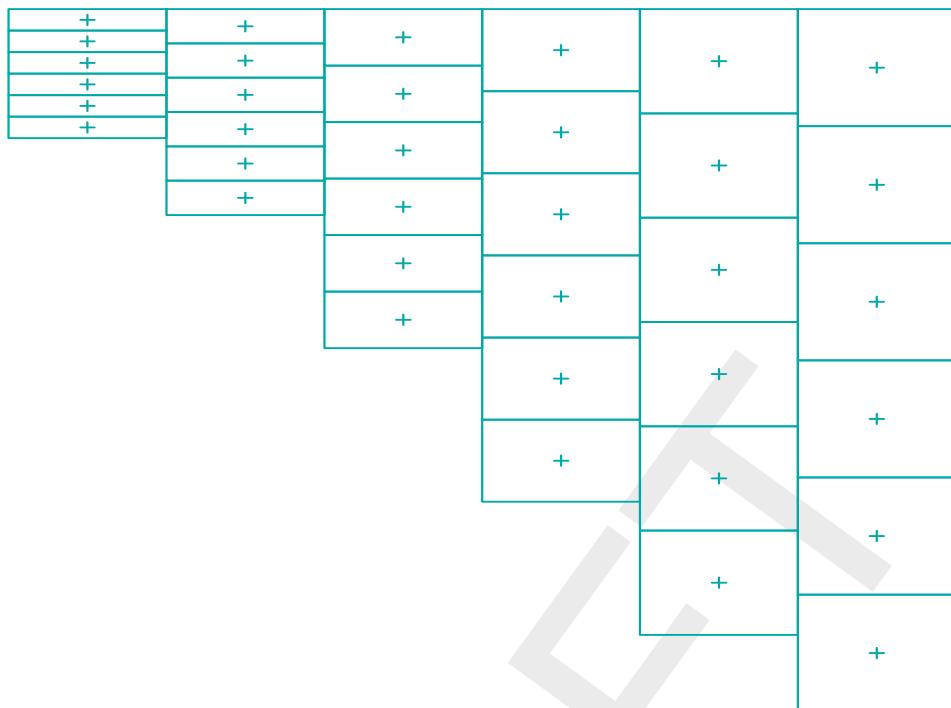


Figure 10.20: Finite Volume for diffusive fluxes and pressure gradients

This approach guarantees a positive definite operator, also on the numerical grid ([Beckers et al., 1998](#)). For a detailed description we refer to [chapter 9](#).

If the same approach is used for the horizontal diffusion operator in the transport equation:

$$\frac{\partial^2 c}{\partial x^2} \approx \frac{\partial^2 c^*}{\partial x^{*2}}, \quad (10.104)$$

Horizontal diffusion will lead to vertical transport of matter through vertical stratification interfaces (pycnocline) which is unphysical. A more accurate, strict horizontal discretization is needed.

In Delft3D-FLOW an option is available that minimises artificial vertical diffusion and artificial flow due to truncation errors; see [subsection 4.5.8](#) option *Correction for sigma-co-ordinates*. A method has been implemented which gives a consistent, stable and monotonic approximation of both the horizontal pressure gradient and the horizontal diffusion term, even when the hydrostatic consistency condition Eq. is not fulfilled. This “anti-creep” option is based upon a Finite Volume approach; see [Figure 10.20](#). The horizontal diffusive fluxes and baroclinic pressure gradients are approximated in Cartesian co-ordinates by defining rectangular finite volumes around the σ -co-ordinate grid points. Since these boxes are not nicely connected to each other, see [Figure 10.21](#), an interpolation in z co-ordinates is required to compute the fluxes at the interfaces.

Since the centres of the finite volumes on the left-hand side and right-hand side of a vertical interval are not at the same vertical level, a z -interpolation of the scalar concentration c is needed to compute strictly horizontal derivatives. The values obtained from this interpolation are indicated by c_1^* and c_2^* respectively in [Figure 10.21](#). [Stelling and Van Kester \(1994\)](#) apply a non-linear filter to combine the two consistent approximations of the horizontal gradient,

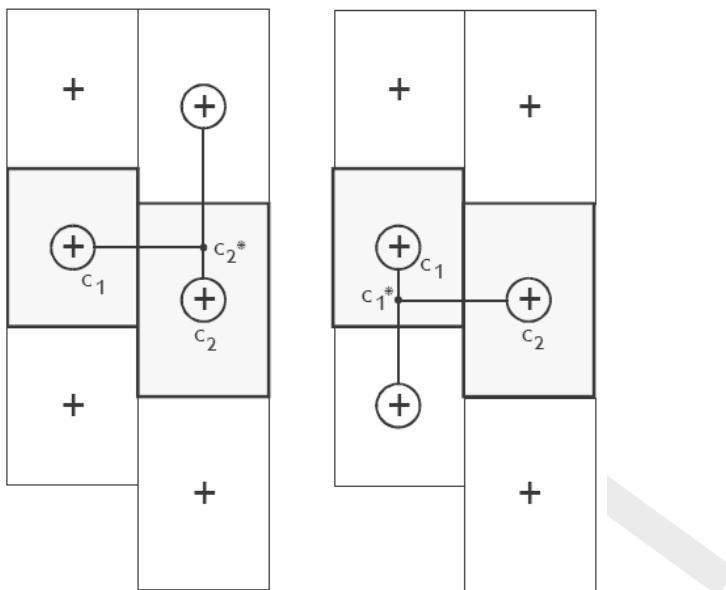


Figure 10.21: Left and right approximation of a strict horizontal gradient

$s_1 = (c_2^* - c_1)/\Delta x$ and $s_2 = (c_2 - c_1^*)/\Delta x$:

```

If  $s_1 \times s_2 < 0$  Then
 $\frac{\Delta c}{\Delta x} = 0$ 
Else
 $\frac{\Delta c}{\Delta x} = sign(s_1) \times \min(|s_1|, |s_2|)$ 
Endif

```

(10.105)

If an interval has only grid boxes at one side, the derivative is directly set to zero. The horizontal fluxes are summed for each control volume to compute the diffusive transport. The integration of the horizontal diffusion term is explicit with time step limitation:

$$\Delta t \leq \frac{1}{D_H} \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right)^{-1}. \quad (10.106)$$

The derivatives are used in the integral for the baroclinic pressure force in the momentum equation:

$$P_x(x, z) = \frac{1}{\rho_0} \int_z^\zeta g \frac{\partial \rho(x, s)}{\partial x} ds. \quad (10.107)$$

Originally, this approach was implemented in Delft3D-FLOW. [Slørdal \(1997\)](#) stated that the above approximation may sometimes produce errors of the same sign which leads to a systematic underestimation of the baroclinic pressure term. This underestimation can be ascribed to the non-linear filter, which selects the minimum of the two gradients under consideration. This limiter is fully analogous to the min-mod limiter used for the construction of monotone advection schemes ([Hirsch, 1990](#)). Since the same approximation of the horizontal gradient is used for the horizontal diffusion flux, it is important to ensure that the difference operator is positive definite in order to get physically realistic solutions. The maximum and minimum of a variable being transported by diffusion do not increase or decrease (min-max principle). By taking the minimum of the gradients, [Stelling and Van Kester \(1994\)](#) show that, the min-max

principle is fulfilled. [Beckers et al. \(1998\)](#) show that any nine-point consistent linear discretization of the horizontal diffusion on the σ -grid does not fulfil the min-max principle. From numerical tests [Slørdal \(1997\)](#) concluded that the underestimation is reduced by increasing the vertical resolution, but is sometimes enhanced by increasing the horizontal resolution.

Let s_4 be a consistent approximation of the horizontal gradient $s_4 = (s_1 + s_2)/2$. [Slørdal \(1997\)](#) suggested to take s_4 as approximation of the horizontal gradient. He calls his approach the “modified Stelling and Van Kester scheme”. It is equivalent to linear interpolation at a certain z -level before taking the gradient. It is more accurate than taking the minimum of the absolute value of the two slopes s_1 and s_2 but it does not fulfil the min-max principle for the diffusion operator. It may introduce wiggles and a small persistent artificial vertical diffusion (except for linear vertical density distributions). Due to the related artificial mixing, stratification may disappear entirely for long term simulations, unless the flow is dominated by the open boundary conditions.

By introducing an additional approximation of the horizontal gradient in the filter algorithm defined by $s_3 = (c_2 - c_1)/\Delta x$, the stringent conditions of the minimum operator can be relaxed somewhat. The drawback of underestimation of the baroclinic pressure force reported by [Slørdal \(1997\)](#) can be minimised without loosing that the method fulfils the min-max principle. This third gradient s_3 , which is consistent for $\min(|s_1|, |s_2|) < s_3 < \max(|s_1|, |s_2|)$, has point-to-point transfer properties and therefore leads to a positive scheme for sufficiently small time steps. The following non-linear approach presently available in Delft3D-FLOW is both consistent and assures the min-max principle:

```
If  $s_1 \times s_2 < 0$  Then
   $\frac{\Delta c}{\Delta x} = 0$ 
Elseif  $|s_4| < |s_3|$  Then
   $\frac{\Delta c}{\Delta x} = s_4$ 
Elseif  $\min(|s_1|, |s_2|) < |s_3| < \max(|s_1|, |s_2|)$  Then
   $\frac{\Delta c}{\Delta x} = s_3$ 
Else
   $\frac{\Delta c}{\Delta x} = \text{sign}(s_1) \min(|s_1|, |s_2|)$ 
Endif
```

(10.108)

The method requires a binary search to find the indices of neighbouring grid boxes, which is time consuming. The increase in computation time is about 30%.

If the streamlines are strictly horizontal, transport of matter discretised on a σ co-ordinate grid may still generate some numerical vertical diffusion by the discretisation of the advection terms.

10.11 Smoothing parameter boundary conditions

The solution of the shallow water equations is uniquely determined by a set of initial and boundary conditions. The boundary conditions represent the external forcing and determine the steady state solution. The deviation between the initial condition and the steady state solution generates a transient (mass spring system analogy).

In Delft3D-FLOW the initial conditions for the water level and velocities are obtained from:

- ◊ The results of a previous run (warm start).
- ◊ User-prescribed (space varying or uniform) input fields (cold start).

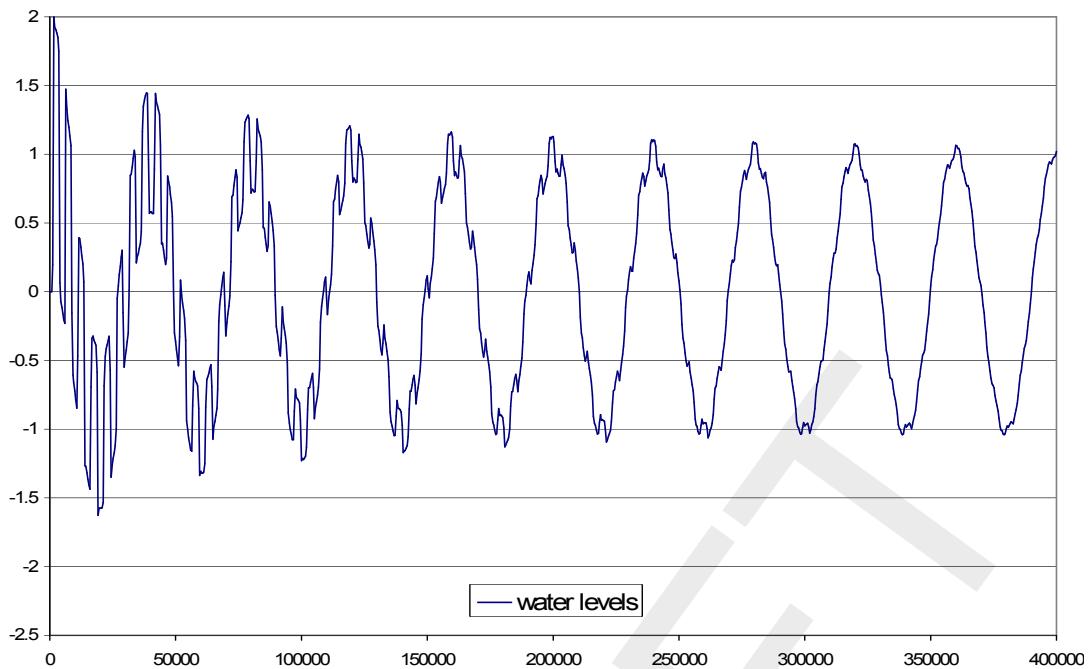


Figure 10.22: Cold start with damping of eigen oscillations due to bottom friction

The initial values are usually inconsistent with the boundary conditions at the start time of the simulation. This will generate a transient solution consisting of waves with eigen frequencies of the model domain. These waves may be reflected at the boundaries and generate a standing wave system. The waves should be dissipated completely by bottom friction and viscosity terms or leave the domain through the open boundaries, see [Figure 10.22](#). The damping of the transient solution determines the spin-up time of the numerical model.

To reduce the amplitude of the transient wave and the spin-up time of a model, Delft3D-FLOW has an option to switch on the boundary forcing gradually by use of a smoothing period (parameter T_{smo}). The boundary forcing is given by:

$$F_b^{smo}(t) = (1 - \alpha)F_b(t) + \alpha F_i(t), \quad (10.109)$$

with:

$$\alpha = \begin{cases} \frac{T_{smo}-t}{T_{smo}}, & t < T_{smo}, \\ 0, & t \geq T_{smo}, \end{cases} \quad (10.110)$$

and:

- $F_i(t)$ The initial value at the boundary.
- $F_b(t)$ The boundary condition.
- $F_b^{smo}(t)$ The boundary condition after smoothing.

Smoothing is possible both for a warm and a cold start. If the initial conditions are consistent with the boundary conditions at the start time of the simulation then the smoothing time should be set to zero.

10.12 Assumptions and restrictions

The solution of the discretized equations is just an approximation of the exact solution. The accuracy of the solution depends not only on the numerical scheme, but also on the way in

which the bottom topography, the geographical area, and the physical processes (turbulence, wave-current interaction) are modelled.

The time integration method strongly influences the wave propagation when applying a large time step. The assumption is made that, by restricting the computational time step, the free surface waves can be propagated correctly.

The open boundaries in a numerical flow model are artificial in the sense that they are introduced to limit the computational area that is modelled. The free surface waves should pass these boundaries completely unhindered. In the numerical model, wave reflections may occur at the open boundaries. These reflections will be observed as spurious oscillations superimposed on the physical results. In Delft3D-FLOW, weakly reflective boundaries are available which diminish these effects.

The open boundary can be divided into segments (sections). The boundary conditions in Delft3D-FLOW are specified for these segments, two values per segment are required, one for each segment end. The boundary condition at internal points within this segment is obtained by linearly interpolation between the end points. Therefore, if the phase variation of the tidal forcing along an open boundary segment is non-linear then the number of open boundary segments should be increased so that the phases at all the segments can be specified. Phase errors may generate an artificial re-circulation flow (eddy) near the open boundary. For steady-state simulations, a similar effect may be observed near the open boundaries if the effect of the Coriolis force on the water level gradient along the open boundary is not taken into account in the boundary conditions.

Care must be taken when time-series of measurements are directly prescribed as forcing functions at the open boundaries. Measurements often contain a lot of undesired noise, due to meteorological or other effects. For tidal flow computations, calibration on processed field data obtained from a tidal analysis or Fourier analysis, avoids this problem.

DRAFT

11 Sediment transport and morphology

11.1 General formulations

11.1.1 Introduction

The sediment transport and morphology module supports both bedload and suspended load transport of non-cohesive sediments and suspended load of cohesive sediments. For schematisation we distinguish “mud” (cohesive suspended load transport), “sand” (non-cohesive bedload and suspended load transport) and “bedload” (non-cohesive bedload only or total load transport) fractions. A model may contain a mixture of up to 99 suspended (i.e. “sand” and “mud”) fraction and an arbitrary amount of “bedload” fractions if computer memory and simulation time allows. The only difference between “bedload” and “sand” fractions lies in the fact that the suspended load advection-diffusion equation is not solved for the “bedload” fraction. If the suspended load is known to be negligible (either due to sediment diameter or sediment transport formula chosen), the “bedload” approach is more efficient. Sediment interactions are taken into account although especially in the field of sand-mud interactions still a couple of processes are lacking.

11.1.2 Suspended transport

Three-dimensional transport of suspended sediment is calculated by solving the three-dimensional advection-diffusion (mass-balance) equation for the suspended sediment:

$$\frac{\partial c^{(\ell)}}{\partial t} + \frac{\partial u c^{(\ell)}}{\partial x} + \frac{\partial v c^{(\ell)}}{\partial y} + \frac{\partial (w - w_s^{(\ell)}) c^{(\ell)}}{\partial z} + \\ - \frac{\partial}{\partial x} \left(\varepsilon_{s,x}^{(\ell)} \frac{\partial c^{(\ell)}}{\partial x} \right) - \frac{\partial}{\partial y} \left(\varepsilon_{s,y}^{(\ell)} \frac{\partial c^{(\ell)}}{\partial y} \right) - \frac{\partial}{\partial z} \left(\varepsilon_{s,z}^{(\ell)} \frac{\partial c^{(\ell)}}{\partial z} \right) = 0, \quad (11.1)$$

where:

$c^{(\ell)}$	mass concentration of sediment fraction (ℓ) [kg/m ³]
u, v and w	flow velocity components [m/s]
$\varepsilon_{s,x}^{(\ell)}, \varepsilon_{s,y}^{(\ell)}$ and $\varepsilon_{s,z}^{(\ell)}$	eddy diffusivities of sediment fraction (ℓ) [m ² /s]
$w_s^{(\ell)}$	(hindered) sediment settling velocity of sediment fraction (ℓ) [m/s]

The local flow velocities and eddy diffusivities are based on the results of the hydrodynamic computations. Computationally, the three-dimensional transport of sediment is computed in exactly the same way as the transport of any other conservative constituent, such as salinity, heat, and constituents. There are, however, a number of important differences between sediment and other constituents, for example, the exchange of sediment between the bed and the flow, and the settling velocity of sediment under the action of gravity. These additional processes for sediment are obviously of critical importance. Other processes such as the effect that sediment has on the local mixture density, and hence on turbulence damping, can also be taken into account. In addition, if a net flux of sediment from the bed to the flow, or vice versa, occurs then the resulting change in the bathymetry should influence subsequent hydrodynamic calculations. The formulation of several of these processes (such as, settling velocity, sediment deposition and pickup) are sediment-type specific, this especially applies for sand and mud. Furthermore, the interaction of sediment fractions is important for many processes, for instance the simultaneous presence of multiple suspended sediment fractions has implications for the calculation of the local hindered settling velocity of any one sediment fraction as well as for the resulting mixture density.

The following sections describe, at a conceptual level, the differences between the suspended transport of sediments and the transport of other conservative constituents. At the same time we discuss some of the differences in general terms and refer for the details of the mathematical formulations to Sections 11.2 and 11.3.



Remarks:

- ◊ The presence of multiple sediment fractions considerably complicates the calculation of the density of the bed and the availability of a particular class of sediment at the bed. See the sections on sediment interaction (Sections 11.2.4 and 11.4.3) and bed composition models ([subsection 11.6.4](#)).
- ◊ Small negative sediment concentrations ($-1 \cdot 10^{-3} \text{ kg/m}^3$) can be found in a computation. These negative concentrations can be suppressed by applying a horizontal Forester filter, Sections 4.5.8 and 10.6.4. However, this can result in a substantially larger computing time. It is suggested to accept small negative concentrations and to apply a Forester filter only when the negative concentrations become unacceptably large.
- ◊ A *vertical* Forester filter applied in a sediment transport computation will not affect the sediments. Since this filter smoothes the vertical profile and thus can have a strong influence on the vertical mixing processes, this vertical filter has been de-activated for sediments.

11.1.3 Effect of sediment on fluid density

By default Delft3D-FLOW uses the empirical relation formulated by [UNESCO \(1981a\)](#) to adjust the density of water in order to take into account varying temperature and salinity. For sediment transport this relation is extended to include the density effect of sediment fractions in the fluid mixture. This is achieved by adding (per unit volume) the mass of all sediment fractions, and subtracting the mass of the displaced water. As a mathematical statement this translates as:

$$\rho_{mix}(S, c^{(\ell)}) = \rho_w(S) + \sum_{\ell=1}^{lsed} c^{(\ell)} \left(1 - \frac{\rho_w(S)}{\rho_s^{(\ell)}} \right) \quad (11.2)$$

where:

$\rho_w(S)$	specific density of water with salinity concentration S [kg/m^3]
S	salinity concentration [ppt]
$\rho_s^{(\ell)}$	specific density of sediment fraction (ℓ) [kg/m^3]
$lsed$	number of sediment fractions

Horizontal density gradients (now also due to differences in sediment concentrations) can create density currents. Vertical density gradients can also have a significant effect on the amount of vertical turbulent mixing present, as discussed below.

You can include or neglect the effect of sediment on the fluid density by setting the DENSIN flag in the morphology input file.



Remark:

- ◊ This option is included as it has been found that a secondary effect of including sediment in the density calculations is a reduction of the flow velocity in the lower computational layers (when compared with a standard logarithmic velocity profile) and a consequent reduction in the computed bed shear stress. This reduction in bed shear stress is particularly pronounced when the $k-\varepsilon$ turbulence closure model is used, and leads to an increase in overall flow velocity and a consequent lowering of the free surface. Our

experience shows that this change in the free surface level (even if very slight) can lead to calibration problems when converting an existing 2DH model to 3D if the model is driven using water level boundary conditions. A simple method of circumventing these problems can be achieved by setting `DENSIN = .false.` which has the effect of preventing the sediment from having any effect on the density of the water/sediment mixture.

11.1.4 Sediment settling velocity

The settling velocity $w_s^{(\ell)}$ for sand and mud are strongly different in formulation; see Sections 11.2.1 and 11.3.1 for details. In high concentration mixtures, the settling velocity of a single particle is reduced due to the presence of other particles. In order to account for this hindered settling effect we follow Richardson and Zaki (1954) and determine the settling velocity in a fluid-sediment mixture as a function of the sediment concentration and the non-hindered settling fall velocity:

$$w_s^{(\ell)} = \left(1 - \frac{c_s^{tot}}{C_{soil}}\right)^5 w_{s,0}^{(\ell)}. \quad (11.3)$$

where C_{soil} is the reference density (input parameter), $w_{s,0}$ is the ‘basic’ sediment fraction specific settling velocity. The total mass concentration c_m^{tot} is the sum of the mass concentrations of the sediment fractions:

$$c_m^{tot} = \sum_{\ell=1}^{l_{sed}} c_s^{(\ell)}. \quad (11.4)$$

As the fall velocity is now a function of the sediment fractions concentration, this implies that each sediment fraction has a fall velocity which is a function of location and time.

Remark:

- ◊ The process of sediment settling is computed with a first-order upwind numerical scheme. While use of the upwind settling formulation does slightly under-predict the mass of sediment settling, the magnitude of this error has been shown to be rather small (Lesser et al., 2000).



11.1.5 Dispersive transport

The eddy diffusivities $\varepsilon_{s,x}^{(\ell)}$, $\varepsilon_{s,y}^{(\ell)}$ and $\varepsilon_{s,z}^{(\ell)}$ depend on the flow characteristics (turbulence level, taking into account the effect of high sediment concentrations on damping turbulent exchange processes) and the influence of waves (due to wave induced currents and enhanced bottom shear stresses). Delft3D-FLOW supports four so-called “turbulence closure models”:

- ◊ Constant coefficient.
- ◊ Algebraic eddy viscosity closure model.
- ◊ $k-L$ turbulence closure model.
- ◊ $k-\varepsilon$ turbulence closure model.

The first is a simple constant value which is specified by you. A constant eddy viscosity will lead to parabolic vertical velocity profiles (laminar flow). The other three turbulence closure models are based on the eddy viscosity concept of Kolmogorov (1942) and Prandtl (1945) and offer zero, first, and second order closures for the turbulent kinetic energy (k) and for the mixing length (L). All three of the more advanced turbulence closure models take into account

the effect that a vertical density gradient has on damping the amount of vertical turbulent mixing. See [section 9.5](#) for a full description of the available turbulence closure models.

The output of a turbulence closure model is the eddy viscosity at each layer interface; from this the vertical sediment mixing coefficient is calculated:

$$\varepsilon_s^{(\ell)} = \beta \varepsilon_f, \quad (11.5)$$

where:

$\varepsilon_s^{(\ell)}$	vertical sediment mixing coefficient for the sediment fraction (ℓ)
β	non-cohesive sediment: Van Rijn's 'beta' factor or effective 'beta' factor. cohesive sediment fractions and fine sand (< 150 μm): 1.0.
ε_f	vertical fluid mixing coefficient calculated by the selected turbulence model.



Remarks:

- ◊ For cohesive sediment fractions the extra turbulent mixing due to waves is not yet included in the eddy diffusivity. This is a limitation of the present implementation. See also [subsection 11.2.2](#).
- ◊ For non-cohesive sediment the effect of waves is accounted for by using a modified or effective 'beta' factor of Van Rijn ($k-\varepsilon$ model) or by using a separate formula to compute ε_f (algebraic or $k-L$) model. See also [subsection 11.3.2](#).

11.1.6 Three-dimensional wave effects

Traditionally wave effects were only incorporated in a depth-averaged manner via a (breaking) wave induced shear stress at the surface, a wave induced mass flux and an increased bed shear stress. Important wave effects such as streaming in the wave boundary layer and wave induced turbulence were not accounted for. The problem of three dimensional wave effects has been studied by [Walstra and Roelvink \(2000\)](#); their main suggestions for improvement are:

- 1 The wave induced mass flux is corrected with the second order Stokes drift.
- 2 The production of turbulent energy associated with wave breaking is incorporated by introducing an extra source term in the kinetic energy and dissipation equations of the $k-\varepsilon$ turbulence model.
- 3 The production of turbulent energy associated with dissipation in the near-bed wave boundary layer is incorporated by introducing an extra source term in the kinetic energy and dissipation equations of the $k-\varepsilon$ turbulence model.
- 4 Streaming (a wave induced current in the wave boundary layer directed in the wave propagation direction) is modelled as a time averaged shear stress.

These improvements have been implemented in Delft3D-FLOW; for full details you are referred to [section 9.7](#). These effects are important when computing the transport of sediment in wave and current situations; see Sections [11.2.5](#) and [11.3.4](#) for full details regarding their effect on cohesive and non-cohesive sediments respectively.

11.1.7 Initial and boundary conditions

To solve [Equation 11.1](#) you need to prescribe initial and boundary conditions for each suspended sediment fraction.

11.1.7.1 Initial condition

The initial conditions for the sediment fractions are handled in exactly the same manner as those for any other conservative constituent, i.e. you can specify:

- ◊ One global initial concentration for each sediment fraction.
- ◊ Space-varying initial concentrations read from a restart file generated by a previous run.
- ◊ Space-varying initial concentrations read from a user-defined input file.

In these options cohesive and non-cohesive sediment fractions are treated in the same way.

In many practical applications the non-cohesive sediment concentrations adapt very rapidly to equilibrium conditions, so in the case of a cold start where the hydrodynamic model also takes some time to stabilise, a uniform zero concentration for the non-cohesive sediment fractions is usually adequate.

11.1.7.2 Boundary conditions

For each of the model boundaries you must prescribe the boundary condition for each sediment fraction. We discuss in short the general type of conditions and refer for the details to the sections to follow.

Water surface boundary

The vertical diffusive flux through the free surface is set to zero for all conservative constituents (except heat, which can cross this boundary). This is left unchanged for suspended sediment.

$$-w_s^{(\ell)} c^{(\ell)} - \varepsilon_{s,z}^{(\ell)} \frac{\partial c^{(\ell)}}{\partial z} = 0, \quad \text{at } z = \zeta \quad (11.6)$$

where $z = \zeta$ is the location of the free surface.

Bed boundary condition

The exchange of material in suspension and the bed is modelled by calculating the sediment fluxes from the bottom computational layer to the bed, and vice versa. These fluxes are then applied to the bottom computational layer by means of a sediment source and/or sink term in each computational cell. The calculated fluxes are also applied to the bed in order to update the bed level. The boundary condition at the bed is given by:

$$-w_s^{(\ell)} c^{(\ell)} - \varepsilon_{s,z}^{(\ell)} \frac{\partial c^{(\ell)}}{\partial z} = D^{(\ell)} - E^{(\ell)}, \quad \text{at } z = z_b \quad (11.7)$$

where:

- | | |
|--------------|--|
| $D^{(\ell)}$ | sediment deposition rate of sediment fraction (ℓ) . |
| $E^{(\ell)}$ | sediment erosion rate of sediment fraction (ℓ) . |

The formulations of $D^{(\ell)}$ and $E^{(\ell)}$ strongly differ for cohesive and non-cohesive sediment; for the details you are referred to Sections 11.2.3 and 11.3.4 respectively.

Open inflow boundaries

Delft3D-FLOW requires you to specify boundary conditions for all conservative constituents at all open inflow boundaries. When modelling in three dimensions you may choose to specify boundary concentrations that have a uniform, linear, or step distribution over the vertical. You may also choose to specify a “Thatcher-Harleman” return time to simulate the re-entry of material that flowed out of the model after the flow reverses direction.

All of these options are also available for sediment constituents, although they are probably more appropriate for fine, cohesive sediment than for sand-sized particles. To assist with modelling coarser material an additional option has been included. This option allows you to specify that, at all open inflow boundaries, the flow should enter carrying all “sand” sediment fractions at their “equilibrium” concentration profiles. This feature has been implemented as a Neumann boundary condition, that is, zero concentration gradient at the boundary. By setting the the sediment concentrations at the boundary equal to those just inside model domain, a near-perfectly adapted flow will enter the domain and very little accretion or erosion should be experienced near the model boundaries. This will generally be the desired situation if the model boundaries are well chosen. This feature can be activated for sand and mud fraction separately by setting NeuBcSand (previously, EqmBc) and/or NeuBcMud to true in the morphology input file.

Open outflow boundaries

No boundary condition is prescribed at outflow boundaries; effectively this means that the dispersive transport of sediment at the outflow boundary is neglected compared to the advective transport.

11.2 Cohesive sediment

11.2.1 Cohesive sediment settling velocity

In salt water cohesive sediment tends to flocculate to form sediment “flocs”, with the degree of flocculation depending on the salinity of the water. These flocs are much larger than the individual sediment particles and settle at a faster rate. In order to model this salinity dependency you must supply two settling velocities and a maximum salinity. The first velocity, $w_{s,0}$, is taken to be the settling velocity of the sediment fraction in fresh water (salinity = 0). The second velocity, $w_{s,max}$, is the settling velocity of the fraction in water having a salinity equal to SALMAX. The settling velocity of the sediment flocs is calculated as follows:

$$w_{s,0}^{(\ell)} = \begin{cases} \frac{w_{s,max}^{(\ell)}}{2} \left(1 - \cos\left(\frac{\pi S}{S_{max}}\right) \right) + \frac{w_{s,f}^{(\ell)}}{2} \left(1 + \cos\left(\frac{\pi S}{S_{max}}\right) \right), & \text{when } S \leq S_{max} \\ w_{s,max}^{(\ell)}, & \text{when } S > S_{max} \end{cases} \quad (11.8)$$

where:

$w_{s,0}^{(\ell)}$	the (non-hindered) settling velocity of sediment fraction (ℓ)
$w_{s,max}^{(\ell)}$	WSM, settling velocity of sediment fraction (ℓ) at salinity concentration SALMAX
$w_{s,f}^{(\ell)}$	WS0, fresh water settling velocity of sediment fraction (ℓ)
S	salinity
S_{max}	SALMAX, maximal salinity at which WSM is specified



Remarks:

- ◊ Modelling turbulence induced flocculation or the break-up of sediment flocs is not yet implemented.

- ◊ The influence of flocculation is disregarded by setting WSM = WS0.

11.2.2 Cohesive sediment dispersion

The vertical mixing coefficient for sediment is equal to the vertical fluid mixing coefficient calculated by the selected turbulence closure model, i.e.:

$$\varepsilon_s^{(\ell)} = \varepsilon_f, \quad (11.9)$$

where:

$\varepsilon_s^{(\ell)}$	vertical sediment mixing coefficient for sediment fraction (ℓ)
ε_f	vertical fluid mixing coefficient calculated by the selected turbulence closure model

11.2.3 Cohesive sediment erosion and deposition

For cohesive sediment fractions the fluxes between the water phase and the bed are calculated with the well-known Partheniades-Krone formulations ([Partheniades, 1965](#)):

$$E^{(\ell)} = M^{(\ell)} S \left(\tau_{cw}, \tau_{cr,e}^{(\ell)} \right), \quad (11.10)$$

$$D^{(\ell)} = w_s^{(\ell)} c_b^{(\ell)} S \left(\tau_{cw}, \tau_{cr,d}^{(\ell)} \right), \quad (11.11)$$

$$c_b^{(\ell)} = c^{(\ell)} \left(z = \frac{\Delta z_b}{2}, t \right), \quad (11.12)$$

where:

$E^{(\ell)}$	erosion flux [$\text{kg m}^{-2} \text{s}^{-1}$]
$M^{(\ell)}$	user-defined erosion parameter EROUNI [$\text{kg m}^{-2} \text{s}^{-1}$]
$S \left(\tau_{cw}, \tau_{cr,e}^{(\ell)} \right)$	erosion step function:

$$S \left(\tau_{cw}, \tau_{cr,e}^{(\ell)} \right) = \begin{cases} \left(\frac{\tau_{cw}}{\tau_{cr,e}^{(\ell)}} - 1 \right), & \text{when } \tau_{cw} > \tau_{cr,e}^{(\ell)}, \\ 0, & \text{when } \tau_{cw} \leq \tau_{cr,e}^{(\ell)}. \end{cases} \quad (11.13)$$

$D^{(\ell)}$	deposition flux [$\text{kg m}^{-2} \text{s}^{-1}$]
$w_s^{(\ell)}$	fall velocity (hindered) [m/s]
$c_b^{(\ell)}$	average sediment concentration in the near bottom computational layer
$S \left(\tau_{cw}, \tau_{cr,d}^{(\ell)} \right)$	deposition step function:

$$S \left(\tau_{cw}, \tau_{cr,d}^{(\ell)} \right) = \begin{cases} \left(1 - \frac{\tau_{cw}}{\tau_{cr,d}^{(\ell)}} \right), & \text{when } \tau_{cw} < \tau_{cr,d}^{(\ell)}, \\ 0, & \text{when } \tau_{cw} \geq \tau_{cr,d}^{(\ell)}. \end{cases} \quad (11.14)$$

τ_{cw}	maximum bed shear stress due to current and waves as calculated by the wave-current interaction model selected by the user; see section 9.7 for full details
$\tau_{cr,e}^{(\ell)}$	user-defined critical erosion shear stress TCEUNI [N/m^2]
$\tau_{cr,d}^{(\ell)}$	user-defined critical deposition shear stress TCDUNI [N/m^2]

**Remark:**

- ◊ Superscript (ℓ) implies that this quantity applies to sediment fraction (ℓ) .

The calculated erosion or deposition flux is applied to the near bottom computational cell by setting the appropriate sink and source terms for that cell. Advection, particle settling, and diffusion through the bottom of the near bottom computational cell are all set to zero to prevent double counting these fluxes.

11.2.4 Interaction of sediment fractions

The following notes hold only in case of multiple sediment fractions. The formulations given in the previous section have been formulated for uniform cohesive sediment beds. However, often the bed will be made up of a range of sediment types and sizes. In such cases the erosion rate will be affected. If the bed stratigraphy is modelled in detail, it may be assumed that the erosion rate is proportional to the availability of the sediment fraction considered in the top-most layer of the bed stratigraphy. On the other hand if the bed stratigraphy is not explicitly included in the model and only the overall characteristics of the local bed composition is known, one must assume either that the bed composition is almost uniform (in which case the erosion rate can again be assumed to be proportional to the bed composition) or that the cohesive sediment fraction considered forms a layer that covers the other sediment fractions (in this case the erosion rate of the cohesive sediment will not be reduced). The former approach is nowadays the default approach for the online-morphology module, but the latter behaviour may be activated by setting the `OldMudFrac` keyword to true in the morphology input file.

**Remarks:**

- ◊ Assuming an erosion rate proportional to the availability of the sediment fraction considered may result in a significant underestimation of the erosion rate if the bed is modelled as a single uniformly mixed layer (default setting) and the mud contents is low.
- ◊ Assuming that the erosion rate is independent of the availability of the sediment fraction considered will lead to an overestimation of the erosion rate. For instance, if the model includes two equal cohesive sediment fractions their total transport rate will be double that of the rate observed in an identical simulation carried out using the total amount of the two sediment fractions in the former simulation.

11.2.5 Influence of waves on cohesive sediment transport

For cohesive sediment fractions the sediment mixing coefficient will still be set following [Equation 11.9](#). This implies that the extra turbulent mixing due to waves will not be included in the suspended sediment transport calculations (for these sediment fractions) except by way of the enhancement of the bed shear stress caused by wave-current interaction, see [section 9.7](#) for details. This is a limitation of the present implementation.

11.2.6 Inclusion of a fixed layer

If the thickness of the sediment layer becomes small then the erosion flux is reduced by a factor f_{FIXFAC} as defined in [subsection 11.4.4](#). This reduction factor is related to the formulations implemented for non cohesive sediment transport (see Sections [11.3.5](#) and [11.4.4](#) for suspended and bedload transport respectively).

11.2.7 Inflow boundary conditions cohesive sediment

Although it is general good advice to locate the open boundaries sufficiently far away from the area of interest, this is not always possible in long-term simulations. In such cases it is desirable to impose some kind of equilibrium boundary conditions. The mud concentrations are in general more loosely coupled to local morphology than the concentrations of coarser non-cohesive sediment fractions; a unique “equilibrium” concentration (profile) does often not exist due to differences in critical shear stresses for erosion and sedimentation. So, Delft3D-FLOW allows for a different approach. For cohesive material you can specify that, at all open inflow boundaries, the flow should enter carrying the same mud concentration as computed in the interior of the model. This feature is enabled by setting `NeuBcMud` in the morphology input file to true (Neumann boundary condition: concentration gradient perpendicular to open boundary equal to zero). Although this option may sometimes be very useful, one must be careful when applying it: the sediment concentration of the incoming flow may take on any value that does not lead to significant deposition in the first grid cell.

By setting `NeuBcMud = false`, the concentrations to be applied at the inflow boundaries are read from the `<*.bcc>` file, which has to be created with the FLOW User Interface. If the parameter is set to true, the values specified in the `<*.bcc>` file are overruled.

11.3 Non-cohesive sediment

For the transport of non-cohesive sediment, [Van Rijn et al. \(2000\)](#) approach is followed by default. You can also specify a number of other transport formulations (see [section 11.5](#))

11.3.1 Non-cohesive sediment settling velocity

The settling velocity of a non-cohesive (“sand”) sediment fraction is computed following the method of [Van Rijn \(1993\)](#). The formulation used depends on the diameter of the sediment in suspension:

$$w_{s,0}^{(\ell)} = \begin{cases} \frac{(s^{(\ell)} - 1)gD_s^{(\ell)2}}{18\nu}, & 65 \mu\text{m} < D_s \leq 100 \mu\text{m} \\ \frac{10\nu}{D_s} \left(\sqrt{1 + \frac{0.01(s^{(\ell)} - 1)gD_s^{(\ell)3}}{\nu^2}} - 1 \right), & 100 \mu\text{m} < D_s \leq 1000 \mu\text{m} \\ 1.1\sqrt{(s^{(\ell)} - 1)gD_s^{(\ell)}}, & 1000 \mu\text{m} < D_s \end{cases} \quad (11.15)$$

where:

$s^{(\ell)}$	relative density $\rho_s^{(\ell)}/\rho_w$ of sediment fraction(ℓ)
$D_s^{(\ell)}$	representative diameter of sediment fraction (ℓ)
ν	kinematic viscosity coefficient of water [m^2/s]

$D_s^{(\ell)}$ is the representative diameter of the suspended sediment given by the user-defined sediment diameter SEDDIA (D_{50} of bed material) multiplied by the user-defined factor FACDSS (see also remarks). This value of $D_s^{(\ell)}$ will be overruled if `IOPSUS=1` and the transport formula of [Van Rijn \(1993\)](#) has been selected, see [subsection 11.5.1](#) for details.

Remark:

- ◊ In the case of non-uniform bed material [Van Rijn \(1993\)](#) concluded that, on the basis of measurements, $D_s^{(\ell)}$ should be in the range of 60 to 100 % of D_{50} of the bed material.



If the bed material is very widely graded (well sorted) consideration should be given to using several sediment fractions to model its behaviour more accurately.

11.3.2 Non-cohesive sediment dispersion

The output of a turbulence closure model is the eddy viscosity at each layer interface; from this the vertical sediment mixing coefficient is calculated using the following expressions:

11.3.2.1 Using the algebraic or k - L turbulence model

Without waves

If the algebraic or k - L turbulence model is selected and waves are inactive then the vertical mixing coefficient for sediment is computed from the vertical fluid mixing coefficient calculated by the selected turbulence closure model. For non-cohesive sediment the fluid mixing coefficient is multiplied by Van Rijn's 'beta factor' which is intended to describe the different diffusivity of a fluid 'particle' and a sand grain. Expressed mathematically:

$$\varepsilon_s^{(\ell)} = \beta \varepsilon_f^{(\ell)}, \quad (11.16)$$

where:

$\varepsilon_s^{(\ell)}$	vertical sediment mixing coefficient for sediment fraction (ℓ)
β	Van Rijn's 'beta' factor for the sediment fraction (see Equation 11.24)
$\varepsilon_f^{(\ell)}$	vertical fluid mixing coefficient calculated by the selected turbulence closure model

Including waves

If waves are included in a simulation using the algebraic or k - L turbulence closure model then the sediment mixing coefficient for non-cohesive sediment fractions is calculated entirely separately from the turbulence closure model, using expressions given by [Van Rijn \(1993\)](#) for both the current-related and wave-related vertical turbulent mixing of sediment.

The current-related mixing is calculated using the 'parabolic-constant' distribution recommended by Van Rijn:

$$\varepsilon_{s,c}^{(\ell)} = \begin{cases} \kappa \beta u_{*,c} z (1 - z/h), & \text{when } z < 0.5h, \\ 0.25 \kappa \beta u_{*,c} h, & \text{when } z \geq 0.5h, \end{cases} \quad (11.17)$$

where:

$\varepsilon_{s,c}^{(\ell)}$	vertical sediment mixing coefficient due to currents (for this sediment fraction)
$u_{*,c}$	current-related bed shear velocity

In the lower half of the water column this expression should produce similar turbulent mixing values to those produced by the algebraic turbulence closure model. The turbulent mixing in the upper half of the water column is generally of little importance to the transport of 'sand' sediment fractions as sediment concentrations in the upper half of the water column are low.

The wave-related mixing is also calculated following [Van Rijn \(1993\)](#). In this case Van Rijn recommends a smoothed step type distribution over the vertical, with a linear transition between the two hinge points, see [Figure 11.1](#).

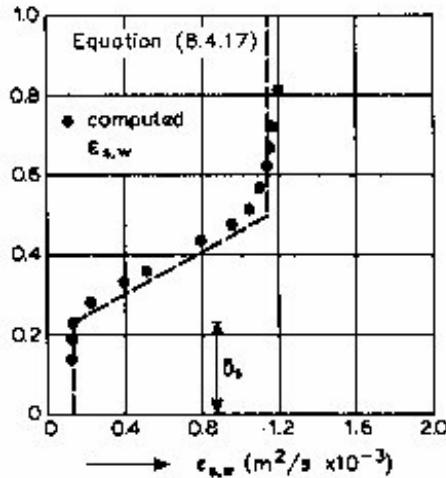


Figure 11.1: Sediment mixing coefficient in non-breaking waves (Source: Van Rijn (1993))

The expressions used to set this distribution are:

$$\varepsilon_{s,w}^{(\ell)} = \begin{cases} \varepsilon_{s,bed}^{(\ell)} = 0.004 D_* \delta_s^{(\ell)} \hat{U}_\delta, & \text{when } z \leq \delta_s^{(\ell)}, \\ \varepsilon_{s,bed}^{(\ell)} + \left(\varepsilon_{s,max}^{(\ell)} - \varepsilon_{s,bed}^{(\ell)} \right) \left(\frac{z - \delta_s^{(\ell)}}{0.5h - \delta_s^{(\ell)}} \right), & \text{when } \delta_s^{(\ell)} < z < 0.5h, \\ \varepsilon_{s,max}^{(\ell)} = \frac{0.035 \gamma_{br} h H_s}{T_p}, & \text{when } z \geq 0.5h, \end{cases} \quad (11.18)$$

where $\delta_s^{(\ell)}$ (the thickness of the near-bed sediment mixing layer) is estimated using Van Rijn's formulation, given by:

$$\delta_s^{(\ell)} = \min [0.5, \max \{0.1, \max (5\gamma_{br}\delta_w, 10\gamma_{br}k_{s,w})\}] \quad (11.19)$$

where:

δ_w thickness of the wave boundary layer:

$$\delta_w = 0.072 \hat{A}_\delta \left(\frac{\hat{A}_\delta}{k_{s,w}} \right)^{-0.25} \quad (11.20)$$

γ_{br} empirical coefficient related to wave breaking:

$$\gamma_{br} = \begin{cases} 1 + \left(\frac{H_s}{h} - 0.4 \right)^{0.5} & \text{when } \frac{H_s}{h} > 0.4 \\ 1 & \text{when } \frac{H_s}{h} \leq 0.4 \end{cases} \quad (11.21)$$

$k_{s,w}$ wave-related bed roughness (as calculated for suspended sediment transport)

We calculate the total vertical sediment mixing coefficient by following Van Rijn and taking the sum of the squares:

$$\varepsilon_s^{(\ell)} = \sqrt{\varepsilon_{s,c}^{(\ell)2} + \varepsilon_{s,w}^{(\ell)2}}, \quad (11.22)$$

where ε_s is the vertical sediment diffusion coefficient used in the suspended sediment transport calculations for this sediment fraction.

11.3.2.2 Using the $k-\varepsilon$ turbulence model

In the case of the $k-\varepsilon$ turbulence closure model the vertical sediment mixing coefficient can be calculated directly from the vertical fluid mixing coefficient calculated by the turbulence closure model, using the following expression:

$$\varepsilon_s^{(\ell)} = \beta_{\text{eff}}^{(\ell)} \varepsilon_f, \quad (11.23)$$

where:

$\varepsilon_s^{(\ell)}$	vertical sediment mixing coefficient of sediment fraction (ℓ)
$\beta_{\text{eff}}^{(\ell)}$	the effective Van Rijn's 'beta' factor of sediment fraction (ℓ) As the beta factor should only be applied to the current-related mixing this is estimated as: $\beta_{\text{eff}}^{(\ell)} = 1 + (\beta^{(\ell)} - 1) \frac{\tau_c}{\tau_w + \tau_c}$, for non-cohesive sediment fractions
$\beta^{(\ell)}$	Van Rijn's 'beta' factor of the sediment fraction (ℓ), Equation 11.24
τ_c	bed shear stress due to currents
τ_w	bed shear stress due to waves
ε_f	vertical fluid mixing coefficient calculated by the $k-\varepsilon$ turbulence closure model

Van Rijn's 'beta' factor is calculated from ([Van Rijn, 1984b](#)):

$$\beta^{(\ell)} = 1 + 2 \left(\frac{w_s^{(\ell)}}{u_{*,c}} \right)^2. \quad (11.24)$$

Where $w_s^{(\ell)}$ is the settling velocity of the non-cohesive sediment fraction, and $u_{*,c}$ is the local bed shear stress due to currents.

This implies that the value of $\beta^{(\ell)}$ is space (and time) varying, however it is constant over the depth of the flow. In addition, due to the limited knowledge of the physical processes involved, we follow [Van Rijn \(1993\)](#) and limit $\beta^{(\ell)}$ to the range $1 < \beta^{(\ell)} < 1.5$.



Remarks:

- ◊ In a wave and current situation [Van Rijn \(1993\)](#) applies the β -factor to only the current-related turbulent mixing, whereas we apply it to the total turbulent mixing calculated by the selected turbulence closure model. However, little is known about the dependence of the β -factor on flow conditions; this discrepancy is expected to be of little importance in practical situations.
- ◊ The $k-\varepsilon$ turbulence closure model has been extended by [Walstra et al. \(2000\)](#) to include the three-dimensional effects of waves. However the effect of wave asymmetry on the bedload transport is not yet included.

11.3.3 Reference concentration

For non-cohesive sediment (e.g. sand), we follow the method of [Van Rijn \(1993\)](#) for the combined effect of waves and currents. The reference height is given by:

$$a = \min \left[\max \left\{ \text{AKSFAC} \cdot k_s, \frac{\Delta_r}{2}, 0.01h \right\}, 0.20h \right], \quad (11.25)$$

where:

a	Van Rijn's reference height
AksFac	user-defined proportionality factor (morphology input file)

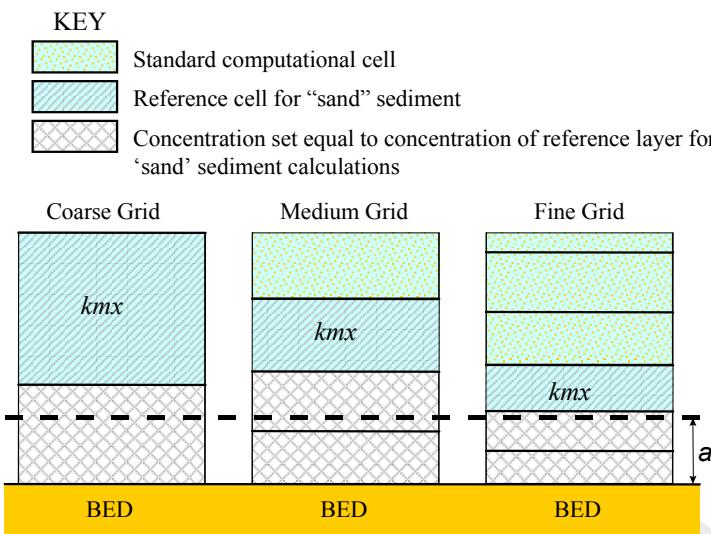


Figure 11.2: Selection of the kmx layer; where a is Van Rijn’s reference height

k_s	user-defined current-related effective roughness height (see options below)
Δ_r	wave-induced ripple height, set to a constant value of 0.025 m
h	water depth

Remark:

- ◊ Van Rijn’s reference height a is limited to a maximum of 20% of the water depth. This precaution is only likely to come into effect in very shallow areas.



With the keyword `IOPKCW` you have two options to calculate k_s (and k_w):

- ◊ k_s and k_w specified by you (constant in space).
- ◊ k_s derived from current-related effective roughness height as determined in the Delft3D-FLOW module (spatially varying) and $k_w = RWAVE \cdot \Delta_r$.



Calculation of the reference concentration

The reference concentration c_a is calculated directly by the sediment transport formula or it is derived from the suspended sediment transport rate given by the sediment transport formula as $c_a = S_s / H_u$. The default transport formula (Van Rijn, 1993) includes a formula for the reference concentration (see subsection 11.5.1). The reference concentration is adjusted proportional to the relative availability of the sediment fraction in the top-layer of the bed (see subsection 11.6.4 on bed composition models).



Remark:

- ◊ The reference concentration and therefore the suspended load can be calibrated using the keyword `Sus` in the morphology input file.

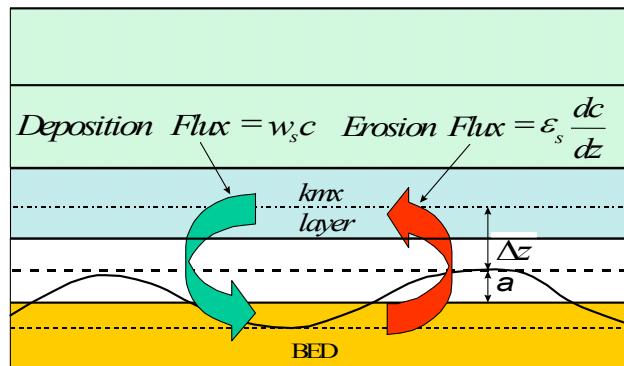


Figure 11.3: Schematic arrangement of flux bottom boundary condition

11.3.4 Non-cohesive sediment erosion and deposition

The transfer of sediment between the bed and the flow is modelled using sink and source terms acting on the near-bottom layer that is entirely above Van Rijn's reference height. This layer is identified as the reference layer and for brevity is referred to as the *k_mx*-layer; see Figure 11.2.

The sediment concentrations in the layer(s) that lie below the *k_mx* layer are assumed to rapidly adjust to the same concentration as the reference layer.

Each half time-step the source and sink terms model the quantity of sediment entering the flow due to upward diffusion from the reference level and the quantity of sediment dropping out of the flow due to sediment settling. A sink term is solved implicitly in the advection-diffusion equation, whereas a source term is solved explicitly. The required sink and source terms for the *k_mx* layer are calculated as follows.

In order to determine the required sink and source terms for the *k_mx* layer, the concentration and concentration gradient at the bottom of the *k_mx* layer need to be approximated. We assume a standard Rouse profile between the reference level *a* and the centre of the *k_mx* layer (see Figure 11.4).

$$c^{(\ell)} = c_a^{(\ell)} \left[\frac{a(h-z)}{z(h-a)} \right]^{A^{(\ell)}}, \quad (11.26)$$

where:

$c^{(\ell)}$	concentration of sediment fraction (ℓ)
$c_a^{(\ell)}$	reference concentration of sediment fraction (ℓ)
a	Van Rijn's reference height
h	water depth
z	elevation above the bed
$A^{(\ell)}$	Rouse number

As the reference concentration and the concentration in the centre of the *k_mx* layer c_{kmx} are known, the exponent $A^{(\ell)}$ can be determined.

$$c_{kmx}^{(\ell)} = c_a^{(\ell)} \left(\frac{a(h-z_{kmx})}{z_{kmx}(h-a)} \right)^{A^{(\ell)}} \Rightarrow A^{(\ell)} = \frac{\ln \left(\frac{c_{kmx}}{c_a} \right)}{\ln \left(\frac{a(h-z_{kmx})}{z_{kmx}(h-a)} \right)} \quad (11.27)$$

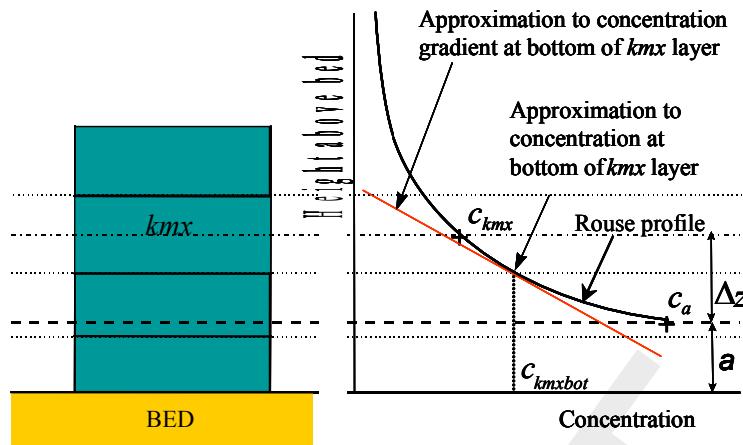


Figure 11.4: Approximation of concentration and concentration gradient at bottom of kmx layer

The concentration at the bottom of the kmx layer is:

$$c_{kmx(bot)}^{(\ell)} = c_a^{(\ell)} \left(\frac{a(h - z_{kmx(bot)})}{z_{kmx(bot)}(h - a)} \right)^{A^{(\ell)}} \quad (11.28)$$

We express this concentration as a function of the known concentration c_{kmx} by introducing a correction factor α_1 :

$$c_{kmx(bot)}^{(\ell)} = \alpha_1^{(\ell)} c_{kmx}^{(\ell)} \quad (11.29)$$

The concentration gradient of the Rouse profile is given by:

$$\frac{\partial c^{(\ell)}}{\partial z} = A^{(\ell)} c_a^{(\ell)} \left(\frac{a(h - z)}{z(h - a)} \right)^{A^{(\ell)} - 1} \cdot \left(\frac{-ah}{z^2(h - a)} \right) \quad (11.30)$$

The concentration gradient at the bottom of the kmx layer is:

$$c'_{kmx(bot)}^{(\ell)} = A^{(\ell)} c_a^{(\ell)} \left(\frac{a(h - z_{kmx(bot)})}{z_{kmx(bot)}(h - a)} \right)^{A^{(\ell)} - 1} \cdot \left(\frac{-ah}{z_{kmx(bot)}^2(h - a)} \right) \quad (11.31)$$

We express this gradient as a function of the known concentrations c_a and c_{kmx} by introducing another correction factor α_2 :

$$c'_{kmx(bot)}^{(\ell)} = \alpha_2^{(\ell)} \left(\frac{c_{kmx}^{(\ell)} - c_a^{(\ell)}}{\Delta z} \right) \quad (11.32)$$

Erosive flux due to upward diffusion

The upward diffusion of sediment through the bottom of the kmx layer is given by the expression:

$$E^{(\ell)} = \varepsilon_s^{(\ell)} \frac{\partial c^{(\ell)}}{\partial z}, \quad (11.33)$$

where $\varepsilon_s^{(\ell)}$ and $\frac{\partial c^{(\ell)}}{\partial z}$ are evaluated at the bottom of the k_{mx} layer.

We approximate this expression by:

$$E^{(\ell)} \approx \alpha_2^{(\ell)} \varepsilon_s^{(\ell)} \left(\frac{c_a^{(\ell)} - c_{k_{mx}}^{(\ell)}}{\Delta z} \right), \quad (11.34)$$

where:

$\alpha_2^{(\ell)}$	correction factor for sediment concentration
$\varepsilon_s^{(\ell)}$	sediment diffusion coefficient evaluated at the bottom of the k_{mx} cell of sediment fraction(ℓ)
$c_a^{(\ell)}$	reference concentration of sediment fraction(ℓ)
$c_{k_{mx}}^{(\ell)}$	average concentration of the k_{mx} cell of sediment fraction(ℓ)
Δz	difference in elevation between the centre of the k_{mx} cell and Van Rijn's reference height: $\Delta z = z_{k_{mx}} - a$

The erosion flux is split in a source and sink term:

$$E^{(\ell)} \approx \frac{\alpha_2^{(\ell)} \varepsilon_s^{(\ell)} c_a^{(\ell)}}{\Delta z} - \frac{\alpha_2^{(\ell)} \varepsilon_s^{(\ell)} c_{k_{mx}}^{(\ell)}}{\Delta z}. \quad (11.35)$$

The first of these terms can be evaluated explicitly and is implemented as a sediment source term. The second can only be evaluated implicitly and is implemented as a (positive) sink term. Thus:

$$\text{Source}_{\text{erosion}}^{(\ell)} = \frac{\alpha_2^{(\ell)} \varepsilon_s^{(\ell)} c_a^{(\ell)}}{\Delta z} \quad (11.36)$$

$$\text{Sink}_{\text{erosion}}^{(\ell)} = \frac{\alpha_2^{(\ell)} \varepsilon_s^{(\ell)} c_{k_{mx}}^{(\ell)}}{\Delta z} \quad (11.37)$$

Deposition flux due to sediment settling

The settling of sediment through the bottom of the k_{mx} cell is given by the expression:

$$D^{(\ell)} = w_s^{(\ell)} c_{k_{mx}(\text{bot})}^{(\ell)}, \quad (11.38)$$

where $w_s^{(\ell)}$ and $c_{k_{mx}(\text{bot})}^{(\ell)}$ are evaluated at the bottom of the k_{mx} layer.

We set:

$$c_{k_{mx}(\text{bot})}^{(\ell)} = \alpha_1^{(\ell)} c_{k_{mx}}^{(\ell)}. \quad (11.39)$$

The deposition flux is approximated by:

$$D^{(\ell)} \approx \alpha_1^{(\ell)} c_{k_{mx}}^{(\ell)} w_s^{(\ell)}. \quad (11.40)$$

This results in a simple deposition sink term:

$$\text{Sink}_{\text{deposition}}^{(\ell)} = \alpha_1^{(\ell)} c_{k_{mx}}^{(\ell)} w_s^{(\ell)}. \quad (11.41)$$

The total source and sink terms is given by:

$$\text{Source}^{(\ell)} = \alpha_2^{(\ell)} c_a^{(\ell)} \left(\frac{\varepsilon_s^{(\ell)}}{\Delta z} \right), \quad (11.42)$$

$$\text{Sink}^{(\ell)} = \left[\alpha_2^{(\ell)} \left(\frac{\varepsilon_s^{(\ell)}}{\Delta z} \right) + \alpha_1^{(\ell)} w_s^{(\ell)} \right] c_{kmx}^{(\ell)}. \quad (11.43)$$

These source and sink terms are both guaranteed to be positive.

11.3.5 Inclusion of a fixed layer

The bedload transport is reduced if the thickness of the sediment layer becomes small (see subsection 11.4.4). The same effect has been implemented as a reduction for the entrainment and deposition terms as well as the equilibrium concentration by a factor f_{FIXFAC} if erosion is expected to occur.

11.3.6 Inflow boundary conditions non-cohesive sediment

Although it is general good advice to locate the open boundaries sufficiently far away from the area of interest, this is not always possible in long-term simulations. In such cases it is desirable to impose some kind of equilibrium boundary conditions. Although equilibrium boundary conditions may be better defined for non-cohesive sediments than for cohesive sediments, we have implemented the open boundary condition in a consistent manner. For non-cohesive suspended material you can specify that, at all open inflow boundaries, the flow should enter carrying the same concentration of sediment as computed in the interior of the model. This feature is enabled by setting `NeuBcSand` in the morphology input file to true (Neumann boundary condition: concentration gradient perpendicular to open boundary equal to zero). This means that the sediment load entering through the boundaries will be near-perfectly adapted to the local flow conditions and very little accretion or erosion should be experienced near the model boundaries. This will generally be the desired situation if the model boundaries are well chosen. This method gives the correct results even when the turbulent mixing profile is clearly non-parabolic.

By setting `NeuBcSand = false`, the concentrations to be applied at the inflow boundaries are read from the `<*.bcc>` file, which has to be created with the FLOW User Interface. If the parameter is set to true, the values specified in the `<*.bcc>` file are overruled. This parameter used to be called `EqmBC`.

11.4 Bedload sediment transport of non-cohesive sediment

Bedload (or, for the simpler transport formulae, total load) transport is calculated for all “sand” and “bedload” sediment fractions by broadly according to the following approach: first, the magnitude and direction of the bedload transport at the cell centres is computed using the transport formula selected (See section 11.5), subsequently the transport rates at the cell interfaces are determined, corrected for bed-slope effect and upwind bed composition and sediment availability.

11.4.1 Basic formulation

For simulations including waves the magnitude and direction of the bedload transport on a horizontal bed are calculated using the transport formula selected assuming sufficient sediment and ignoring bed composition except for e.g. hiding and exposure effects on the critical

shear stresses. The default sediment transport formula is [Van Rijn \(1993\)](#) as documented in [subsection 11.5.1](#).

Some of the sediment transport formulae prescribe the bedload transport direction whereas others predict just the magnitude of the sediment transport. In the latter case the initial transport direction will be assumed to be equal to the direction of the characteristic (near-bed) flow direction. In the case of a depth-averaged simulation, the secondary flow/spiral flow intensity I_s optionally computed by the flow module may be taken into account; the bedload transport direction φ_τ is given by the following formula:

$$\tan(\varphi_\tau) = \frac{v - \alpha_I \frac{u}{U} I_s}{u + \alpha_I \frac{v}{U} I_s} \quad (11.44)$$

in which

$$\alpha_I = \frac{2}{\kappa^2} E_s \left(1 - \frac{\sqrt{g}}{\kappa C} \right) \quad (11.45)$$

where:

g	gravitational acceleration
κ	Von Kármán constant
C	Chézy roughness
U	the depth-averaged velocity
E_s	coefficient to be specified by you as <code>Espir</code> keyword in the morphology input file

The default value of E_s is 0, which implies that the spiral flow effect on the bedload transport direction is not included. The spiral flow effect is of crucial importance in a depth-averaged simulation to get pointbar formation in river bends. This effect is only included for transport formulae that return the bedload transport rate but not its direction, i.e. Engelund & Hansen, Meyer-Peter & Muller, General formula, Van Rijn (1984), Ashida & Michiue and optionally the user-defined formula.

The [Van Rijn \(1993\)](#) formula distinguishes the following transport components that are all treated like bed or total load, i.e. without relaxation effects of an advection diffusion equation:

- ◊ bedload due to currents, S_{bc}
- ◊ bedload due to waves, S_{bw}
- ◊ suspended load due to waves, S_{sw} .

These three transport components can be calibrated independently by using the respective keywords `Bed`, `BedW` and `SusW` in the morphology input file.

11.4.2 Suspended sediment correction vector

The transport of suspended sediment is computed over the entire water column (from $\sigma = -1$ to $\sigma = 0$). However, for “sand” sediment fractions, Van Rijn regards sediment transported below the reference height a as belonging to “bedload sediment transport” which is computed separately as it responds almost instantaneously to changing flow conditions and feels the effects of bed slopes. In order to prevent double counting, the suspended sediment fluxes below the reference height a are derived by means of numerical integration from the suspended transport rates. The opposite of these fluxes are scaled with the upwind sediment availability and subsequently imposed as corrective transport. This suspended load correction is included in the depth-averaged suspended load written to the output files of the program.

11.4.3 Interaction of sediment fractions

The following notes hold only in case of multiple sediment fractions. Sediment fractions may interact in several ways:

- ◊ reference concentrations, erosion rates and sediment transport rates will be reduced proportional to the availability of sediment fraction considered in the bed (less of the fraction available for transport)
- ◊ sediment fractions of different sizes influence each other by means of hiding and exposure: fine sediments hide among coarse sediments and are thereby partly shielded from the main flow while the coarser sediments are more exposed than they would be among other sediments of the same size. This effect is taken into account by increasing the effective critical shear stress for fine sediments while lowering it for coarse sediments. This adjustment is carried out using a multiplicative factor ξ . The following formulations have been implemented:
 - No hiding and exposure correction ($\xi = 1$)
 - Egiazaroff formulation

$$\xi = \left(\frac{^{10}\log 19}{^{10}\log 19 + ^{10}\log(D_i/D_m)} \right)^2. \quad (11.46)$$

- Ashida & Michiue formulation

$$\xi = \begin{cases} 0.8429 \frac{D_m}{D_i} & \text{if } D_i/D_m < 0.38889 \\ \left(\frac{^{10}\log 19}{^{10}\log 19 + ^{10}\log(D_i/D_m)} \right)^2 & \text{otherwise} \end{cases}. \quad (11.47)$$

- Parker, Klingeman & McLean or Soehngen, Kellermann & Loy formulation

$$\xi = \left(\frac{D_m}{D_i} \right)^\alpha. \quad (11.48)$$

where α is given by the ASKLHE keyword in the morphology input file.

- Wu, Wang & Jia formulation

$$\varphi^{(\ell)} = \sum_i \eta^{(i)} \frac{D^{(i)}}{D^{(\ell)} - D^{(i)}} \quad (11.49)$$

$$\xi^{(\ell)} = \left(\frac{1 - \varphi^{(\ell)}}{\varphi^{(\ell)}} \right)^m \quad (11.50)$$

where m is given by the MWWJHE keyword in the morphology input file.

The hiding and exposure effect has been implemented for the following transport formulae containing a critical shear stress: Meyer-Peter & Muller, general formula, Ashida-Michiue and optionally the user-defined formula.

11.4.4 Inclusion of a fixed layer

Inclusion of a fixed layer implies that the quantity of sediment at the bed is finite and may, if excessive erosion occurs, become exhausted and be unavailable to supply sediment to suspended and bedload transport modes. In case the bed is covered by bedforms, the troughs of the bedforms will start to expose the non-erodible layer before sediment runs out completely.

This results in a gradual reduction of the transport capacity over a certain sediment thickness indicated by THRESH. This effect is taken into account in the bedload formulations by comparing the thickness of the sediment layer available at the bed with a user-defined threshold value. If the quantity of sediment available is less than the threshold then the magnitude of the calculated bedload transport vector is reduced as follows:

$$S_b'' = f_{\text{FIXFAC}} S_b'', \quad (11.51)$$

where:

- S_b'' magnitude of the bedload transport vector (before correction for bed slope effects)
- f_{FIXFAC} upwind fixed layer proximity factor: $f_{\text{FIXFAC}} = \frac{\text{DPSED}}{\text{THRESH}}$, limited to the range $0 \leq f_{\text{FIXFAC}} \leq 1$.
- DPSED depth of sediment available at the bed
- THRESH user-defined erosion threshold

The equilibrium suspended load concentration in the sediment pickup term is reduced by the same fixed layer proximity factor (in this case of course the local value and not some upwind value is used since suspended sediment pickup has no associated horizontal direction).

In effect, because of the upwind approach used to transfer the bedload transport components to the U and V velocity points, this method limits the sediment that can leave a computational cell, if the quantity of the sediment at the bed is limited. One implication of the use of this rather simple approach is that a finite (although always less than the user-defined threshold) thickness of sediment is required at the bed if a non-zero magnitude of the bedload transport vector is required.



Remarks:

- ◊ Areas may be initially specified as containing zero bottom sediment if non-erodible areas are required. It is likely that these areas will accrete a little sediment in order to allow an equilibrium bedload transport pattern to develop.
- ◊ This effect has also been included for cohesive and non cohesive suspended sediment as indicated in Sections 11.2.6 and 11.3.5.

11.4.5 Calculation of bedload transport at open boundaries

At open boundaries the user may either prescribe the bed level development or the bedload transport rates. In the latter case the bedload transport rates are known from the model input, whereas in the former case the effective bedload transport rates at the boundary could be derived from the mass balance at the open boundary point. The bed level boundary condition is imposed at the same location where a water level boundary condition is imposed, that is at the grid cell just outside the model domain. A consequence of this approach is that the bed level at the first grid cell inside the model domain will not exactly behave as you imposed, but in general it will follow the imposed behaviour closely. In case of multiple sediment fractions, a boundary condition for the bed composition is also needed at inflow boundaries. See Appendices B.9.2 and B.9 for imposing various morphological boundary conditions.

11.4.6 Bedload transport at U and V velocity points

As the control volume for bed level change calculations is centred on the water level points, see Figure 11.5, the bedload transport vector components are actually required at the U and V velocity points, rather than at the water level points where $S_{b,x}$ and $S_{b,y}$ are calculated. By default, we use a simple “upwind” numerical scheme to set the bedload transport components

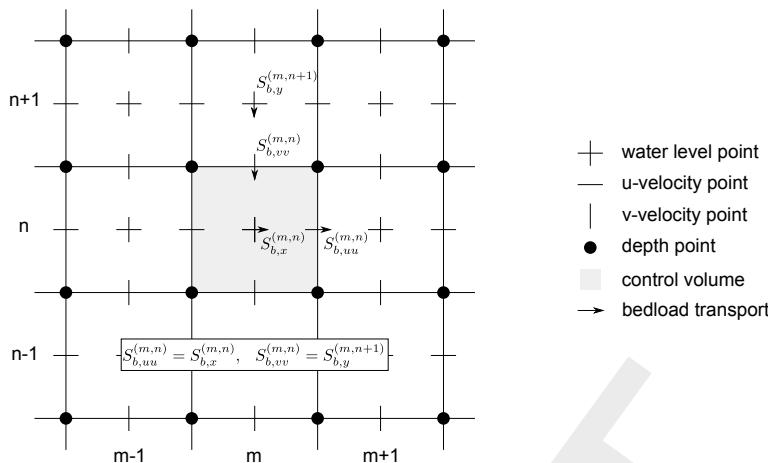


Figure 11.5: Setting of bedload transport components at velocity points

at the U and V points as this ensures that the bed will remain stable. For each active velocity point the upwind direction is determined by summing the bedload transport components at the water level points on either side of the velocity point and taking the upwind *direction* relative to the resulting net transport direction. The bedload transport component at the velocity point is then set equal to the component computed at the water level point immediately “upwind” (see Figure 11.5). In the example shown in Figure 11.5 the bedload transport component $S_{b,uu}^{(m,n)}$ is set equal to $S_{b,x}^{(m,n)}$ and the component $S_{b,vv}^{(m,n)}$ is set equal to $S_{b,y}^{(m,n+1)}$. It is possible to switch from upwind to central approach by setting the `UpwindBedload` keyword in the morphology input file to false; although the central approach is more accurate, it is less stable (less damping).

11.4.7 Adjustment of bedload transport for bed-slope effects

Bedload transport is affected by bed level gradients. Two bed slope directions are distinguished: the slope in the initial direction of the transport (referred to as the longitudinal bed slope) and the slope in the direction perpendicular to that (referred to as the transverse bed slope). The longitudinal bed slope results in a change in the sediment transport rate as given by:

$$\vec{S}'_b = \alpha_s \vec{S}''_b, \quad (11.52)$$

or, in vector component form:

$$S'_{b,x} = \alpha_s S''_{b,x}, \quad (11.53)$$

$$S'_{b,y} = \alpha_s S''_{b,y}, \quad (11.54)$$

whereas the primary effect of the transverse bed slope is a change in transport towards the downslope direction (this may be accomplished by either a pure rotation of the transport vector or by adding a transverse transport component). You may choose one of the following formulations for these effects.

- 1 no effect of bed slope on bedload transport
- 2 [Bagnold \(1966\)](#) for longitudinal slope and [Ikeda \(1982, 1988\)](#) as presented by [Van Rijn \(1993\)](#) for transverse slope. This is the default option for the bedload transport of all sediment transport formulae. In this case α_s is given by

$$\alpha_s = 1 + \alpha_{bs} \left(\frac{\tan(\phi)}{\cos(\tan^{-1}(\frac{\partial z}{\partial s})) (\tan(\phi) + \frac{\partial z}{\partial s})} - 1 \right), \quad (11.55)$$

where α_{bs} is a user-defined tuning parameter, ALFABS keyword in the morphology input file (default = 1.0). An additional bedload transport vector is subsequently calculated, perpendicular to the main bedload transport vector. The magnitude of this vector is calculated using a formulation based on the work of Ikeda (1982, 1988) as presented by Van Rijn (1993). Van Rijn's equation (7.2.52) is modified to [Equation 11.56](#) by setting the reference co-ordinates s and n aligned with and perpendicular to the local characteristic flow direction respectively. This implies that there is no flow in the n direction: i.e. $u_{b,n} = 0$:

$$S_{b,n} = |S'_b| \alpha_{bn} \frac{u_{b,cr}}{|\vec{u}_b|} \frac{\partial z_b}{\partial n}, \quad (11.56)$$

where:

$S_{b,n}$	additional bedload transport vector. The direction of this vector is normal to the unadjusted bedload transport vector, in the down slope direction
$ S'_b $	magnitude of the unadjusted bedload transport vector (adjusted for longitudinal bed slope only): $ S'_b = \sqrt{(S'_{b,x})^2 + (S'_{b,y})^2}$.
α_{bn}	user-defined coefficient, ALFABN (default = 1.5)
$u_{b,cr}$	critical (threshold) near-bed fluid velocity
\vec{u}_b	near-bed fluid velocity vector
$\frac{\partial z_b}{\partial n}$	bed slope in the direction normal to the unadjusted bedload transport vector

To evaluate [Equation 11.56](#) we substitute:

$$\frac{u_{b,cr}}{|\vec{u}_b|} = \sqrt{\frac{\tau_{b,cr}}{|\vec{\tau}_b|}}, \quad (11.57)$$

where:

$\tau_{b,cr}$	critical bed shear stress
$\vec{\tau}_b$	bed shear stress due to current and waves: $\vec{\tau}_b = \mu_c \vec{\tau}_{b,cw} + \mu_w \vec{\tau}_{b,w}$.

resulting in:

$$S_{b,n} = |S'_b| f_{norm}, \quad (11.58)$$

where:

$$f_{norm} = \alpha_{bn} \sqrt{\frac{\tau_{b,cr}}{|\vec{\tau}_b|}} \frac{\partial z_b}{\partial n}. \quad (11.59)$$

The two components of this vector are then added to the two components of the bedload transport vector as follows:

$$\begin{aligned} S_{b,x} &= S'_{b,x} - S'_{b,y} f_{norm} \\ S_{b,y} &= S'_{b,y} + S'_{b,x} f_{norm} \end{aligned} \quad (11.60)$$

where $S_{b,x}$ and $S_{b,y}$ are the components of the required bedload transport vector, calculated at the water level points

- 3 Koch and Flokstra (1980) as extended by Talmon *et al.* (1995). In this case α_s is given by

$$\alpha_s = 1 - \alpha_{bs} \frac{\partial z}{\partial s}, \quad (11.61)$$

where α_{bs} is a user-defined tuning parameter, ALFABS keyword in the morphology input file (default = 1.0). The direction of the bedload is adjusted according to the following formulation:

$$\tan(\varphi_s) = \frac{\sin(\varphi_\tau) + \frac{1}{f(\theta)} \frac{\partial z_b}{\partial y}}{\cos(\varphi_\tau) + \frac{1}{f(\theta)} \frac{\partial z_b}{\partial x}}, \quad (11.62)$$

in which φ_τ is the original direction of the sediment transport and φ_s is the final direction and where $f(\theta)$ equals:

$$f(\theta) = A_{sh} \theta_i^{B_{sh}} \left(\frac{D_i}{H} \right)^{C_{sh}} \left(\frac{D_i}{D_m} \right)^{D_{sh}}, \quad (11.63)$$

where A_{sh} , B_{sh} , C_{sh} and D_{sh} are tuning coefficients specified by you in the morphology input file as keywords Ashld, Bshld, Cshld and Dshld.

- 4 Parker and Andrews (1985). The same formulae for α_s and φ_s hold as in the previous case except for $f(\theta)$ which now equals:

$$f(\theta) = \frac{c_L}{1 + \mu c_L} \sqrt{\frac{\theta}{\max\left(\frac{1}{10}\theta, \xi\theta_{cr}\right)}}, \quad (11.64)$$

where Coulomb friction parameter c_L , lift-drag ratio μ and critical shields parameter θ_{cr} should be specified by you in the morphology input file as keywords CoulFri, F1FdRat and ThetaCr. Note that this formula includes the hiding and exposure factor ξ .

This completes the calculation of the bedload transport field. The transports at the U and V velocity points are then stored for use in the computation of bed level changes, as described in the section 11.6. In all cases the bed slope has been defined as follows.

Longitudinal bed slope

This bed slope is calculated as:

$$\frac{\partial z_b}{\partial s} = \frac{\partial z_{(u)}}{\partial x} \frac{S''_{b,x}}{|S''_b|} + \frac{\partial z_{(v)}}{\partial y} \frac{S''_{b,y}}{|S''_b|}, \quad (11.65)$$

$$\left(\frac{\partial z_b}{\partial s} \right)_{\max} = 0.9 \tan(\phi), \quad (11.66)$$

where:

$\frac{\partial z_b}{\partial s}$	bed slope in the direction of bedload transport
$\frac{\partial z_{(u)}}{\partial x}$	bed slope in the positive x -direction evaluated at the U -point
$\frac{\partial z_{(v)}}{\partial y}$	bed slope in the positive y -direction evaluated at the V -point
ϕ	internal angle of friction of bed material (assumed to be 30°)

Remarks:

- ◊ z_b is the depth down to the bed from a reference height (positive down), a downward bed slope returns a positive value).
- ◊ The bed slope is calculated at the U and V points as these are the locations at which the bedload transport vector components will finally be applied.



Transverse bed slope

This bed slope is calculated as:

$$\frac{\partial z_b}{\partial n} = -\frac{\partial z_{(u)}}{\partial x} \frac{S''_{b,y}}{|S''_b|} + \frac{\partial z_{(v)}}{\partial y} \frac{S''_{b,x}}{|S''_b|}. \quad (11.67)$$

Table 11.1: Additional transport relations

Formula	Bedload	Waves
11.5.1, Van Rijn (1993)	Bedload + suspended	Yes
11.5.2, Engelund-Hansen (1967)	Total transport	No
11.5.3, Meyer-Peter-Muller (1948)	Total transport	No
11.5.4, General formula	Total transport	No
11.5.5, Bijker (1971)	Bedload + suspended	Yes
11.5.6, Van Rijn (1984)	Bedload + suspended	No
11.5.7, Soulsby/Van Rijn	Bedload + suspended	Yes
11.5.8, Soulsby	Bedload + suspended	Yes
11.5.9, Ashida–Michiue (1974)	Total transport	No
11.5.10, Wilcock–Crowe (2003)	Bedload	No
11.5.11, Gaeuman et al. (2009) laboratory calibration	Bedload	No
11.5.12, Gaeuman et al. (2009) Trinity River calibration	Bedload	No

11.5 Transport formulations for non-cohesive sediment

This special feature offers a number of standard sediment transport formulations for non-cohesive sediment. [Table 11.1](#) gives a summary of the available additional formulae.

Additionally, you can implement your own sediment transport formula in a shared library (<dll> or <so>) and call it from Delft3D-FLOW. See [subsection B.9.3](#) for this option. Now, let us continue with a general description of the sediment transport formulae included by default.

11.5.1 Van Rijn (1993)

[Van Rijn \(1993\)](#) distinguishes between sediment transport below the reference height a which is treated as bedload transport and that above the reference height which is treated as suspended-load. Sediment is entrained in the water column by imposing a reference concentration at the reference height.

Reference concentration

The reference concentration is calculated in accordance with [Van Rijn et al. \(2000\)](#) as:

$$c_a^{(\ell)} = 0.015 \rho_s^{(\ell)} \frac{D_{50}^{(\ell)} \left(T_a^{(\ell)} \right)^{1.5}}{a \left(D_*^{(\ell)} \right)^{0.3}} \quad (11.68)$$

where:

$c_a^{(\ell)}$ mass concentration at reference height a

In order to evaluate this expression the following quantities must be calculated:

$D_*^{(\ell)}$ non-dimensional particle diameter:

$$D_*^{(\ell)} = D_{50}^{(\ell)} \left[\frac{(s^{(\ell)} - 1)g}{\nu^2} \right]^{1/3} \quad (11.69)$$

$T_a^{(\ell)}$ non-dimensional bed-shear stress:

$$T_a^{(\ell)} = \frac{(\mu_c^{(\ell)} \tau_{b,cw} + \mu_w^{(\ell)} \tau_{b,w}) - \tau_{cr}^{(\ell)}}{\tau_{cr}^{(\ell)}} \quad (11.70)$$

$\mu_c^{(\ell)}$ efficiency factor current:

$$\mu_c^{(\ell)} = \frac{f'_c^{(\ell)}}{f_c} \quad (11.71)$$

$f'_c^{(\ell)}$ gain related friction factor:

$$f'_c^{(\ell)} = 0.24 \left[10 \log \left(\frac{12h}{3D_{90}^{(\ell)}} \right) \right]^{-2} \quad (11.72)$$

$f_c^{(\ell)}$ total current-related friction factor:

$$f_c^{(\ell)} = 0.24 \left[10 \log \left(\frac{12h}{k_s} \right) \right]^{-2} \quad (11.73)$$

$\tau_{b,cw}$ bed shear stress due to current in the presence of waves. Note that the bed shear velocity u_* is calculated in such a way that Van Rijn's wave-current interaction factor α_{cw} is not required.

$$\tau_{b,cw} = \rho_w u_*^2 \quad (11.74)$$

$\mu_w^{(\ell)}$ efficiency factor waves:

$$\mu_w^{(\ell)} = \max \left(0.063, \frac{1}{8} \left(1.5 - \frac{H_s}{h} \right)^2 \right) \quad (11.75)$$

$\tau_{b,w}$ bed shear stress due to waves:

$$\tau_{b,w} = \frac{1}{4} \rho_w f_w (\hat{U}_\delta)^2 \quad (11.76)$$

f_w total wave-related friction factor (\equiv Equations 9.205, 11.116 and 11.157):

$$f_w = \exp \left[-6 + 5.2 \left(\frac{\hat{A}_\delta}{k_{s,w}} \right)^{-0.19} \right] \quad (11.77)$$

To avoid the need for excessive user input, the wave related roughness $k_{s,w}$ is related to the estimated ripple height, using the relationship:

$$k_{s,w} = RWAVE \cdot \Delta_r, \text{ with } \Delta_r = 0.025 \text{ and } 0.01 \text{ m} \leq k_{s,w} \leq 0.1 \text{ m} \quad (11.78)$$

where:

RWAVE the user-defined wave roughness adjustment factor. Recommended to be in the range 1–3, default = 2.

$\tau_{cr}^{(\ell)}$ critical bed shear stress:

$$\tau_{cr}^{(\ell)} = (\rho_s^{(\ell)} - \rho_w) g D_{50}^{(\ell)} \theta_{cr}^{(\ell)} \quad (11.79)$$

$\theta_{cr}^{(\ell)}$ threshold parameter $\theta_{cr}^{(\ell)}$ is calculated according to the classical Shields curve as modelled by [Van Rijn \(1993\)](#) as a function of the non-dimensional grain size D_* . This avoids the need for iteration.



Note: for clarity, in this expression the symbol D_* has been used where $D_*^{(\ell)}$ would be more correct:

$$\theta_{cr}^{(\ell)} = \begin{cases} 0.24D_*^{-1}, & 1 < D_* \leq 4 \\ 0.14D_*^{-0.64}, & 4 < D_* \leq 10 \\ 0.04D_*^{-0.1}, & 10 < D_* \leq 20 \\ 0.013D_*^{0.29}, & 20 < D_* \leq 150 \\ 0.055, & 150 < D_* \end{cases} \quad (11.80)$$

a	Van Rijn's reference height
\hat{A}_δ	peak orbital excursion at the bed: $\hat{A}_\delta = \frac{T_p \hat{U}_\delta}{2\pi}$.
$D_{50}^{(\ell)}$	median sediment diameter
$D_{90}^{(\ell)}$	90 % sediment passing size: $D_{90}^{(\ell)} = 1.5D_{50}^{(\ell)}$
h	water depth
k_a	apparent bed roughness felt by the flow when waves are present. Calculated by Delft3D-FLOW using the wave-current interaction formulation selected; see section 9.7 for details: $k_a \leq 10k_s$
k_s	user-defined current-related effective roughness height (space varying)
$k_{s,w}$	wave-related roughness, calculated from ripple height, see Equation 11.78
u_z	velocity magnitude taken from a near-bed computational layer. In a current-only situation the velocity in the bottom computational layer is used. Otherwise, if waves are active, the velocity is taken from the layer closest to the height of the top of the wave mixing layer δ .
\hat{U}_δ	peak orbital velocity at the bed: $\sqrt{2} \times RMS_{orbital\ velocity}$ at bed, taken from the wave module.
z_u	height above bed of the near-bed velocity (u_z) used in the calculation of bottom shear stress due to current
Δ_r	estimated ripple height, see Equation 11.78
δ_m	thickness of wave boundary mixing layer following Van Rijn (1993): $3\delta_w$ (and $\delta_m \geq k_a$)
δ_w	wave boundary layer thickness: $\delta_w = 0.072\hat{A}_\delta \left(\frac{\hat{A}_\delta}{k_{s,w}} \right)^{-0.25}.$

We emphasise the following points regarding this implementation:

- ◊ The bottom shear stress due to currents is based on a near-bed velocity taken from the hydrodynamic calculations, rather than the depth-averaged velocity used by Van Rijn.
- ◊ All sediment calculations are based on hydrodynamic calculations from the previous half time-step. We find that this is necessary to prevent unstable oscillations developing.

The apparent roughness felt by the flow (k_a) is dependent on the hydrodynamic wave-current interaction model applied. At this time, Van Rijn's wave-current interaction model is not available in Delft3D-FLOW. This means that it is not possible for a user to exactly reproduce results obtained using Van Rijn's full formulations for waves and currents.

Adjustment of the representative diameter of suspended sediment

The representative diameter of the suspended sediment $D_s^{(\ell)}$ generally given by the user-defined sediment diameter SEDDIA (D_{50} of bed material) multiplied by the user-defined factor FACDSS (see also remarks) can be overruled in case the [Van Rijn \(1993\)](#) transport formula is selected. This achieved by setting IOPSUS=1 the representative diameter of the suspended sediment will then be set to:

$$D_s^{(\ell)} = \begin{cases} 0.64D_{50} & \text{for } T_A^{(\ell)} \leq 1 \\ D_{50}^{(\ell)} \left(1 + 0.015 \left(T_A^{(\ell)} - 25 \right) \right) & \text{for } 1 < T_A^{(\ell)} \leq 25 \\ D_{50}^{(\ell)} & \text{for } 25 < T_A^{(\ell)} \end{cases} \quad (11.81)$$

where $T_a^{(\ell)}$ is given by equation 11.70.

Bedload transport rate

For simulations including waves the magnitude and direction of the bedload transport on a horizontal bed are calculated using an approximation method developed by [Van Rijn et al. \(2003\)](#). The method computes the magnitude of the bedload transport as:

$$|S_b| = 0.006\rho_s w_s D_{50}^{(\ell)} M^{0.5} M_e^{0.7} \quad (11.82)$$

where:

S_b	bedload transport [$\text{kg m}^{-1} \text{s}^{-1}$]
M	sediment mobility number due to waves and currents [-]
M_e	excess sediment mobility number [-]

$$M = \frac{v_{eff}^2}{(s-1) g D_{50}} \quad (11.83)$$

$$M_e = \frac{(v_{eff} - v_{cr})^2}{(s-1) g D_{50}} \quad (11.84)$$

$$v_{eff} = \sqrt{v_R^2 + U_{on}^2} \quad (11.85)$$

in which:

v_{cr}	critical depth averaged velocity for initiation of motion (based on a parameterisation of the Shields curve) [m/s]
v_R	magnitude of an equivalent depth-averaged velocity computed from the velocity in the bottom computational layer, assuming a logarithmic velocity profile [m/s]
U_{on}	near-bed peak orbital velocity [m/s] in onshore direction (in the direction of wave propagation) based on the significant wave height

U_{on} (and U_{off} used below) are the high frequency near-bed orbital velocities due to short waves and are computed using a modification of the method of [Isobe and Horikawa \(1982\)](#). This method is a parameterisation of fifth-order Stokes wave theory and third-order cnoidal wave theory which can be used over a wide range of wave conditions and takes into account the non-linear effects that occur as waves propagate in shallow water ([Grasmeijer and Van Rijn, 1998](#)).

The direction of the bedload transport vector is determined by assuming that it is composed of two parts: part due to current ($S_{b,c}$) which acts in the direction of the near-bed current, and

part due to waves ($S_{b,w}$) which acts in the direction of wave propagation. These components are determined as follows:

$$S_{b,c} = \frac{S_b}{\sqrt{1 + r^2 + 2|r|\cos\varphi}} \quad (11.86)$$

$$|S_{b,w}| = r |S_{b,c}| \quad (11.87)$$

where:

$$r = \frac{(|U_{on}| - v_{cr})^3}{(|v_R| - v_{cr})^3} \quad (11.88)$$

$S_{b,w} = 0$ if $r < 0.01$, $S_{b,c} = 0$ if $r > 100$, and φ = angle between current and wave direction for which [Van Rijn \(2003\)](#) suggests a constant value of 90° .

Also included in the “bedload” transport vector is an estimation of the suspended sediment transport due to wave asymmetry effects. This is intended to model the effect of asymmetric wave orbital velocities on the transport of suspended material within about 0.5 m of the bed (the bulk of the suspended transport affected by high frequency wave oscillations).

This wave-related suspended sediment transport is again modelled using an approximation method proposed by [Van Rijn \(2001\)](#):

$$S_{s,w} = f_{susw} \gamma U_A L_T \quad (11.89)$$

where:

$S_{s,w}$	wave-related suspended transport [kg/(ms)]
f_{susw}	user-defined tuning parameter
γ	phase lag coefficient (= 0.2)
U_A	velocity asymmetry value [m/s] = $\frac{U_{on}^4 - U_{off}^4}{U_{on}^3 + U_{off}^3}$
L_T	suspended sediment load [kg/m ²] = $0.007\rho_s D_{50} M_e$

The three separate transport modes are imposed separately. The direction of the bedload due to currents $S_{b,c}$ is assumed to be equal to the direction of the current, whereas the two wave related transport components $S_{b,w}$ and $S_{s,w}$ take on the wave propagation direction. This results in the following transport components:

$$S_{bc,u} = \frac{u_{b,u}}{|u_b|} |S_{b,c}| \quad (11.90)$$

$$S_{bc,v} = \frac{u_{b,v}}{|u_b|} |S_{b,c}| \quad (11.91)$$

$$S_{bw,u} = S_{b,w} \cos \phi \quad (11.92)$$

$$S_{bw,v} = S_{b,w} \sin \phi \quad (11.93)$$

$$S_{sw,u} = S_{s,w} \cos \phi \quad (11.94)$$

$$S_{sw,v} = S_{s,w} \sin \phi \quad (11.95)$$

where ϕ is the local angle between the direction of wave propagation and the computational grid. The different transport components can be calibrated independently by using the `Bed`, `BedW` and `SusW` keywords in the morphology input file.

11.5.2 Engelund-Hansen (1967)

The Engelund-Hansen sediment transport relation has frequently been used in rivers and estuaries. It reads:

$$S = S_b + S_{s,eq} = \frac{0.05\alpha q^5}{\sqrt{gC^3}\Delta^2 D_{50}} \quad (11.96)$$

where:

q	magnitude of flow velocity
Δ	the relative density $(\rho_s - \rho_w)/\rho_w$
C	Chézy friction coefficient
α	calibration coefficient ($O(1)$)

The transport rate is imposed as bedload transport due to currents S_{bc} . The following formula specific parameters have to be specified in the input files of the Transport module (See subsection B.9.3): calibration coefficient α and roughness height r_k .

Remarks:



- ◊ The D_{50} grain size diameter is based on the sediment fraction considered.
- ◊ A second formula specific input parameter (r_k) is required for the Engelund-Hansen formula. This parameter, which represents the roughness height for currents alone in [m], is only used to determine the C value when the Chézy friction in the flow has not been defined. Generally, this parameter can thus be treated as a dummy parameter.

11.5.3 Meyer-Peter-Muller (1948)

The Meyer-Peter-Muller sediment transport relation is slightly more advanced than the Engelund-Hansen formula, as it includes a critical shear stress for transport. It reads:

$$S = 8\alpha D_{50} \sqrt{\Delta g D_{50}} (\mu\theta - \xi\theta_{cr})^{3/2} \quad (11.97)$$

where:

α	calibration coefficient ($O(1)$)
Δ	the relative density $(\rho_s - \rho_w)/\rho_w$
μ	ripple factor or efficiency factor
θ_{cr}	critical Shields' mobility parameter ($= 0.047$)
ξ	hiding and exposure factor for the sediment fraction considered

and the Shields' mobility parameter θ given by

$$\theta = \left(\frac{q}{C}\right)^2 \frac{1}{\Delta D_{50}} \quad (11.98)$$

in which q is the magnitude of the flow velocity [m/s]. The ripple factor μ reads:

$$\mu = \min \left(\left(\frac{C}{C_{g,90}} \right)^{1.5}, 1.0 \right) \quad (11.99)$$

where $C_{g,90}$ is the Chézy coefficient related to grains, given by:

$$C_{g,90} = 18^{10} \log \left(\frac{12(d + \zeta)}{D_{90}} \right) \quad (11.100)$$

with D_{90} specified in [m]. The transport rate is imposed as bedload transport due to currents S_{bc} . The following formula specific parameters have to be specified in the input files of the Transport module (See [subsection B.9.3](#)): calibration coefficient α and a dummy value.



Remark:

- ◊ The D_{50} is based on the sediment fraction considered, the D_{90} grain size diameters is based on the composition of the local sediment mixture.

11.5.4 General formula

The general sediment transport relation has the structure of the Meyer-Peter-Muller formula, but all coefficients and powers can be adjusted to fit your requirements. This formula is aimed at experienced users that want to investigate certain parameters settings. In general this formula should not be used. It reads:

$$S = \alpha D_{50} \sqrt{\Delta g D_{50}} \theta^b (\mu \theta - \xi \theta_{cr})^c \quad (11.101)$$

where ξ is the hiding and exposure factor for the sediment fraction considered and

$$\theta = \left(\frac{q}{C} \right)^2 \frac{1}{\Delta D_{50}} \quad (11.102)$$

in which q is the magnitude of the flow velocity.

The transport rate is imposed as bedload transport due to currents S_{bc} . The following parameters have to be specified in the input files of the Transport module (See [subsection B.9.3](#)): calibration coefficient α , powers b and c , ripple factor or efficiency factor μ , critical Shields' mobility parameter θ_{cr} .

11.5.5 Bijker (1971)

The Bijker formula sediment transport relation is a popular formula which is often used in coastal areas. It is robust and generally produces sediment transport of the right order of magnitude under the combined action of currents and waves. Bedload and suspended load are treated separately. The near-bed sediment transport (S_b) and the suspended sediment transport (S_s) are given by the formulations in the first sub-section. It is possible to include sediment transport in the wave direction due to wave asymmetry and bed slope following the Bailard approach, see [Bailard \(1981\)](#), [Stive \(1986\)](#). Separate expressions for the wave asymmetry and bed slope components are included:

$$\vec{S}_b = \vec{S}_{b0} + \vec{S}_{b,asymm} + \vec{S}_{s,asymm} + \vec{S}_{b,slope} + \vec{S}_{s,slope} \quad (11.103)$$

$$\vec{S}_s = \vec{S}_{s0} \quad (11.104)$$

where S_{b0} and S_{s0} are the sediment transport in flow direction as computed according to the formulations of Bijker in the first sub-section, and the asymmetry and bed slope components for bedload and suspended transport are defined in the second sub-section. Both bedload and suspended load terms are incorporated in the bedload transport for further processing. The transport vectors are imposed as bedload transport vector due to currents S_{bc} and suspended load transport magnitude S_s , from which the equilibrium concentration is derived, respectively.

11.5.5.1 Basic formulation

The basic formulation of the sediment transport formula according to Bijker is given by:

$$S_b = b D_{50} \frac{q}{C} \sqrt{g} (1 - \phi) \exp(A_r) \quad (11.105)$$

$$S_s = 1.83 S_b \left(I_1 \ln \left(\frac{33.0h}{r_c} \right) + I_2 \right) \quad (11.106)$$

where

C Chézy coefficient (as specified in input of Delft3D-FLOW module)

h water depth

q flow velocity magnitude

ϕ porosity

and

$$A_r = \max(-50, \min(100, A_{ra})) \quad (11.107)$$

$$b = BD + \max \left(0, \min \left(1, \frac{(h_w/h) - C_d}{C_s - C_d} \right) \right) (BS - BD) \quad (11.108)$$

$$I_1 = 0.216 \frac{\left(\frac{r_c}{h}\right)^{z_*-1}}{\left(1 - \frac{r_c}{h}\right)^{z_*}} \int_{r_c/h}^1 \left(\frac{1-y}{y}\right)^{z_*} dy \quad (11.109)$$

$$I_2 = 0.216 \frac{\left(\frac{r_c}{h}\right)^{z_*-1}}{\left(1 - \frac{r_c}{h}\right)^{z_*}} \int_{r_c/h}^1 \ln y \left(\frac{1-y}{y}\right)^{z_*} dy \quad (11.110)$$

where

BS Coefficient b for shallow water (default value 5)

BD Coefficient b for deep water (default value 2)

C_s Shallow water criterion (H_s/h) (default value 0.05)

C_d Deep water criterion (default value 0.4)

r_c Roughness height for currents [m]

and

$$A_{ra} = \frac{-0.27 \Delta D_{50} C^2}{\mu q^2 \left(1 + 0.5 \left(\psi \frac{U_b}{q} \right)^2 \right)} \quad (11.111)$$

$$\mu = \left(\frac{C}{18^{10} \log(12h/D_{90})} \right)^{1.5} \quad (11.112)$$

$$z_* = \frac{w}{\frac{\kappa q \sqrt{g}}{C} \sqrt{1 + 0.5 \left(\psi \frac{U_b}{q} \right)^2}} \quad (11.113)$$

$$U_b = \frac{\omega h_w}{2 \sinh(k_w h)} \quad (11.114)$$

$$\omega = \frac{2\pi}{T} \quad (11.115)$$

$$f_w = \exp \left(-5.977 + \frac{5.123}{a_0^{0.194}} \right) \quad (11.116)$$

(≡ Equations 9.205, 11.77 and 11.157):

$$a_0 = \max \left(2, \frac{U_b}{\omega r_c} \right) \quad (11.117)$$

$$\psi = \begin{cases} C \sqrt{\frac{f_w}{2g}} & \text{if wave effects are included } (T > 0) \\ 0 & \text{otherwise} \end{cases} \quad (11.118)$$

where

C	Chézy coefficient (as specified in input of Delft3D-FLOW module)
h_w	wave height (H_{rms})
k_w	wave number
T	wave period computed by the waves model or specified by you as T user.
U_b	wave velocity
w	sediment fall velocity [m/s]
Δ	relative density $(\rho_s - \rho_w)/\rho_w$
κ	Von Kármán constant (0.41)

The following formula specific parameters have to be specified in the input files of the Transport module (See subsection B.9.3): BS , BD , C_s , C_d , dummy argument, r_c , w , ε and T user.

11.5.5.2 Transport in wave propagation direction (Bailard-approach)

If the Bijker formula is selected it is possible to include sediment transport in the wave direction due to wave asymmetry following the Bailard approach, see [Bailard \(1981\)](#) and [Stive \(1986\)](#). For a detailed description of the implementation you are referred to [Nipius \(1998\)](#).

Separate expressions for the wave asymmetry and bed slope components are included for both bedload and suspended load. Both extra bedload and suspended load transport vectors are added to the bedload transport as computed in the previous sub-section:

$$\vec{S}_b = \vec{S}_{b0} + \vec{S}_{b,asymm} + \vec{S}_{s,asymm} + \vec{S}_{b,slope} + \vec{S}_{s,slope} \quad (11.119)$$

where the asymmetry components for respectively the bedload and suspended transport in wave direction are written as:

$$S_{b;asymm}(t) = \frac{\rho c_f \varepsilon_b}{(\rho_s - \rho) g (1 - \phi) \tan \varphi} |u(t)|^2 u(t) \quad (11.120)$$

$$S_{s;\text{asym}}(t) = \frac{\rho c_f \varepsilon_s}{(\rho_s - \rho) g (1 - \phi) w} |u(t)|^3 u(t) \quad (11.121)$$

from which the components in ξ and η direction are obtained by multiplying with the cosine and sine of the wave angle θ_w and the bed slope components as:

$$S_{b;\text{slope},\xi}(t) = \frac{\rho c_f \varepsilon_b}{(\rho_s - \rho) g (1 - \phi) \tan \varphi} \frac{1}{\tan \varphi} |u(t)|^3 \frac{\partial z_b}{\partial \xi} \quad (11.122)$$

$$S_{s;\text{slope},\xi}(t) = \frac{\rho c_f \varepsilon_s}{(\rho_s - \rho) g (1 - \phi) w} \frac{\varepsilon_s}{w} |u(t)|^5 \frac{\partial z_b}{\partial \xi} \quad (11.123)$$

and similar for the η direction, where:

$u(t)$	near bed velocity signal [m/s]
ρ	density of water [kg/m ³]
ρ_s	density of the sediment [kg/m ³]
c_f	coefficient of the bottom shear stress [-] (constant value of 0.005)
ϕ	porosity [-] (constant value of 0.4)
φ	natural angle of repose [-] (constant value of $\tan \varphi = 0.63$)
w	sediment fall velocity [m/s]
ε_b	efficiency factor of bedload transport [-] (constant value of 0.10)
ε_s	efficiency factor of suspended transport [-] (constant value of 0.02, but in implemented expression for suspended bed slope transport the second ε_s is replaced by a user-defined calibration factor; see Equation 11.126).

These transports are determined by generating velocity signals of the orbital velocities near the bed by using the [Rienecker and Fenton \(1981\)](#) method, see also [Roelvink and Stive \(1989\)](#).

The (short wave) averaged sediment transport due to wave asymmetry, Equations [11.120](#) and [11.121](#), is determined by using the following averaging expressions of the near bed velocity signal (calibration coefficients included):

$$\langle u |u|^2 \rangle = FacA \langle \tilde{u} |\tilde{u}|^2 \rangle + 3FacU \bar{u} \langle |\tilde{u}|^2 \rangle \quad (11.124)$$

$$\langle u |u|^3 \rangle = FacA \langle \tilde{u} |\tilde{u}|^3 \rangle + 4FacU \bar{u} \langle |\tilde{u}|^3 \rangle \quad (11.125)$$

in which:

\tilde{u}	orbital velocity signal
\bar{u}	averaged flow velocity (due to tide, undertow, wind, etc.)
$FacA$	user-defined calibration coefficient for the wave asymmetry
$FacU$	user-defined calibration coefficient for the averaged flow

The suspended transport relation due to the bed slope according to [Equation 11.123](#) is implemented as:

$$S_{s;\text{slope},\xi}(t) = \frac{\rho c_f \varepsilon_s}{(\rho_s - \rho) g (1 - \phi) w} \frac{\varepsilon_{sl}}{w} |u(t)|^5 \frac{\partial z_b}{\partial \xi} \quad (11.126)$$

where:

ε_{sl} user-defined calibration coefficient EpsSL

To activate this transport option, you have to create a separate file named <coef.inp> which contains on three separate lines the calibration coefficients: FacA, FacU and EpsSL. The other parameters are read from the transport input file or are specified as general sediment characteristics.



Note: the user-defined *FacU* value is currently treated as a dummy value, *FacU* = 0.0 will always be used.

A validation study ([Nipius, 1998](#)) showed that the following coefficient settings yielded the best results for the Dutch coast:

$$\begin{aligned} \text{FacA} &= 0.4 \\ \text{FacU} &= 0.0 \\ \text{EpsSL} &= 0.11 \end{aligned}$$

If a relatively straight coast is considered the effect of the parameters is:

- ◊ The wave asymmetry causes onshore directed sediment transport (i.e. in the wave propagation direction). An increased FacA results in an increased onshore transport and hence steepening of the cross-shore bottom profile.
- ◊ The bed slope transport is in general offshore directed. By increasing EpsSL an increased flattening of the bottom profile occurs (i.e. increased offshore transports).
- ◊ The ratio between these parameters determines the balance between onshore and offshore transport and hence the shape and slope of the cross-shore bottom profile. The associated response time of the cross-shore morphology can be influenced by modifying the values of the two parameters, but maintaining a constant ratio. Increased values result in increased gross transports and consequently a reduced morphological response time (and vice versa).

11.5.6 Van Rijn (1984)

The [Van Rijn \(1984a,b,c\)](#) sediment transport relation is a transport formula commonly used for fine sediments in situations without waves. Separate expressions for bedload and suspended load are given. The bedload transport rate is given by:

$$S_b = \begin{cases} 0.053\sqrt{\Delta g D_{50}^3} D_*^{-0.3} T^{2.1} & \text{for } T < 3.0 \\ 0.1\sqrt{\Delta g D_{50}^3} D_*^{-0.3} T^{1.5} & \text{for } T \geq 3.0 \end{cases} \quad (11.127)$$

where T is a dimensionless bed shear parameter, written as:

$$T = \frac{\mu_c \tau_{bc} - \tau_{bcr}}{\tau_{bcr}} \quad (11.128)$$

It is normalised with the critical bed shear stress according to Shields (τ_{bcr}), the term $\mu_c \tau_{bc}$ is the effective shear stress. The formulas of the shear stresses are

$$\tau_{bc} = \frac{1}{8} \rho_w f_{cb} q^2 \quad (11.129)$$

$$f_{cb} = \frac{0.24}{(10 \log(12h/\xi_c))^2} \quad (11.130)$$

$$\mu_c = \left(\frac{18 \log(12h/\xi_c)}{C_{g,90}} \right)^2 \quad (11.131)$$

where $C_{g,90}$ is the grain related Chézy coefficient

$$C_{g,90} = 18^{10} \log \left(\frac{12h}{3D_{90}} \right) \quad (11.132)$$

The critical shear stress is written according to Shields:

$$\tau_{bcr} = \rho_w \Delta g D_{50} \theta_{cr} \quad (11.133)$$

in which θ_{cr} is the Shields parameter which is a function of the dimensionless particle parameter D_* :

$$D_* = D_{50} \left(\frac{\Delta g}{\nu^2} \right)^{\frac{1}{3}} \quad (11.134)$$

The suspended transport formulation reads:

$$S_s = f_{cs} q h C_a \quad (11.135)$$

In which C_a is the reference concentration, q depth averaged velocity, h the water depth and f_{cs} is a shape factor of which only an approximate solution exists:

$$f_{cs} = \begin{cases} f_0(z_c) & \text{if } z_c \neq 1.2 \\ f_1(z_c) & \text{if } z_c = 1.2 \end{cases} \quad (11.136)$$

$$f_0(z_c) = \frac{(\xi_c/h)^{z_c} - (\xi_c/h)^{1.2}}{(1 - \xi_c/h)^{z_c} (1.2 - z_c)} \quad (11.137)$$

$$f_1(z_c) = \left(\frac{\xi_c/h}{1 - \xi_c/h} \right)^{1.2} \ln(\xi_c/h) \quad (11.138)$$

where ξ_c is the reference level or roughness height (can be interpreted as the bedload layer thickness) and z_c the suspension number:

$$z_c = \min \left(20, \frac{w_s}{\beta \kappa u_*} + \phi \right) \quad (11.139)$$

$$u_* = q \sqrt{\frac{f_{cb}}{8}} \quad (11.140)$$

$$\beta = \min \left(1.5, 1 + 2 \left(\frac{w_s}{u_*} \right)^2 \right) \quad (11.141)$$

$$\phi = 2.5 \left(\frac{w_s}{u_*} \right)^{0.8} \left(\frac{C_a}{0.65} \right)^{0.4} \quad (11.142)$$

The reference concentration is written as:

$$C_a = 0.015 \alpha_1 \frac{D_{50}}{\xi_c} \frac{T^{1.5}}{D_*^{0.3}} \quad (11.143)$$

The bedload transport rate is imposed as bedload transport due to currents S_{bc} , while the computed suspended load transport rate is converted into a reference concentration equal to $f_{cs}C_a$. The following formula specific parameters have to be specified in the input files of the Transport module (See subsection B.9.3): calibration coefficient α_1 , dummy argument, reference level (bedload layer thickness) or roughness height ξ_c [m] and settling velocity w_s [m/s].

11.5.7 Soulsby/Van Rijn

The sediment transport relation has been implemented based on the formulations provided in [Soulsby \(1997\)](#). References in the following text refer to this book.

If the wave period T_p is smaller than 10^{-6} s, the wave period T_p is set to 5 s and the root-mean-square wave height is set to 1 cm. Furthermore, the wave period is limited to values larger than 1 s. The root-mean-square wave height is limited to values smaller than $0.4 H$, where H is the water depth.

The sediment transport is set to zero in case of velocities smaller than 10^{-6} m/s, water depth larger than 200 m or smaller than 1 cm.

The root-mean-square orbital velocity is computed as:

$$U_{rms} = \sqrt{2} \frac{\pi H_{rms}}{T_p \sinh(kH)} \quad (11.144)$$

Furthermore, D_* is defined as ([Soulsby, 1997](#), p.104):

$$D_* = \left(\frac{g\Delta}{\nu^2} \right)^{1/3} D_{50} \quad (11.145)$$

Using the critical bed shear velocity according to Van Rijn ([Soulsby, 1997](#), p.176):

$$U_{cr} = \begin{cases} 0.19 D_{50}^{0.1} 10 \log(4H/D_{90}) & \text{if } D_{50} \leq 0.5 \text{ mm} \\ 8.5 D_{50}^{0.6} 10 \log(4H/D_{90}) & \text{if } 0.5 \text{ mm} < D_{50} \leq 2 \text{ mm} \end{cases} \quad (11.146)$$

larger values of D_{50} lead to an error and to the halting of the program.

The sediment transport is split into a bedload and suspended load fraction. The direction of the bedload transport is assumed to be equal to the direction of the depth-averaged velocity in a 2D simulation and equal to the direction of the velocity at the reference height a (see subsection 11.3.3) in a 3D simulation ([Soulsby, 1997](#), p.183):

$$S_{bx} = A_{cal} A_{sb} u \xi \quad (11.147)$$

$$S_{by} = A_{cal} A_{sb} v \xi \quad (11.148)$$

and the suspended transport magnitude is given by the following formula (this quantity is lateron converted to a reference concentration to feed the advection-diffusion equation for the suspended sediment transport as indicated in subsection 11.3.3):

$$S_s = A_{cal} A_{ss} \xi \sqrt{u^2 + v^2} \quad (11.149)$$

where

A_{cal} a user-defined calibration factor
 A_{sb} bedload multiplication factor

$$A_{sb} = 0.005H \left(\frac{D_{50}/H}{\Delta g D_{50}} \right)^{1.2} \quad (11.150)$$

A_{ss} suspended load multiplication factor

$$A_{ss} = 0.012D_{50} \frac{D_*^{-0.6}}{(\Delta g D_{50})^{1.2}} \quad (11.151)$$

ξ a general multiplication factor

$$\xi = \left(\sqrt{U^2 + \frac{0.018}{C_D} U_{rms}^2} - U_{cr} \right)^{2.4} \quad (11.152)$$

where U is the total depth-averaged velocity and C_D is the drag coefficient due to currents, defined by:

$$C_D = \left(\frac{\kappa}{\ln(H/z_0) - 1} \right)^2 \quad (11.153)$$

where z_0 equals 6 mm and the Von Kármán constant κ is set to 0.4.

The bedslope correction factor is not explicitly included in this formula as it is a standard correction factor available in the online morphology module. The method is intended for conditions in which the bed is rippled.

The following formula specific parameters have to be specified in the input files of the Transport module (See [subsection B.9.3](#)): the calibration factor A_{cal} , the ratio of the two characteristic grain sizes D_{90}/D_{50} and the z_0 roughness height.

11.5.8 Soulsby

The sediment transport relation has been implemented based on the formulations provided in [Soulsby \(1997\)](#). References in the following text refer to this book.

If the wave period T_p is smaller than 10^{-6} s, the wave period T_p is set to 5 s and the root-mean-square wave height is set to 1 cm. Furthermore, the wave period is limited to values larger than 1 s. The root-mean-square wave height is limited to values smaller than 0.4 H , where H is the water depth.

The sediment transport is set to zero in case of velocities smaller than 10^{-6} m/s, water depth larger than 200 m or smaller than 1 cm.

The root-mean-square orbital velocity U_{rms} and the orbital velocity U_{orb} are computed as

$$U_{rms} = \sqrt{2}U_{orb} = \sqrt{2} \frac{\pi H_{rms}}{T_p \sinh(kH)} \quad (11.154)$$

For a flat, non-rippled bed of sand the z_0 roughness length is related to the grain size as ([Soulsby, 1997](#), eq.25, p.48) where χ is a user-defined constant:

$$z_0 = \frac{D_{50}}{\chi} \quad (11.155)$$

The relative roughness is characterised using a_* :

$$a_* = \frac{U_{orb} T_p}{z_0} \quad (11.156)$$

which is subsequently used to determine the friction factor of the rough bed according to Swart (1974) (\equiv Equations 9.205, 11.77 and 11.116):

$$f_w = \begin{cases} 0.3 & \text{if } a_* \leq 30\pi^2 \\ 0.00251 \exp(14.1a_*^{-0.19}) & \text{if } a_* > 30\pi^2 \end{cases} \quad (11.157)$$

which corresponds to formulae 60a/b of Soulsby (p.77) using $r = a_*/(60\pi)$ where r is the relative roughness used by Soulsby. The friction factor is used to compute the amplitude of the bed shear-stress due to waves as:

$$\tau_w = 0.5\rho f_w U_{orb}^2 \quad (11.158)$$

Furthermore, the shear stress due to currents is computed as:

$$\tau_c = \rho C_D U^2 \quad (11.159)$$

where

$$C_D = \left(\frac{\kappa}{1 + \ln(z_0/H)} \right)^2 \quad (11.160)$$

as defined on Soulsby (1997, p.53–55). The interaction of the currents and waves is taken into account using the factor Y in the following formula for mean bed shear stress during a wave cycle under combined waves and currents (Soulsby, 1997, p.94):

$$\tau_m = Y (\tau_w + \tau_c) \quad (11.161)$$

The formula for Y is given by:

$$Y = X [1 + bX^p (1 - X)^q] \quad (11.162)$$

where:

$$X = \frac{\tau_c}{\tau_c + \tau_w} \quad (11.163)$$

and b is computed using:

$$b = (b_1 + b_2 |\cos \phi|^J) + (b_3 + b_4 |\cos \phi|^J) \cdot 10 \log(f_w/C_D) \quad (11.164)$$

and p and q are determined using similar equations. In this formula ϕ equals the angle between the wave angle and the current angle, and the coefficients are determined by the model index *modind* and tables 11.2 and 11.3 (related to Soulsby (1997, Table 9, p.91)):

Using the shear stresses given above, the following two Shields parameters are computed:

$$\theta_m = \frac{\tau_m}{\rho g \Delta D_{50}} \text{ and } \theta_w = \frac{\tau_w}{\rho g \Delta D_{50}} \quad (11.165)$$

Table 11.2: Overview of the coefficients used in the various regression models ([Soulsby et al., 1993a](#))

Model	b1	b2	b3	b4	p1	p2	p3	p4
1 (FR84)	0.29	0.55	-0.10	-0.14	-0.77	0.10	0.27	0.14
2 (MS90)	0.65	0.29	-0.30	-0.21	-0.60	0.10	0.27	-0.06
3 (HT91)	0.27	0.51	-0.10	-0.24	-0.75	0.13	0.12	0.02
4 (GM79)	0.73	0.40	-0.23	-0.24	-0.68	0.13	0.24	-0.07
5 (DS88)	0.22	0.73	-0.05	-0.35	-0.86	0.26	0.34	-0.07
6 (BK67)	0.32	0.55	0.00	0.00	-0.63	0.05	0.00	0.00
7 (CJ85)	0.47	0.29	-0.09	-0.12	-0.70	0.13	0.28	-0.04
8 (OY88)	-0.06	0.26	0.08	-0.03	-1.00	0.31	0.25	-0.26

Table 11.3: Overview of the coefficients used in the various regression models, continued ([Soulsby et al., 1993a](#))

Model	q1	q2	q3	q4	J
1 (FR84)	0.91	0.25	0.50	0.45	3.0
2 (MS90)	1.19	-0.68	0.22	-0.21	0.50
3 (HT91)	0.89	0.40	0.50	-0.28	2.7
4 (GM79)	1.04	-0.56	0.34	-0.27	0.50
5 (DS88)	-0.89	2.33	2.60	-2.50	2.7
6 (BK67)	1.14	0.18	0.00	0.00	3.0
7 (CJ85)	1.65	-1.19	-0.42	0.49	0.60
8 (OY88)	0.38	1.19	0.25	-0.66	1.50

Furthermore, D_* is defined as ([Soulsby, 1997](#), p.104):

$$D_* = \left(\frac{g\Delta}{\nu^2} \right)^{1/3} D_{50} \quad (11.166)$$

with which a critical Shields parameter is computed ([Soulsby, 1997](#), eq.77, p.106):

$$\theta_{cr} = \frac{0.30}{1 + 1.2D_*} + 0.055 (1 - \exp(-0.02D_*)) \quad (11.167)$$

The sediment transport rates are computed using the following formulations for normalised transport in current direction and normal direction ([Soulsby, 1997](#), eq.129, p.166/167):

$$\Phi_{x1} = 12(\theta_m - \theta_{cr}) \sqrt{\theta_m + \varepsilon} \quad (11.168)$$

$$\Phi_{x2} = 12(0.95 + 0.19 \cos(2\phi)) \theta_m \sqrt{\theta_w + \varepsilon} \quad (11.169)$$

$$\Phi_x = \max(\Phi_{x1}, \Phi_{x2}) \quad (11.170)$$

$$\Phi_y = \frac{12(0.19\theta_m\theta_w^2 \sin(2\phi))}{(\theta_w + \varepsilon)^{1.5} + 1.5(\theta_m + \varepsilon)^{1.5}} \quad (11.171)$$

where ε is a small constant (10^{-4}) to prevent numerical complications. From these expression are finally the actual bedload transport rates obtained:

$$S_{b,x} = \frac{\sqrt{g\Delta D_{50}^3}}{U} (\Phi_x u - \Phi_y v) \quad (11.172)$$

$$S_{b,y} = \frac{\sqrt{g\Delta D_{50}^3}}{U} (\Phi_x v - \Phi_y u) \quad (11.173)$$

The transport vector is imposed as bedload transport due to currents. The following formula specific parameters have to be specified in the input files of the Transport module (See [subsection B.9.3](#)): calibration coefficient A_{cal} , the model index for the interaction of wave and current forces *modind* (integer number 1 to 8) and the D_{50}/z_0 ratio χ (about 12).

11.5.9 Ashida–Michiue (1974)

The transport rate is given by a generalised version of the Ashida-Michiue formulation:

$$S_{bc} = \alpha \sqrt{\Delta g D_{50}^3} \theta^m \left(1 - \xi \frac{\theta_c}{\theta} \right)^p \left(1 - \sqrt{\xi \frac{\theta_c}{\theta}} \right)^q \quad (11.174)$$

where ξ is the hiding and exposure factor for the sediment fraction considered and:

$$\theta = \left(\frac{q}{C} \right)^2 \frac{1}{\Delta D_{50}} \quad (11.175)$$

in which q is the magnitude of the flow velocity. The transport rate is imposed as bedload transport due to currents S_{bc} . The following formula specific parameters have to be specified in the input files of the Transport module (See [subsection B.9.3](#)): α , θ_c , m , p and q (Ashida and Michiue recommend $\alpha=17$, $\theta_c=0.05$, $m=1.5$, $p=1$ and $q=1$).

11.5.10 Wilcock–Crowe (2003)

The Wilcock-Crowe transport model is a fractional surface based transport model for calculating bedload transport of mixed sand and gravel sediment. The equations and their development are described in [Wilcock and Crowe \(2003\)](#). The bedload transport rate of each size fraction is given by:

$$S_{bi} = \frac{W_i^* F_i U_*^3}{\Delta g} \quad (11.176)$$

$$W_i^* = \begin{cases} 0.002\phi^{7.5} & \text{for } \phi < 1.35 \\ 14 \left(1 - \frac{0.894}{\phi^{0.5}}\right)^{4.5} & \text{for } \phi \geq 1.35 \end{cases} \quad (11.177)$$

$$\phi = \frac{\tau}{\tau_{ri}} \quad (11.178)$$

$$\frac{\tau_{ri}}{\tau_{rm}} = \left(\frac{D_i}{D_m}\right)^b \quad (11.179)$$

$$\tau_{rm} = (0.021 + 0.015 \exp(-20F_s)) (\rho_s - \rho_w) g D_g \quad (11.180)$$

$$b = \frac{0.67}{1 + \exp\left(1.5 - \frac{D_i}{D_g}\right)} \quad (11.181)$$

where:

D_i	D_{50} of size fraction i
D_g	geometric mean grain size of whole grain size distribution
F_i	proportion of size fraction i on the bed surface
F_s	proportion of sand on the bed surface
S_{bi}	bedload transort rate of size fraction i
W_i^*	dimensionless bedload transport rate of size fraction i
Δ	the relative density of the sediment $(\rho_s - \rho_w) / \rho_w$
τ_{ri}	reference shear stress of grains of size D_i
τ_{rm}	reference shear stress of grains of size D_g

Remarks:

- ◊ The Wilcock-Crowe model incorporates its own hiding function so no external formulation should be applied.
- ◊ The roughness height used for the calculation of grain shear stress during the development of the Wilcock-Crowe transport model was $k_s = 2D_{65}$.
- ◊ This sediment transport formula does not have any input parameters that can be, or need to be, tuned.



11.5.11 Gaeuman et al. (2009) laboratory calibration

The Gaeuman et al. sediment transport model is a modified form of the Wilcock-Crowe model which uses the variance of grain size distribution on the phi scale (σ_ϕ^2) rather than the fraction of sand on the bed surface (F_s) as a measure of the bed surface condition for use in the calculation of reference shear stress. The 'laboratory calibration' implementation of the Gaeuman et al. transport model is calibrated to the experimental data used in the derivation of the Wilcock-Crowe transport model. The model, it's derivation and calibration is described in [Gaeuman et al. \(2009\)](#).

The formulae for the calculation of S_{bi} , W_i^* , ϕ and τ_{ri} are the same as for the Wilcock-Crowe transport model (Equations 11.176, 11.177, 11.178 and 11.179) but the calculation of τ_{rm}

and b differs.

$$\tau_{rm} = \left(\theta_{c0} + \frac{0.015}{1 + \exp(10.1\sigma_\phi^2 - 14.14)} \right) (\rho_s - \rho_w) g D_g \quad (11.182)$$

$$b = \frac{1 - \alpha_0}{1 + \exp\left(1.5 - \frac{D_i}{D_g}\right)} \quad (11.183)$$

$$\sigma_\phi^2 = \sum_{i=1}^n \left(\log\left(\frac{D_i}{D_g}\right) \right)^2 F_i \quad (11.184)$$

where θ_{c0} and α_0 are user specified parameters (See subsection B.9.3). If the values $\theta_{c0} = 0.021$ and $\alpha_0 = 0.33$ are specified the original relation calibrated to the Wilcock-Crowe laboratory data is recovered.



Remark:

- ◊ The Gaeuman et al. model incorporates its own hiding function so no external formulation should be applied.

11.5.12 Gaeuman et al. (2009) Trinity River calibration

The 'Trinity River calibration' implementation of the Gaeuman et al. transport model is calibrated to observed bedload transport rates in the Trinity River, USA and is described in [Gaeuman et al. \(2009\)](#). It differs from the 'laboratory calibration' implementation in the calculation of τ_{rm} and b .

$$\tau_{rm} = \left(\theta_{c0} + \frac{0.022}{1 + \exp(7.1\sigma_\phi^2 - 11.786)} \right) (\rho_s - \rho_w) g D_g \quad (11.185)$$

$$b = \frac{1 - \alpha_0}{1 + \exp\left(1.9 - \frac{D_i}{3D_g}\right)} \quad (11.186)$$

where θ_{c0} and α_0 are user specified parameters (See subsection B.9.3). If the values $\theta_{c0} = 0.03$ and $\alpha_0 = 0.3$ are specified the original Gaeuman et al. formulation calibrated to the Trinity River is recovered.



Remark:

- ◊ The Gaeuman et al. model incorporates its own hiding function so no external formulation should be applied.

11.6 Morphological updating

The elevation of the bed is dynamically updated at each computational time-step. This is one of the distinct advantages over an offline morphological computation as it means that the hydrodynamic flow calculations are always carried out using the correct bathymetry.

At each time-step, the change in the mass of bed material that has occurred as a result of the sediment sink and source terms and transport gradients is calculated. This change in mass is then translated into a bed level change based on the dry bed densities of the various sediment fractions. Both the bed levels at the cell centres and cell interfaces are updated.



Remark:

- ◊ The depths stored at the depth points (which are read directly from the bathymetry specified as input) are only updated for writing to the communication file and the result files.

A number of additional features have been included in the morphological updating routine in order to increase the flexibility. These are discussed below.

Morphological “switch”

You can specify whether or not to update the calculated depths to the bed by setting the `MorUpd` (or equivalently `BedUpd`) flag in the morphology input file. It may be useful to turn bottom updating off if only the initial patterns of erosion and deposition are required, or an investigation of sediment transport patterns with a constant bathymetry is required.

Remark:

- ◊ The use of `MorUpd` or `BedUpd` only affects the updating of the depth values (at ζ and velocity points); the amount of sediment available in the bed will still be updated. Use the `CmpUpd` flag to switch off the updating of the bed composition. If you wish to prevent any change in both the bottom sediments *and* flow depths from the initial condition then this may also be achieved by either setting the morphological delay interval `MorStt` to a value larger than the simulation period, or by setting the morphological factor `MorFac` to 0. See below for a description of these two user variables.



Morphological delay

Frequently, a hydrodynamic simulation will take some time to stabilise after transitioning from the initial conditions to the (dynamic) boundary conditions. It is likely that during this stabilisation period the patterns of erosion and accretion that take place do not accurately reflect the true morphological development and should be ignored. This is made possible by use of `MorStt` whereby you can specify a time interval (in minutes after the start time) after which the morphological bottom updating will begin. During the `MorStt` time interval all other calculations will proceed as normal (sediment will be available for suspension for example) however the effect of the sediment fluxes on the available bottom sediments will not be taken into account.

Morphological time scale factor

One of the complications inherent in carrying out morphological projections on the basis of hydrodynamic flows is that morphological developments take place on a time scale several times longer than typical flow changes (for example, tidal flows change significantly in a period of hours, whereas the morphology of a coastline will usually take weeks, months, or years to change significantly). One technique for approaching this problem is to use a “morphological time scale factor” whereby the speed of the changes in the morphology is scaled up to a rate that it begins to have a significant impact on the hydrodynamic flows. This can be achieved by specifying a non-unity value for the variable `MorFac` in the morphology input file.

Remark:

- ◊ The *Morphological scale factor* can also be time-varying, see [subsection B.9.8](#). This feature is not yet supported by the GUI. You have to edit the `<*.mor>` file manually.



The implementation of the morphological time scale factor is achieved by simply multiplying the erosion and deposition fluxes from the bed to the flow and vice-versa by the `MorFac`-factor, at each computational time-step. This allows accelerated bed-level changes to be incorporated dynamically into the hydrodynamic flow calculations.

While the maximum morphological time scale factor that can be included in a morphodynamic model without affecting the accuracy of the model will depend on the particular situation being modelled, and will remain a matter of judgement, tests have shown that the computations

remain stable in moderately morphologically active situations even with MorFac-factors in excess of 1 000. We also note that setting MorFac=0 is often a convenient method of preventing both the flow depth and the quantity of sediment available at the bottom from updating, if an investigation of a steady state solution is required.

**Remarks:**

- ◊ Verify that the morphological factor that you use in your simulation is appropriate by varying it (e.g. reducing it by a factor of 2) and verify that such changes do not affect the overall simulation results.
- ◊ The interpretation of the morphological factor differs for coastal and river applications. For coastal applications with tidal motion, the morphological variations during a tidal cycle are often small and the hydrodynamics is not significantly affected by the bed level changes. By increasing the morphological factor to for instance 10, the morphological changes during one simulated tidal cycle are increased by this factor. From a hydrodynamical point of view this increase in morphological development rate is allowed if the hydrodynamics is not significantly influenced. In that case the morphological development after one tidal cycle can be assumed to represent the morphological development that would in real life only have occurred after 10 tidal cycles. In this example the number of hydrodynamic time steps required to simulate a certain period is reduced by a factor of 10 compared to a full 1:1 simulation. This leads to a significant reduction in simulation time. However, one should note that by following this approach the order of events is changed, possible conflicts may arise in combination with limited sediment availability and bed stratigraphy simulations. In river applications there is no such periodicity as a tidal cycle. For such applications, the morphological factor should be interpreted as a speed-up factor for morphological development without changing the order of events. Effectively, it means that the morphological development is simulated using a, for instance 10 times, larger time step than the hydrodynamics, or phrased more correctly the hydrodynamics is simulated at a 10 times faster rate. This means that in case of time-varying boundary conditions (e.g. river hydrograph) the time-scale of these forcings should be sped up: a 20 day flood peak will be compressed in 2 days. However, one should take care that by speeding up the hydrodynamic forcings one does not substantially change the nature of the overall hydrodynamic and morphological development: a quasi-steady flood period should not become a short, dynamic flash flood. For river applications, changing the morphological factor must be associated with changing all external time-varying forcings. For coastal applications only the overall simulation time should be adjusted. Note that the combination of a river-like flood peak and a tidal motion will cause problems when interpreting morphological factor not equal to 1.
- ◊ The effect of the morphological factor is different for bed and suspended load. At each time step bedload is picked-up from the bed and deposited on the bed: only the transports are increased by the morphological factor used for the time step considered. However, in case of suspended load there is a time-delay between the time of erosion and the time of deposition. The erosion and deposition fluxes are increased by the morphological factor, but the suspended concentrations are not (since that would influence the density effects). It is possible to vary the morphological factor during a simulation to speed up relatively quiet periods more than relatively active periods. Such changes in the morphological factor will not influence the mass balance of a bed or total load simulation since pickup and deposition are combined into one time step. However, in case of suspended load the entrainment and deposition may occur at time-steps governed by different morphological factors. In such cases the entrainment flux that generated a certain suspended sediment concentration will differ from the deposition flux that was caused by the settling of the same suspended sediment. A change in morphological factor during a period of non-zero suspended sediment concentrations, will thus lead to a mass-balance error in the order of the suspended sediment volume times the change

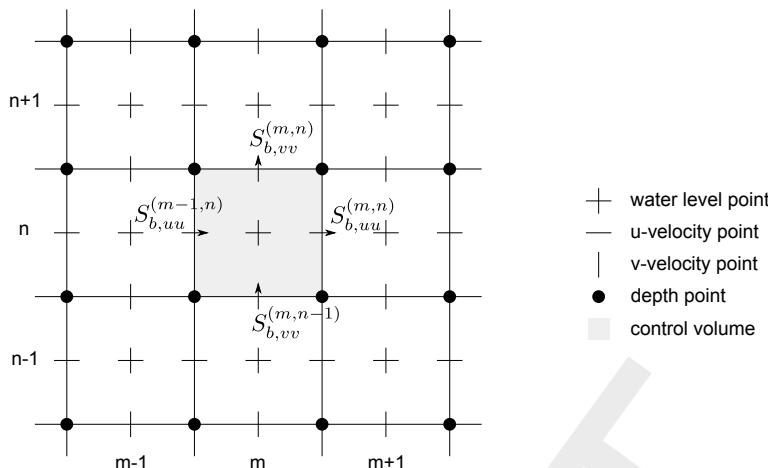


Figure 11.6: Morphological control volume and bedload transport components

in morphological factor. The error may be kept to a minimum by appropriately choosing the transition times.

11.6.1 Bathymetry updating including bedload transport

The change in the quantity of bottom sediments caused by the bedload transport is calculated using the expression:

$$\Delta_{SED}^{(m,n)} = \frac{\Delta t f_{MORFAC}}{A^{(m,n)}} \left(\begin{array}{l} S_{b,uu}^{(m-1,n)} \Delta y^{(m-1,n)} - S_{b,uu}^{(m,n)} \Delta y^{(m,n)} + \\ S_{b,vv}^{(m,n-1)} \Delta x^{(m,n-1)} - S_{b,vv}^{(m,n)} \Delta x^{(m,n)} \end{array} \right), \quad (11.187)$$

where:

$\Delta_{SED}^{(m,n)}$	change in quantity of bottom sediment at location (m, n) [kg/m^2]
Δt	computational time-step [s]
f_{MORFAC}	user-defined morphological acceleration factor, MORFAC
$A^{(m,n)}$	area of computational cell at location (m, n) [m^2]
$S_{b,uu}^{(m,n)}$	computed bedload sediment transport vector in u direction, held at the u point of the computational cell at location (m, n) [$\text{kg}/(\text{m s})$]
$\Delta x^{(m,n)}$	cell width in the x direction, held at the V point of cell (m, n) [m]
$\Delta y^{(m,n)}$	cell width in the y direction, held at the U point of cell (m, n) [m]

This calculation is repeated for all ‘sand’ and ‘bedload’ sediment fractions, if more than one is present, and the resulting change in the bottom sediment mass is added to the change due to the suspended sediment sources and sinks and included in the bed composition and bed level updating scheme.

11.6.2 Erosion of (temporarily) dry points

In the case of erosion near a dry beach or bank, the standard scheme will not allow erosion of the adjacent cells, even when a steep scour hole would develop right next to the beach. Therefore a scheme has been implemented that allows the (partial) redistribution of an erosion flux from a wet cell to the adjacent dry cells. The distribution is governed by a user-defined factor $\Theta_{er,SD}$, which determines the fraction of the erosion to assign (evenly) to the adjacent cells. If $\Theta_{er,SD}$ equals zero the standard scheme is used, i.e. all erosion occurs at the wet cell. If $\Theta_{er,SD}$ equals 1 all erosion that would occur in the wet cell is assigned to the adjacent dry cells. The ‘wet’ and ‘dry’ cells in the paragraph above are defined as cells at which the

water depths are, respectively, above and below the threshold depth `SedThr` for computing sediment transport.

A modification to this method may be activated by specifying a parameter `HMaxTH` larger than the threshold depth `SedThr` for computing sediment transport. In this case, the factor `ThetSD` is used as upper limit for the fraction of the erosion to be transferred to adjacent dry cells. The actual factor to be transferred is equal to `Thet`, which is computed as:

$$\text{Thet} = (h_1 - \text{SedThr}) / (\text{HMaxTH} - \text{SedThr}) \times \text{ThetSD}$$

where $\text{Thet} = \min(\text{Thet}, \text{ThetSD})$ (11.188)

Here, h_1 is the local water depth. The purpose of this formulation is to allow erosion of parts that are inactive in terms of transport but still wet, while limiting the erosion of the dry beach. If erosion of the dry beach is desired, this option is not recommended, so `HMaxTH` should be set less than `SedThr`.



Remark:

- ◊ The overall erosion flux is redistributed to the adjacent cells. Depending on the availability of individual sediment fractions at the central ‘wet’ cell and the surrounding ‘dry’ cells, the erosion from the adjacent cells will replenish the eroded cell with different sediment fractions than those that were eroded.

11.6.3 Dredging and dumping

If the bed levels are updated, you may also include some dredging and dumping activities at the end of each half time step. This feature can also be used for sand mining (only dredging, no associated dumping within the model domain) and sediment nourishment (only dumping, no associated dredging within the model domain). Dredging and dumping is performed at this stage in the following order:

- ◊ For each dredge area: if the bed level exceeds a threshold level (or the water depth drops below a certain level) then the bed level is lowered based on the dredging option and the corresponding volume of sediment is removed. If the dredging capacity is less than the volume to be dredged, the sequence of dredging (e.g. top first or uniform) determines which grid cells are dredging at the current point in time.
- ◊ The volume of dredged material is summed over all cells in a dredge area and distributed over the dump areas, using the link percentages or the link order (up to the dump capacity). In simulations with multiple sediment fractions the sediment composition is tracked.
- ◊ For each dump area: the bed level is raised and the bed composition is adjusted based on the volume and characteristics of material to be dumped. The sediment may be distributed equally or non-uniformly (e.g. deepest points first) over the grid cells in the dump area.



Remark:

- ◊ Dredging and dumping may also be performed during initialization, before the first time-step.



Warning:

- ◊ Dredging large amounts of material may harm the stability of the calculation.

The dredging and dumping feature allows you to specify dredging and dumping areas as x,y polygons. Within each dredging polygon the bed levels are lowered to a user-defined depth; by default grid cells are considered to lie within a polygon if their centre lies within the polygon. It is possible to distribute the dredged material over multiple dumping locations.

You may also decide to not dump the sediment back into the model (feature referred to as sand mining); this can be implemented by defining a dump polygon outside the grid, or by not specifying any dump polygon at all. This option cannot be combined with the option to dredge only as much as dump capacity is available. For sediment nourishment one should use a [nourishment] block specifying the amount (and, if applicable, the composition) of the nourished sediment. The dredging and dumping activities should be specified in a <*.dad> file; for a description of this attribute file see [subsection A.2.23](#). The <*.mdf> file should contain a keyword `Fildad` referring to the file used. The <*.dad> file refers to the file containing the polygons.

11.6.4 Bed composition models and sediment availability

The morphology module currently implements two bed composition models:

- ◊ A uniformly mixed bed (one sediment layer). There is no bookkeeping of the order in which sediments are deposited and all sediments are available for erosion.
- ◊ A layered bed stratigraphy (multiple sediment layers). A user-defined number of bed composition bookkeeping layers may be included to keep track of sediment deposits. When sediments are deposited, they are initially added to the top-most layer. After mixing in the top layer, sediments are pushed towards the bookkeeping layers beneath it. The bookkeeping layers are filled up to a user-defined maximum thickness, if this threshold is exceeded a new layer is created. If the creation of a new layer would exceed the maximum number of layers specified by you, layers at the bottom of the stratigraphy stack will be merged. Only sediments in the top-most layer are available for erosion. After erosion, the top-most layer is replenished from below.

The default bed composition model is the uniformly mixed one. Currently only the default bed composition model is supported by the user interface. See [subsection B.9.2](#) on how to select the other bed composition model.

At input you must specify the amount of sediment available at the bed as the total (dry) mass of all sediment fractions in [kg/m^2]. This may be a constant value for the entire model or, alternatively, a space-varying initial sediment file (values to be specified at cell centres). The initial bed composition is assumed to be uniformly mixed.¹ The thickness of the total sediment layer is calculated from the sediment mass by dividing by the user-defined dry bed density `CDryB`. Currently, `CDryB` is constant in time and space for each individual sediment fraction. The top of these sediment deposits will coincide with the initial bed level. Below the bottom of these deposits the model assumes a non-erodible bed (sometimes referred to as a fixed layer).

When the model almost runs out of sediment at a particular location, the sediment flux terms will be reduced. The reduction starts when the available sediment thickness drops below a user-defined threshold `Thresh`. The flux terms affected are slightly different for cohesive and non-cohesive sediments, as described below.

¹The uniformly mixed bed can be used as input for both bed composition models. If you have more detailed information on the bed stratigraphy, you may use the bed stratigraphy model and specify an initial layering of the bed composition by means of the `IniComp` keyword (see [subsection B.9.2](#)) and associated initial bed composition file (see [subsection B.9.9](#)). In that case the bed composition given in the <*.sed> file will be overruled, you have to specify dummy values though.

Cohesive sediment fractions

In the case of cohesive sediment, the erosive sediment source term is reduced proportionally to the ratio of available sediment thickness over Thresh. The deposition term is never reduced.

Non-cohesive sediment fractions

In the case of non-cohesive sediments all bedload transport rates out of a grid cell are reduced by the upwind ratio of available sediment thickness over Thresh. The source and sink terms of the advection-diffusion equation are not reduced unless the erosive sediment source term is predicted to be larger than the deposition (sink) term, in that case both terms are reduced by the ratio of available sediment thickness over Thresh as shown by the following equations:

$$\text{Source}_{total} = \text{Source}_{total} * f_r, \quad (11.189)$$

$$\text{Sink}_{total} = \text{Sink}_{total} * f_r, \quad (11.190)$$

where f_r is a reduction factor determined by:

$$f_r = \min \left(\frac{\Delta_{sed}}{\text{Thresh}}, 1 \right), \quad (11.191)$$

where Δ_{sed} is the thickness of sediment at the bed.

The likelihood of erosive conditions occurring is assessed by calculating the total sediment source and sink terms using the concentration from the previous time-step to evaluate the implicit sink term. If the sink term is greater than the source term, then deposition is expected, and f_r is set to 1.0 so that deposition close to the fixed layer is not hindered.

11.7 Specific implementation aspects

Negative water depth check

In rare situations (with high morphological acceleration factors) it is possible that, in one time-step, the bed accretes more than the water depth. If this occurs the water depth will become negative (water surface level is below the bed level). This situation is checked for and, if it occurs, the water surface level for the cell is set equal to the new bed level. The cell will then be set dry.

Threshold depth for sediment calculations introduced

If the water depth in a cell is less than SedThr, specified in the morphology input file, then the sediment source and sink terms for the suspended sediment fractions ('sand' and 'mud') will be set to zero (such that neither erosion nor resuspension will occur there). Note that the advection diffusion equation may bring suspended sediment into such cells but it will not interact with the bed; the only way for this sediment to leave the water column is by an increase in water depth (such that interaction is allowed) or by advective/diffusive transport out of the shallow cell. Furthermore, the bedload transport into (and out of) these cells for 'sand' and 'bedload' sediment fractions will be set to zero (bed load transport for 'mud' fractions is zero always). This restriction has been included in order to prevent numerical problems during the computation of the reference concentration, e.g. to prevent sudden bursts of sediment from occurring when computational cells are flooded.



Remark:

- ◊ In areas with very shallow water depths and sediment sources and sinks, you must ensure that the user-defined threshold depth for drying and flooding, see subsection 4.5.8, is not set too large.

Calculation of bed shear in wave and current situations altered

The calculation of the bed shear velocity u_* has been simplified in situations with waves and currents. The bed shear is always calculated using the velocities computed in the bottom computational layer, rather than using the computational layer closest to the top of the sediment mixing layer. The reference velocity in the bottom computational layer is adjusted to the top of the sediment mixing layer using the apparent bed roughness k_a before being used to compute the bed shear velocity using the physical bed roughness k_s .

Depth at grid cell faces (velocity points)

During a morphological simulation the depth stored at the U and V velocity points must be updated to reflect the bed level changes calculated in the water level points. This *used to be* performed by setting the new depth for the velocity point by copying the new depth held at the water level point, using a simple upwind numerical scheme. As this may introduce instabilities in the flow computation, especially near drying and flooding and in tidal simulations, this method has been replaced by setting the depth at U and V points equal to the *minimum* of the adjacent depths in water level points. This change significantly improves the smoothness of flooding dry cells.

Remarks:

- ◊ The setting of depths at velocity points as the minimum of the adjacent water level points only comes into effect if sediment is present and the user-defined flag MORUPD is `.true.` (i.e. bathymetrical changes are expected to occur at some point during the simulation period). If this condition is not met then the depths at the velocity points do not need to be updated during the course of the simulation.
- ◊ The program still requires the depth at velocity points to be set to MOR for morphological simulations. This anticipated that this restriction is lifted in a coming release.
- ◊ Since the MOR and MIN procedures for computing the depth at cell interfaces are equivalent, we advise you to use the MIN procedure during the calibration of a hydronamic model that will later on be converted into a morphological model.



11.8 Validation

To test and validate the formulations and implementation of the 3D sediment transport feature many simulations have been executed. These are reported in [Lesser et al. \(2000\)](#); [Lesser \(2003\)](#); [Ruessink and Roelvink \(2000\)](#); [Lesser et al. \(2004\)](#); [Roelvink \(2003\)](#).

Remarks:

- ◊ Small negative sediment concentrations ($-1 \cdot 10^{-3} \text{ kg/m}^3$) can be found in a computation. These negative concentrations can be suppressed by applying a horizontal Forester filter, Sections [4.5.8](#) and [10.6.4](#). However, this can result in a substantially larger computing time. It is suggested to accept small negative concentrations and to apply a Forester filter only when the negative concentrations become unacceptably large.
- ◊ A vertical Forester filter applied in a sediment transport computation will not affect the sediments. Since it smoothes the vertical profile and thus can have a strong influence on the vertical mixing processes, the vertical Forester filter is always de-activated for sediment.



DRAFT

12 Fixed layers in Z -model

12.1 Background

In coastal seas, estuaries and lakes, stratified flow occurs in combination with steep topography. 3D numerical modelling of the hydrodynamics and water quality in these areas requires accurate treatment of the vertical exchange processes. The existence of vertical stratification influences the turbulent exchange of heat, salinity and passive contaminants. The accuracy of the discretisation of the vertical exchange processes is determined by the vertical grid system. The vertical grid should:

- ◊ resolve the boundary layer near the bottom to allow an accurate evaluation of the bed stress
- ◊ be fine around pycnocline
- ◊ avoid large truncation errors in the approximation of strict horizontal gradients.

The commonly used σ co-ordinate system does not meet all the requirements. The σ co-ordinate system is boundary fitted but will not always have enough resolution around the pycnocline. The grid co-ordinate lines intersect the density interfaces. The σ co-ordinate gives significant errors in the approximation of strictly horizontal density gradients ([Leendertse, 1990](#); [Stelling and Van Kester, 1994](#)) in areas with steep bottom topography. Therefore, in 2003 a second vertical grid co-ordinate system based on Cartesian co-ordinates (Z -grid) was introduced in Delft3D-FLOW for 3D simulations of weakly forced stratified water systems, referred as Z -model in this manual.

The Cartesian Z co-ordinate system has horizontal co-ordinate lines, which are (nearly) parallel with density interfaces (isopycnals) in regions with steep bottom slopes. This is important to reduce artificial mixing of scalar properties such as salinity and temperature. The Z -model is not boundary-fitted in the vertical. The bottom (and free surface) is usually not a co-ordinate line and is represented as a staircase (zig-zag boundary). The number of grid cells in the vertical varies for each horizontal grid point. In the Z co-ordinate system, the vertical index of the free surface cell is “ $kmax$ ” and the vertical index of the bottom layer is “ $kmin$ ”. These indices are dependent on the horizontal index. The vertical layer index in the Z -model decreases from top to bottom. In the σ -model of Delft3D-FLOW the vertical index of the free surface cell is always “1” and the vertical index of the bottom layer is “ $kmax$ ”, which is independent of the horizontal index. In other words, the vertical index increases from top to bottom. The difference between the numbering in both grid systems has a historical background.

The staircase representation of the bottom, see [Figure 12.1](#) leads to inaccuracies in the approximation of the bed stress and the horizontal advection near the bed ([Bijvelds, 2001](#)). A transport flux along the bed is split into a horizontal and vertical part, which leads to numerical cross-wind diffusion in the transport equation for matter. The inaccuracies related to the staircase boundary representation of the bed in the Z -model are reduced by simple adjustments of the determination of the bed shear stress and the advection near solid vertical walls.

Grid spacing in the σ co-ordinate model is constructed by lines of constant σ . In a finite-difference model, due to the σ -transformation, the number of control volumes in the vertical direction is constant over the entire computational domain. The relative layer thickness $\Delta\sigma$ does not depend on the horizontal co-ordinates x and y . This makes it impossible to locally refine the grid around pycnocline in regions with steep bed topography. Moreover, the σ -transformation gives rise to, not always required, high grid resolution in shallow areas (tidal flats) and possibly insufficient grid resolution in deeper parts (holes) of the computational domain. At tidal flats at low tide, the mapping may even become singular. The numerical scheme may become non-convergent in these areas due to hydrostatic inconsistency ([Haney](#),

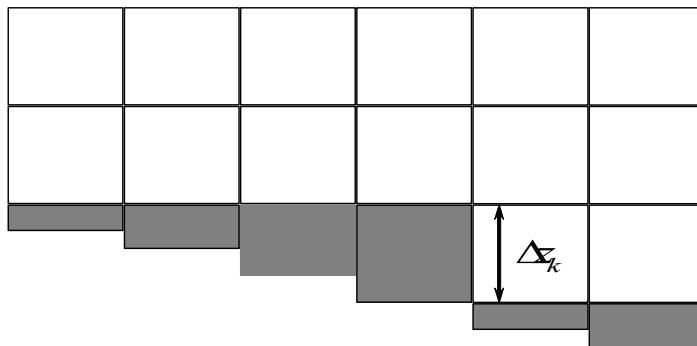


Figure 12.1: Irregular representation of bottom boundary layer in the Z -model

1991).

The vertical grid system of a Z co-ordinate model is based on horizontal surfaces with constant z co-ordinate value intersecting the water column; see Figure 12.1. The concept of layers used here should not be confused with layers of constant density in stratified flows. The layer thickness here is defined as the distance between two consecutive grid surfaces and is independent of space and time for an intermediate layer. The layer thickness of the top layer Δz_{kmax} is defined as the distance between the free surface and the first horizontal surface. The layer thickness may vary in space and time. The free surface moves through the vertical grid (Casulli and Cheng, 1992). The vertical index k of the top layer of neighbouring horizontal grid cells may vary. In that case, fluxes may be defined at cell faces that do not necessarily have a “wet” neighbouring grid cell. The thickness of the bottom layer is the distance between the bottom $z = -d(x, y)$ and the first horizontal surface above the bed. The layer thickness of the top and bottom cells can be very small, even approaching zero as the top cell becomes dry.

Let $z = z_k$ be strict horizontal surfaces, where k is an integer indicating the layer index. In the present model, the vertical grid spacing Δz_k is defined by:

$$\Delta z_k(x, y, t) = \min[\zeta(x, y, t), z_k] - \max[-d(x, y), z_{k-1}] \quad (12.1)$$

Taking into account variable grid sizes near the bed and allowing the free surface to move through the vertical grid introduces a lot of book keeping and makes the free surface boundary elaborate to treat in the numerical method. The grid points that are “wet” are determined every half time step. A computational cell is set “wet” when ever $\Delta z_{i,j,k} > 0$. Since the grid spacing near the bed and free surface may vary as a function of space and time, velocity points on the staggered grid of two adjacent grid cells may be situated at different vertical positions. Formally, this leads to additional terms in the discretized equations but these terms are not taken into account. The variation of the free surface and bed topography is smooth in most areas, which justifies the neglect of the cross terms involved.

The 3D shallow-water models in Delft3D-FLOW using σ co-ordinates and z co-ordinates respectively, are based on almost the same numerical methods.

The 3D shallow-water equations are discretized on a staggered grid (Arakawa C-grid). The shallow-water equations (SWE) are solved by an ADI-type of factorization for the barotropic pressure (Stelling, 1984). Both the horizontal components of the velocity vector, u and v , are computed once in a full time step Δt . The vertical grid space may vanish due to drying and flooding of shallow areas. The vertical viscosity terms are integrated fully implicitly in order to avoid an excessive small time step imposed by the relatively small vertical grid spaces. The

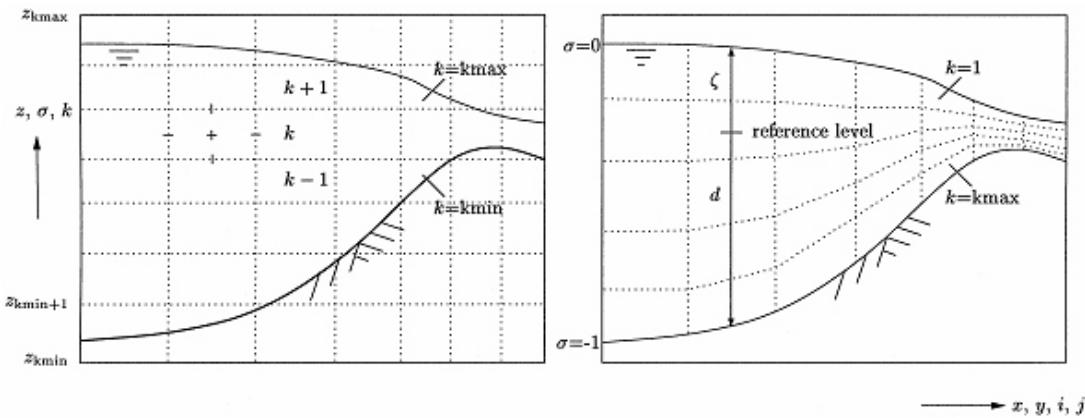


Figure 12.2: Vertical computational grid Z-model (left) and σ -model (right)

vertical advection terms can be integrated using either an implicit scheme (central differences), or an explicit upwind (finite volume scheme), depending on the application. The explicit scheme introduced a time step criterium for stability. The horizontal viscosity terms are integrated explicitly. For the integration of the horizontal advection terms, different schemes are available:

- ◊ Explicit Multi-Directional Upwind scheme (`MOMSOL = MDUE`, default option)
- ◊ Implicit Multi-Directional Upwind scheme (`MOMSOL = MDUI`)
- ◊ Implicit (first-order) Upwind scheme (`MOMSOL = IUPW`)
- ◊ Explicit Flooding scheme (`MOMSOL = FLOOD`)
- ◊ Explicit Upwind Finite-Volume scheme (`MOMSOL = FINVOL`)

The characteristics of these different schemes are explained in [subsection 12.5.1](#).

For the explicit integration schemes (or when flooding of dry cells is involved), the time step is restricted by the Courant-Friedrichs-Lowy condition for horizontal advection.

For the computation of the vertical eddy viscosity and eddy diffusivity several methods are available, just as for the σ -model. The standard $k-\varepsilon$ turbulence closure model ([Rodó, 1984](#)) is commonly recommended. It uses two partial differential equations to compute the transport of turbulent kinetic energy and energy dissipation rate. The production term only depends on the vertical gradients of the horizontal velocity. The presence of stratification is taken into account by the buoyancy flux.

A finite volume approach is used for the discretisation of the scalar transport equation, which ensures mass conservation. To circumvent time step restrictions imposed by the small vertical grid size in drying areas, implicit time integration is used for the vertical derivatives in the transport equation. In horizontal and vertical direction, diffusion is discretized using central differences. For horizontal derivatives, either explicit or implicit approximations can be used, depending on the application. The horizontal advective terms can be computed by either the Van Leer-2 TVD scheme or the Implicit Upwind scheme (first order accurate) and vertical advection is computed using an implicit central difference method to avoid time step limitations. Apart from the free surface, the vertical grid spacing is not a function of time.

An overview of all available schemes for horizontal and vertical advection and diffusion for both the momentum equation and the transport equation in the Z -layer model is given in [Table 12.1](#). For comparison also the options available in the σ -model have been included.

Table 12.1: Available advection and diffusion schemes in the Z -layer model (for comparison also the options available in the σ -model have been included).

Method	Process	Options	Remarks
σ -model			
	Time integration	ADI	section 10.4
	Hor. advection (momentum)	- CYCLIC - WAQUA - FLOOD	subsection 10.5.1
	Vert. advection (momentum)	Central implicit	subsection 10.5.2
	Hor. advection and diffusion (transport)	- Cyclic - Van Leer-2	section 10.6
Z -model			
	Time integration	ADI	section 12.2
	Hor. advection (momentum)	- MDUE - MDUI - IUPW - FLOOD - FINVOL	subsection 12.5.1
	Vert. advection (momentum)	- Central implicit - Upwind explicit (finite volume)	subsection 12.5.2
	Hor. advection (transport)	- Van Leer-2 - IUPW	section 12.6
	Hor. diffusion (transport)	- Central explicit - Central implicit	section 12.6

12.2 Time integration of the 3D shallow water equations

12.2.1 ADI time integration method

The 3D shallow-water equations are discretized on a staggered grid (Arakawa C-grid). The shallow-water equations (SWE) are solved by an ADI-type of factorization for the barotropic pressure ([Stelling, 1984](#)). Both the horizontal components of the velocity vector, u and v , are computed once in a full time step Δt . The vertical advection and viscosity term are integrated fully implicitly in order to avoid an excessive small time step imposed by the relatively small vertical grid spaces near the bottom and the free surface. The vertical grid space may vanish due to drying and flooding of the top layer. The horizontal advection and viscosity terms are integrated explicitly. The time step is restricted by the Courant-Friedrichs-Lowy condition for horizontal advection.

In vector form (for the 2D case), the ADI-method is given by:

Step 1:

$$\frac{\vec{U}^{\ell+\frac{1}{2}} - \vec{U}^\ell}{\frac{1}{2}\Delta t} + \frac{1}{2}A_x \vec{U}^{\ell+\frac{1}{2}} + \frac{1}{2}A_y \vec{U}^\ell + B \vec{U}^{\ell+\frac{1}{2}} = \vec{d}, \quad (12.2)$$

Step 2:

$$\frac{\vec{U}^{\ell+1} - \vec{U}^{\ell+\frac{1}{2}}}{\frac{1}{2}\Delta t} + \frac{1}{2}A_x \vec{U}^{\ell+\frac{1}{2}} + \frac{1}{2}A_y \vec{U}^{\ell+1} + B \vec{U}^{\ell+1} = \vec{d}, \quad (12.3)$$

with:

$$A_x = \begin{pmatrix} 0 & -f & g \frac{\partial}{\partial x} \\ 0 & 0 & 0 \\ H \frac{\partial}{\partial x} & 0 & u \frac{\partial}{\partial x} \end{pmatrix}, \quad (12.4)$$

$$A_y = \begin{pmatrix} 0 & 0 & 0 \\ f & 0 & g \frac{\partial}{\partial y} \\ 0 & H \frac{\partial}{\partial y} & v \frac{\partial}{\partial y} \end{pmatrix}, \quad (12.5)$$

and:

$$B = \begin{pmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad (12.6)$$

with λ the linearised bottom friction coefficient.

To improve stability the bottom friction is integrated implicitly for each stage. \vec{d} is the right-hand side containing the horizontal advection terms, external forcing like wind and atmospheric pressure. The time integration of the horizontal viscosity terms is discussed in [subsection 10.5.1](#) and is dependent on the formulation.

In the first stage, the time level proceeds from ℓ to $\ell + \frac{1}{2}$ and the simulation time from $t = \ell\Delta t$ to $t = (\ell + \frac{1}{2})\Delta t$. In this stage the U -momentum equation is solved, which is implicitly coupled with the continuity equation, [Equation 9.3](#), by the free surface gradient. In the second stage, the time level proceeds from $\ell + \frac{1}{2}$ to $\ell + 1$. The V -momentum equation is solved, which is implicitly coupled with the continuity equation by the free surface gradient.

For the 3D shallow water equations, the horizontal velocity components are coupled in the vertical direction by the vertical advection and viscosity term. In the vertical direction a fully implicit time integration method is applied, which is first-order accurate in time and leads to tri-diagonal systems of equations. The vertical coupling of the discretised momentum equations is eliminated by a double sweep algorithm.

12.2.2 Linearisation of the continuity equation

The non-linear terms in the coupled continuity equation and momentum equations, given by Eqs. 9.3, 9.7 and 9.8, are removed by linearisation of the fluxes in time. For the terms containing U velocity in the continuity equation, the linearisation leads to¹:

$$\begin{aligned} \frac{\partial}{\partial \xi} \left(\sqrt{G_{\eta\eta}} H U \right)^{\ell+\frac{1}{2}} &\approx \frac{\partial}{\partial \xi} \left(\sqrt{G_{\eta\eta}} H^\ell U^{\ell+\frac{1}{2}} \right) \approx \\ &\approx \frac{\left(\sqrt{G_{\eta\eta}} H^\ell U^{\ell+\frac{1}{2}} \right)_{m+\frac{1}{2},n} - \left(\sqrt{G_{\eta\eta}} H^\ell U^{\ell+\frac{1}{2}} \right)_{m-\frac{1}{2},n}}{\Delta \xi} \end{aligned} \quad (12.7)$$

12.3 Bed stress term

The bed stress term is computed using the logarithmic boundary layer relation expressed by [Equation 9.58](#). In this expression the grid distance of the first grid point above the bed, Δz_b is used to determine the bed stress. When the distribution of the layer thickness at the bottom shows large variation then large errors in the water level gradient may be introduced. This is caused by local maxima of the turbulent energy level computed by the turbulence closure model that affects the vertical viscosity term and vertical velocity.

In the Z -model this situation is more likely to occur than in the σ -model because the grid distance of the first grid point above the bed, Δz_b can vary strongly in the x -, y -space and its value can locally be quite small; see [Figure 12.1](#). This will result in a bed stress term that is inaccurate and discontinuous.

To reduce this effect, the bed stress term is computed using the velocity at one grid point above the bed (unless the number of active layers equals one):

$$u^* = \frac{u_{bottom+1} \kappa}{\ln \left(1 + \frac{\frac{\Delta z_{bottom+1}}{2} + \Delta z_{bottom}}{z_0} \right)} \quad (12.8)$$

An additional option is added to Delft3D-FLOW to improve the accuracy and smoothness of the computed bottom shear stress significantly. This option involves the local remapping of the near-bed layering to an equidistant layering, as described in [Platzek et al. \(2012\)](#), see [section B.26](#).

12.4 Horizontal viscosity terms

The horizontal viscosity term is simplified (see Eqs. 9.26 and 9.27). The simplification yields a Laplace operator along grid lines. The u -momentum equation involves only second-order derivatives of the u -velocity.

In the momentum equations, the complete Reynolds stress tensor, given by Eqs. 9.23 to 9.25, is used in the following cases:

- ◊ partial slip at closed boundaries,
- ◊ no slip at the closed boundaries,
- ◊ HLES-model for sub-grid viscosity.

¹This linearisation of the continuity equation is also used in the σ -model in combination with the Flooding scheme for advection

Table 12.2: Time step limitations shallow water solver Delft3D-FLOW

Points per wave period T	$\Delta t \leq \frac{T}{40}$
Accuracy ADI for barotropic mode for complex geometries	$C_f = 2\Delta t \sqrt{gH \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right)} < 4\sqrt{2}$
Explicit advection scheme	$\frac{\Delta t u }{\Delta x} < 2$
Stability baroclinic mode	$\Delta t \sqrt{\frac{\Delta \rho}{\rho} \frac{gH}{4} \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right)} < 1$
Explicit algorithm flooding	$\frac{\Delta t u }{\Delta x} < 2$
Stability horizontal viscosity term	$\Delta t 2\nu_H \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) < 1$

For the complete Reynolds stress tensor, the shear stress $\tau_{\xi\eta}$ in the u -momentum equation contains derivatives of the v -velocity. The horizontal viscosity term is integrated explicitly, which leads to the following additional stability condition:

$$\Delta t \leq \frac{1}{2\nu_H} \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right)^{-1}. \quad (12.9)$$

In case of a curvilinear grid $\Delta x = \sqrt{G_{\xi\xi}}$ and $\Delta y = \sqrt{G_{\eta\eta}}$.

12.4.1 Overview time step limitations

In Table 12.2 an overview of the time step limitations due to stability and accuracy for the time integration limitations are given for the shallow water code Delft3D-FLOW. Which of the limitations is most restrictive is dependent on the kind of application: length scale, velocity scale, with or without density-coupling, etc. In the relations below Δx and Δy are horizontal grid sizes. For a curvilinear grid $\Delta x = \sqrt{G_{\xi\xi}}$ and $\Delta y = \sqrt{G_{\eta\eta}}$.

12.5 Spatial discretisations of 3D shallow water equations

12.5.1 Horizontal advection terms

The horizontal advective terms in the Z-model can be approximated by a number of different schemes:

- ◊ Explicit Multi-Directional Upwind scheme (`MOMSOL = MDUE`, default option)
- ◊ Implicit Multi-Directional Upwind scheme (`MOMSOL = MDUI`)
- ◊ Implicit (first-order) Upwind scheme (`MOMSOL = IUPW`)
- ◊ Explicit Flooding scheme (`MOMSOL = FLOOD`)
- ◊ Explicit Upwind Finite-Volume scheme (`MOMSOL = FINVOL`)

The Multi-Directional Upwind `MDUE` (explicit) and `MDUI` (Implicit) discretisations approximate the advection terms along streamlines (Van Eijkeren et al., 1993). Although this method is formally of first order accuracy, it was found to be much less diffusive than the standard first order upwind method (Bijvelds, 1997). The discretisation stencil is dependent on the direction of the flow; see Figure 12.3. The discretisation for flow with a positive U and V -component is

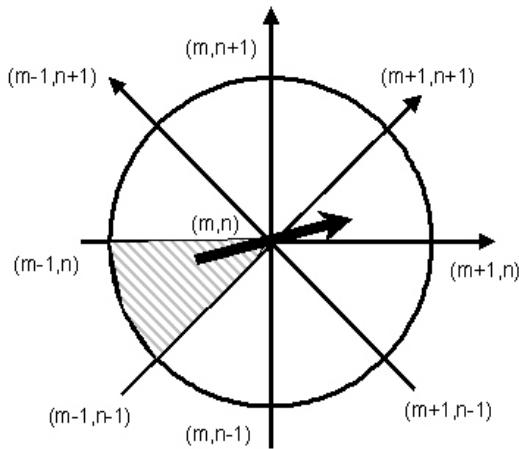


Figure 12.3: discretisation along streamlines. Grid points in difference stencil dependent on flow direction

given by the following two equations.

$$\frac{u}{\sqrt{G_{\xi\xi}}}\frac{\partial u}{\partial \xi}\Big|_{m,n,k} = \begin{cases} \frac{u_{m,n,k}}{(\sqrt{G_{\eta\eta}})_{m,n}} \left(\frac{u_{m,n-1,k} - u_{m-1,n-1,k}}{\Delta\xi} \right) & \bar{v}_{m,n,k}^{\xi\eta} > u_{m,n,k} > 0 \\ \frac{u_{m,n,k}}{(\sqrt{G_{\eta\eta}})_{m,n}} \left(\frac{u_{m,n,k} - u_{m-1,n,k}}{\Delta\xi} \right) & u_{m,n,k} > \bar{v}_{m,n,k}^{\xi\eta} > 0 \end{cases}, \quad (12.10)$$

and:

$$\frac{v}{\sqrt{G_{\eta\eta}}}\frac{\partial u}{\partial \eta}\Big|_{m,n,k} = \begin{cases} \frac{\bar{v}_{m,n,k}^{\xi\eta}}{(\sqrt{G_{\eta\eta}})_{m,n}} \left(\frac{u_{m,n,k} - u_{m,n-1,k}}{\Delta\eta} \right) & \bar{v}_{m,n,k}^{\xi\eta} > u_{m,n,k} > 0 \\ \frac{\bar{v}_{m,n,k}^{\xi\eta}}{(\sqrt{G_{\eta\eta}})_{m,n}} \left(\frac{u_{m-1,n,k} - u_{m-1,n-1,k}}{\Delta\eta} \right) & u_{m,n,k} > \bar{v}_{m,n,k}^{\xi\eta} > 0 \end{cases}, \quad (12.11)$$

The implicit first order upwind *IUPW* method can be employed when stability is most important and accuracy is of less interest. This method provides most damping or numerical diffusion of the available options.

The flooding solver *FLOOD* is the same as for the σ -model (see subsection 10.5.1). It switches between conservation of momentum and conservation of energy based on local flow expansions and contractions due to bottom gradients.

The finite-volume *FINVOL* scheme is momentum-conservative and can also be applied in flooding simulations where conservation of momentum is required.

The *FLOOD* and *FINVOL* scheme are both fully explicit and thus require the time step to fulfill the Courant-Friedrichs-Lowy stability condition.

Near the boundaries, the discretisation stencils for the advection terms may contain grid points on or across the boundary. To avoid an artificial boundary layer or instabilities, the discretisations are reduced to smaller stencils.

12.5.2 Vertical advection term

The horizontal velocities of adjacent vertical layers are coupled by the vertical advection and the vertical viscosity term. The *Z*-model can have a very thin layer near the bottom or near the free surface. To prevent instabilities, we recommend the use of the fully implicit time integration for the vertical exchange terms (default). This results in tridiagonal systems of equations in the vertical.

In a shallow water model the horizontal length scale is much larger than the vertical length scale. In the vertical direction the eddy viscosity term dominates the advection term, except in stratified flows where the turbulent exchange is reduced and advection may be dominant. For the space discretisation of the vertical advection term, a second order central difference is used:

$$w \frac{\partial u}{\partial z} \Big|_{m,n,k} = \bar{w}_{m,n,k}^{\xi z} \left[\frac{u_{m,n,k+1} - u_{m,n,k-1}}{\frac{1}{2} h_{m,n,k-1} + h_{m,n,k} + \frac{1}{2} h_{m,n,k+1}} \right], \quad (12.12)$$

where $h_{m,n,k}$ denotes the thickness of layer with index k defined by $h_{m,n,k} = z_k - z_{k-1}$.

An explicit, momentum-conservative, upwind finite volume scheme is also available if momentum conservation is considered to be more crucial than stability for a certain application. At present, this scheme for vertical advection can only be used in combination with the finite volume *FINVOL* scheme for horizontal advection. The vertical advection scheme is automatically switched to this explicit upwind approach when the horizontal advection scheme is set to `MOMSOL = FINVOL`. The user should note that this scheme is fully explicit and thus requires the time step to fulfil the Courant-Friedrichs-Lowy stability condition, also for vertical advection, i.e. depending on the vertical grid spacing Δz and the vertical velocity w .

12.5.3 Viscosity terms

The approximation of the viscosity terms are based on central differences. The vertical viscosity term is discretised as:

$$\begin{aligned} \frac{\partial}{\partial z} \left(\nu_V \frac{\partial u}{\partial z} \right) \Big|_{m,n,k} &= \frac{\nu_V|_{m,n,k+1}}{h_{m,n,k}} \left(\frac{u_{m,n,k+1} - u_{m,n,k}}{\frac{1}{2} (h_{m,n,k+1} + h_{m,n,k})} \right) + \\ &\quad - \frac{\nu_V|_{m,n,k}}{h_{m,n,k}} \left(\frac{u_{m,n,k} - u_{m,n,k-1}}{\frac{1}{2} (h_{m,n,k} + h_{m,n,k-1})} \right). \end{aligned} \quad (12.13)$$

The vertical eddy viscosity is computed at the layer interface.

We note that near the bottom, the grid layering may be highly non-uniform due to the fact that the bottom introduces thin layers. In such situations, the approximation given in [Equation 12.13](#), is very inaccurate due to the linear approximations of the (often) near-logarithmic velocity profiles near the bottom, see e.g. [Platzek et al. \(2012\)](#). To avoid these problems, a near-bed, layer-remapping approach was implemented as proposed by [Platzek et al. \(2012\)](#). This option can be switched on using an additional keyword, described in [section B.26](#).

12.6 Solution method for the transport equation

A robust and accurate solver for scalar transport has to satisfy the following demands:

- ◊ mass conservation by consistency with the discrete continuity equation,

- ◊ monotonicity (positive solution),
- ◊ accuracy,
- ◊ suitable for both time-dependent and steady state problems and
- ◊ computationally efficient.

To ensure that the total mass is conserved, the transport equation is discretised with a mass conserving Finite Volume approach (flux form). The transport equation formulated in a conservative form in Cartesian co-ordinates is given by:

$$\begin{aligned} \frac{\partial(\Delta z_{m,n,k} c)}{\partial t} + \frac{1}{\Delta x \Delta y} \{ \partial_x (uc \Delta y \Delta z_{m,n,k}) + \partial_y (vc \Delta x \Delta z_{m,n,k}) \} \\ + (wc)_{m,n,k} - (wc)_{m,n,k-1} = \\ + \frac{1}{\Delta x \Delta y} \left\{ \partial_x \left[D_H \Delta y \Delta z_{m,n,k} \frac{\partial c}{\partial x} \right] + \partial_y \left[D_H \Delta x \Delta z_{m,n,k} \frac{\partial c}{\partial y} \right] \right\} + \\ + \left[D_V \frac{\partial c}{\partial z} \right]_{m,n,k} - \left[D_V \frac{\partial c}{\partial z} \right]_{m,n,k-1} - \lambda_d \Delta z_{m,n,k} c + S, \quad (12.14) \end{aligned}$$

with λ_d representing the first order decay process and S the source and sink terms per unit area. Δx and Δy are the grid spaces in the physical space. The index of the computational layer k increases in the positive Z -direction. k_{min} and k_{max} are the indices of the bottom and top cells respectively. The range of vertical index varies over the horizontal grid. The vertical grid spacing of the top layer $\Delta z_{k_{max}}$ changes in time as the free surface moves. There is no scalar flux through the free surface and the bottom.

Two methods have been implemented in the Delft3D-FLOW transport solver in Z -model to approximate the horizontal advective fluxes. The default option is the so-called *Van Leer-2* scheme ([Van Leer, 1974](#)), which guarantees monotonicity of the solution. The time integration of the *Van Leer-2* scheme is explicit and therefore the CFL condition for advection gives a stability condition:

$$C_{adv} = \max \left(\frac{u \Delta t}{\Delta x}, \frac{v \Delta t}{\Delta y} \right) \leq 1, \quad (12.15)$$

The second available method is an implicit first-order upwind scheme *IUPW*, which does not have the CFL stability condition, but which is less accurate than the *Van Leer-2* scheme.

The horizontal diffusive terms are discretized conform the choice made for the horizontal advective terms, i.e. when the *Van Leer-2* scheme is chosen, the diffusive terms are approximated explicitly, whereas when the *IUPW* scheme is chosen, they are modelled implicitly. In the case the explicit integration of the horizontal diffusive fluxes is used (i.e. for the *Van Leer-2* scheme) an upper limit for the time step is given by:

$$\Delta t \leq \frac{1}{2D_H} \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right)^{-1}. \quad (12.16)$$

The scalar concentrations are coupled in the vertical direction by the vertical advection and diffusion term. The vertical transport is computed at the layer interfaces which are situated entirely under the free surface layer both at the old and the new time level. An explicit time integration of the vertical exchange terms near the bottom and free surface would lead to very

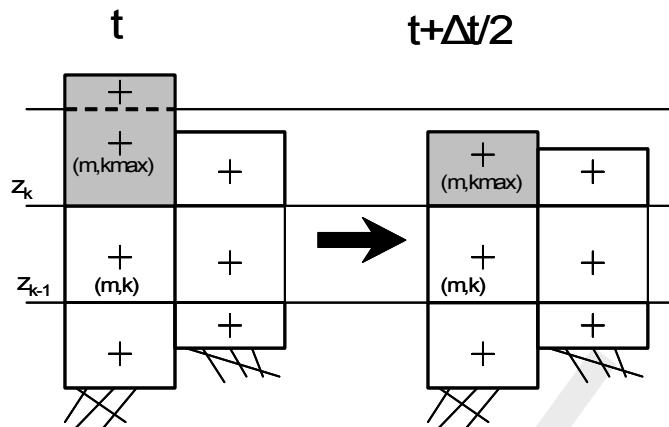


Figure 12.4: Aggregation of Control volumes in the vertical due to variation position free surface

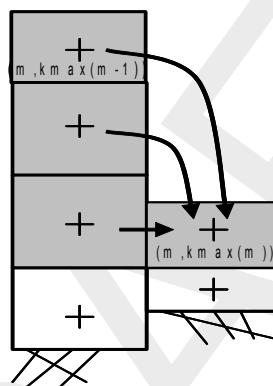


Figure 12.5: Horizontal fluxes between neighbouring cells with variation in position free surface

severe time step limitations:

$$\Delta t \leq \frac{(\Delta z)^2}{2D_V}, \quad (12.17)$$

$$\Delta t \leq \frac{\Delta z}{w}. \quad (12.18)$$

Therefore in the vertical direction a fully implicit time integration method is applied, which is first order in time and leads to tri-diagonal systems of equations. The vertical coupling of the discretised transport equations is removed by a double sweep algorithm.

To combine the horizontal and vertical transport, we introduce an approach based on a fractional step method. The water column is divided into two parts; see [Figure 12.4](#). In this case, the free surface at the old time level t and the new time level $t + \frac{1}{2}\Delta t$ are in a different computational layer. The 3D part of the water column consists of the cells (finite volumes) which are situated entirely under the free surface layer at both time levels. The remaining “wet” cells are aggregated to one Control Volume and the horizontal fluxes are summed, see [Figure 12.5](#). The concentrations at the new time level, $t + \frac{1}{2}\Delta t$ are computed. If the index of free surface the cell has increased, the concentration at the new time level is assumed to be constant over all the computational layers in the free surface cell.

Source terms are integrated explicitly. In order to avoid negative concentrations and instabilities, sink terms are integrated fully implicitly.

12.6.1 Horizontal advection

Two methods have been implemented in the Delft3D-FLOW transport solver to approximate the horizontal advective fluxes. The default option is the so-called *Van Leer-2* scheme ([Van Leer, 1974](#)), the second option is an implicit first-order upwind scheme *IUPW*.

12.6.1.1 Van Leer-2 scheme

The *Van Leer-2* scheme ([Van Leer, 1974](#)) is used for the approximation of the horizontal transport terms. It combines two numerical schemes, namely a first order upwind scheme and the second order upwind scheme developed by Fromm. In case of a local minimum or maximum the first order upwind scheme is applied, whereas the upwind scheme of Fromm is used in case of a smooth numerical solution.

The interpolation formula for the horizontal fluxes is given by:

$$F_{m,n,k} = u_{m,n,k} h_{m,n,k} \Delta y \begin{cases} c_{m,n,k} + \alpha (1 - CFL_{adv-u}) (c_{m,n,k} - c_{m-1,n,k}) \frac{c_{m+1,n,k} - c_{m,n,k}}{c_{m+1,n,k} - c_{m-1,n,k}}, \\ \text{when } u_{m,n,k} \geq 0, \\ c_{m+1,n,k} + \alpha (1 + CFL_{adv-u}) (c_{m,n,k} - c_{m-1,n,k}) \frac{c_{m+1,n,k} - c_{m+2,n,k}}{c_{m,n,k} - c_{m+2,n,k}}, \\ \text{when } u_{m,n,k} < 0, \end{cases} \quad (12.19)$$

with:

$$CFL_{adv-u} = \frac{\Delta t |u|}{\Delta x} \quad (12.20)$$

and:

$$\alpha = \begin{cases} 0, & \left| \frac{c_{m+1,n,k} - 2c_{m,n,k} + c_{m-1,n,k}}{c_{m+1,n,k} - c_{m-1,n,k}} \right| > 1, \text{ (local max. or min.)}, \\ 1, & \left| \frac{c_{m+1,n,k} - 2c_{m,n,k} + c_{m-1,n,k}}{c_{m+1,n,k} - c_{m-1,n,k}} \right| \leq 1, \text{ (monotone)}. \end{cases} \quad (12.21)$$

In y -direction, a similar discretisation is applied. The time integration of the *Van Leer-2* scheme is explicit. The Courant number for advection should be smaller than 1.

12.6.1.2 Implicit upwind scheme

The interpolation formula for the horizontal fluxes in x -direction is given by:

$$F_{m,n,k}^{\ell+1} = u_{m,n,k}^\ell h_{m,n,k}^\ell \Delta y \begin{cases} c_{m,n,k}^{\ell+1}, \\ \text{when } u_{m,n,k}^\ell \geq 0, \\ c_{m+1,n,k}^{\ell+1}, \\ \text{when } u_{m,n,k}^\ell < 0, \end{cases} \quad (12.22)$$

It is a first order upwind scheme. Note that we now added the time level superscript ℓ to indicate the implicit treatment of the concentrations c in the fluxes. In y -direction the fluxes are discretized similarly.

12.6.2 Vertical advection

In the vertical direction, the fluxes are discretised with a central scheme:

$$(wc)_{m,n,k} - (wc)_{m,n,k-1} = w_{m,n,k} \left(\frac{c_{m,n,k} + c_{m,n,k+1}}{2} \right) - w_{m,n,k-1} \left(\frac{c_{m,n,k} + c_{m,n,k-1}}{2} \right). \quad (12.23)$$

The time integration in the vertical direction is fully implicit. The vertical advection leads to a tri-diagonal system in the vertical. If the flow in the vertical is advection dominated due to vertical stratification in combination with up welling or down welling near a closed boundary, a sill or a discharge of buoyant water, the central differences in the vertical may give rise to non-physical spurious oscillations. The scalar concentration then computed has an unphysical maximum or minimum (overshoot or undershoot).

12.6.3 Forester filter

In 3D, the central differences in the vertical may give rise to non-physical spurious oscillations, so-called “wiggles” (Gresho and Lee, 1981) in the solution. These wiggles arise in the vicinity of steep gradients of the quantity to be resolved. The wiggles in the concentration may be introduced in stratified areas near closed boundaries and steep bottom slopes. Positive solutions are not guaranteed, because there the vertical transport is large. In case of negative concentrations, an iterative filter procedure based on local diffusion along Z -lines followed by a vertical filter is started in order to remove the negative values. These filters can be switched on by the user (see subsection 4.5.8). The filtering technique in this procedure is the so-called Forester filter (Forester, 1979), a non-linear approach which removes the computational noise without inflicting significant amplitude losses in sharply peaked solutions.

If concentration $c_{m,n,k}$ is negative, then the iterative filtering process in the x -direction is given by:

$$c_{m,n,k}^{p+1} = c_{m,n,k}^p + \frac{c_{m+1,n,k}^p - 2c_{m,n,k}^p + c_{m-1,n,k}^p}{4}, \quad (12.24)$$

This filter is applied only in grid cells where a negative concentration occurs.

The superscript p denotes the iteration number. The filter smooths the solution and reduces the local minima (negative concentrations). Consequently, a positive concentration will remain positive, i.e. it will not introduce negative concentrations irrespective the steepness of the concentration gradients. A negative concentration surrounded by positive concentrations, usually the result of ill-represented steep gradients (wiggles), will be less negative after one iteration and is effectively removed after several iterations by adding enough (local) diffusion to force the concentration to become positive. Maximally 100 iterations are carried out. If there is still a grid cell with a negative concentration after 100 iterations, then a warning is generated.

Local maxima and minima in temperature or salinity in the vertical direction, generated by the computational method may give physically unstable density profiles and can also better be removed by a numerical filter than by turbulent vertical mixing. A similar filtering technique as in the horizontal direction is applied for points with a local maximum or minimum in the vertical:

local maximum: $c_{m,n,k} > \max(c_{m,n,k+1}, c_{m,n,k-1}) + \varepsilon$,

local minimum: $c_{m,n,k} < \min(c_{m,n,k+1}, c_{m,n,k-1}) + \varepsilon$,

the filter is applied, with $\varepsilon = 10^{-3}$. The numerical diffusion coefficient of the vertical filter is:

$$D_{num} = \frac{\Delta z^2}{2\Delta t}. \quad (12.25)$$

Smooth but unstable vertical density profiles of salinity and temperature in the vertical direction, can sometimes also better be vertically mixed by a numerical filter than by the turbulence model. For salinity, the algorithm is given by:

If $s_{m,n,k} > s_{m,n,k-1} + \varepsilon$ Then

$$\begin{aligned} s_{m,n,k} &= s_{m,n,k} - \min(\Delta z_k, \Delta z_{k-1}) \frac{(s_{m,n,k} - s_{m,n,k-1})}{2\Delta z_k} \\ s_{m,n,k-1} &= s_{m,n,k-1} + \min(\Delta z_k, \Delta z_{k-1}) \frac{(s_{m,n,k} - s_{m,n,k-1})}{2\Delta z_{k-1}} \end{aligned} \quad (12.26)$$

Endif

with $\varepsilon = 10^{-3}$.

If both the horizontal and vertical filters are switched on, then first the filter in the horizontal direction is carried out. It is followed by the filter in the vertical direction, thereby minimising the additional vertical mixing.



Remark:

- ◊ The vertical Forester filter does not affect other constituents. When activated it only smooths salinity and temperature.

12.7 Baroclinic pressure term

The transport equation is coupled with the momentum equations by the baroclinic pressure term; see Eqs. 9.15 and 9.16 and subsection 9.3.4. The baroclinic pressure term reads:

$$P_x(x, z) = \frac{1}{\rho_0} \int_z^\zeta g \frac{\partial \rho(x, z)}{\partial x} dz. \quad (12.27)$$

In the Z -model, the horizontal derivatives of the density can be discretised straightforwardly on the computational grid:

$$P_x|_{m,n,k} = \frac{g}{\rho_0} \left\{ \frac{1}{2} \Delta z_{m,n,k} \frac{\rho_{m+1,n,k} - \rho_{m,n,k}}{\Delta x} + \sum_{j=k+1}^{kmax} \left[\Delta z_{m,n,j} \frac{\rho_{m+1,n,j} - \rho_{m,n,j}}{\Delta x} \right] \right\} \quad (12.28)$$

The temporal variations in salinity and temperature are slow compared to the variations in the flow and therefore the baroclinic term in the momentum equations is treated explicitly, introducing a stability condition for internal gravity waves (baroclinic mode), see Table 12.2. The coupling with the flow is weak and in Delft3D-FLOW, the transport equation is solved independently of the flow for each half time step.

12.8 Numerical implementation of the turbulence models

The turbulence closure models in Delft3D-FLOW are all based on the eddy viscosity concept; see [section 9.5](#). The eddy viscosity is always based on information of the previous half time step. The transport equations of turbulent kinetic energy k , [Equation 9.127](#), and dissipation rate ε , [Equation 9.128](#) are solved in a *non-conservative* form. For turbulent boundary flows local production, dissipation, and vertical diffusion are the dominant processes. On the staggered grid, the turbulent quantities k , ε and the eddy viscosity ν_V are positioned at the layer interfaces in the centre of the computational cell. This choice makes it possible to discretise the vertical gradients in the production term and buoyancy term accurately and to implement the vertical boundary conditions at the bed and the free surface. First order upwind differencing for the advection provides positive solutions. For more details we refer to [Uittenbogaard et al. \(1992\)](#) and [Bijvelds \(2001\)](#).

12.9 Drying and flooding

Just as in the σ -model, in the Z -model shallow parts of estuaries and coastal seas are subject to drying and flooding during the tidal cycle. The drying and flooding procedure in the Z -model is almost identical to the procedure applied in the σ -model. Due to different representation of the bottom depth (see following section) the procedure is somewhat simplified.

Furthermore, a technical difference is introduced due to the nature of the grid definition in the vertical. In a σ -model, when a point is set to dry, then all the layers are deactivated simultaneously. In a Z -model however, the free surface can move freely through the vertical grid, resulting in the fact that . The top layer can vanish due to 'wetting and drying' in the vertical.

The crucial issues in a wetting and drying algorithm are:

- ◊ The way in which the bottom depth is defined at a water level point.
- ◊ The way in which the water level is defined at velocity points.
- ◊ The way in which the total water depth is defined.
- ◊ Criteria for setting a velocity and/or water level point wet or dry.

These items will be discussed below.

12.9.1 Bottom depth at water level points

The main difference between the σ -model and the Z -model is the representation of the bottom in the model. In the Z -model it is represented as a staircase around the depth in the water level points; see [Figure 12.1](#). However, due to grid staggering, the bottom depth in a water level point $d_{m,n}^\zeta$ is not uniquely defined; see [Figure 12.6](#).

The manner in which this depth value can be determined from the four surrounding depth points may be influenced by the user. In both the σ - and the Z -model, four options are available: MEAN, MAX, DP, and MIN. The option is specified through the value of the parameter DP SOFT. For the definition and a detailed discussion on this topic we refer you to Chapter 10.

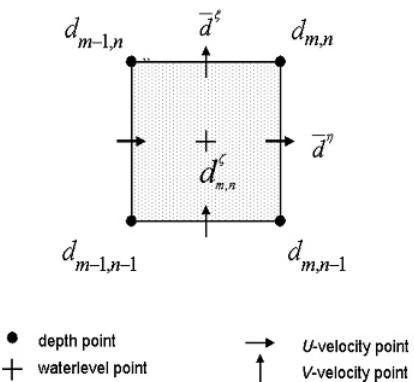


Figure 12.6: Definition bottom depth on Delft3D-FLOW grid

Drying and flooding switch

The drying and flooding switch, `DRYFLP`, works exactly the same as in the σ -model; see subsection 10.8.3. To activate the additional drying and flooding procedure at a water level point, based on the evaluation of the value of the total water depth, the value of `DRYFLP` must be set to YES. `DRYFLP = NO` implies that the drying and flooding check is only to be based on the individual test of the depth values at the cell interfaces.

If the total water depth in a water level point is negative:

$$H_{m,n}^{\zeta} = d_{m,n}^{\zeta} + \zeta_{m,n} \leq 0, \quad (12.29)$$

the horizontal cell is taken out of the computation and the half time step is repeated. The initial water level at a dry cell is determined by the depth at a water level point:

$$\zeta_{m,n} = -d_{m,n}^{\zeta}. \quad (12.30)$$

The surface layer thickness

The vertical grid size near the free surface depends on the spatial location and on time. Once the new free surface location has been computed, the following equation is used to determine the vertical grid size:

$$\Delta z_k(x, y, t) = \min [\zeta(x, y, t), z_k] - \max [-d(x, y), z_{k-1}], \quad (12.31)$$

except for the situation where the water level exceeds the maximum grid layer interface Z_{top} . In that case the vertical grid size of the top layer is increased to include the free surface:

$$\Delta z_k(x, y, t) = \zeta(x, y, t) - \max [-d(x, y), z_{k-1}]. \quad (12.32)$$

12.9.2 Bottom depth at velocity points

Due to the staggered grid applied in Delft3D-FLOW, see Figure 12.6, the bottom and total water depth at velocity points are not uniquely defined. For the Z -model the bottom is represented as a staircase (`DPUOPT=MIN`) of tiles, centred around the water level points, see Figure 12.7. The bottom depth at a velocity point is thus the minimum depth of the two surrounding bottom depths in the ζ -points:

$$\text{DPUOPT} = \text{MIN} : \bar{d}^{\eta} = \min \left(d_{m,n}^{\zeta}, d_{m+1,n}^{\zeta} \right), \quad (12.33)$$

In contrast to the σ -model, this value for `DPUOPT` is fixed. So any other value will not be accepted by the program.

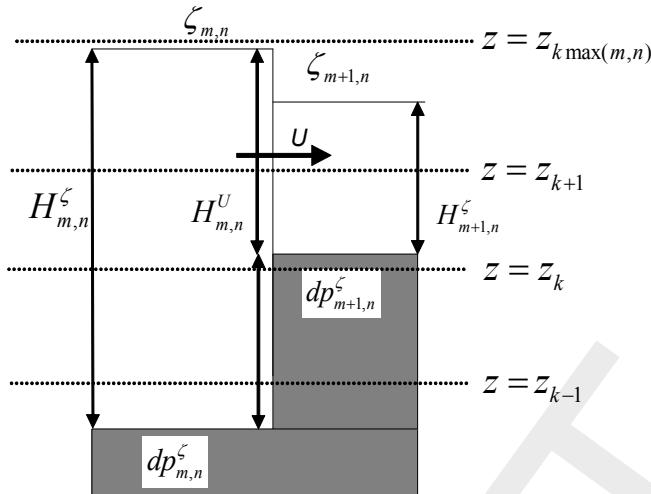


Figure 12.7: The flow-through height is determined by the flow direction. The bottom is represented as a staircase around the depth in water level points.

12.9.3 Upwinding of the water level in defining the total water depth

The total water depth in a U -velocity point is computed using the upwind water level:

$$H_{m,n}^U = \begin{cases} \bar{d}^\eta + \zeta_{m,n}, & U_{m,n} > 0, \\ \bar{d}^\eta + \zeta_{m+1,n}, & U_{m,n} < 0, \\ \bar{d}^\eta + \max(\zeta_{m,n}, \zeta_{m+1,n}), & U_{m,n} = 0, \end{cases} \quad (12.34)$$

with $U_{m,n}$ representing the depth averaged velocity both for 2D and 3D. The computation of the upwind total water depth $H_{m,n}^V$ in a V -velocity point is similar. The upwind approach is *physically more realistic* for velocity points between cells with different bottom depth at low water and falling tide (Figure 12.12) or for weir like situations.

Upwinding the water level in the determination of the total water depth at the velocity points as specified above enhances the discharge. The computed water level is generally higher than the average water level, resulting in a larger flow area, which allows the water level gradient to drive a larger amount of water into the neighbouring cell during the next time step. Taking the maximum of the two surrounding water levels at a dry cell face prevents that a velocity point is artificially kept dry.

The method above is *physically less realistic* if the flow has the opposite direction as the water level gradient (wind driven flow). This may result into flip-flop behaviour of the computational cell where it is alternately set to dry and wet during the computation.

12.9.4 Drying and flooding criteria

As described in section 10.4 an Alternating Direction Implicit (ADI) time integration method is used in Delft3D-FLOW. This method consists of two stages (half-time steps). At both stages the same drying and flooding algorithm is applied. Therefore, we will only describe the drying and flooding algorithm for the first half time step. If a new water level is computed, both the horizontal geometry (wet versus dry) and the vertical geometry (number of vertical layers) are updated.

The total water depth $H_{m,n}^U$ at a velocity point should at least be positive to guarantee a realistic discharge across a cell face. If the total water level drops below half of a user-specified threshold, then the velocity point is set dry. In 3D simulations, the velocity of a computational

layer is set to zero if the vertical grid space Δz_k^U vanishes. The computational cell is closed for the side normal to that velocity point. If the water level rises and the total water depth is larger than the threshold, the velocity point is set wet again. The drying threshold is given half the value of the wetting threshold (*hysteresis*) to inhibit changes of state in two consecutive time steps (“flip-flop”), due to oscillations introduced by the algorithm itself. If the vertical grid space Δz_k^U is larger than a threshold Δz_{min} the computational layer is taken into account. The initial velocity of the layer is the same as the velocity of the top layer of the previous half time level.

In Delft3D-FLOW it is also possible to initiate the bottom friction term in velocity points that have just become active in case of flooding. This makes sure that the flow through such a cell interface, directly experiences some bottom friction, which would not be the case if the cell face would be initialised with zero velocity. This algorithm is automatically switched on when using the bottom depth option DPSOPT=DP. Otherwise, it can be switched on using the additional parameter (see [Appendix B](#)):

SlpLim=#Y#,

(a slope limiter), which also makes sure that the flow along steep bottom slopes occurs with a limited water level gradient (see [section B.27](#)).

In 2D, the total water depth $H_{m,n}^\zeta$ at a water level point should at least be positive to guarantee a positive control volume. If the total water depth becomes negative, the four velocity points at the cell sides are set dry. In 3D simulations, the vertical grid space Δz_k^ζ should be positive.

The thickness of the water layer of a dry cell (retention volume) is dependent on the threshold d specified by you. Therefore, the threshold value d must fulfil the following condition:

$$\delta \geq \frac{\partial \zeta}{\partial t} \frac{\Delta t}{2}. \quad (12.35)$$

In general, the magnitude of the disturbances generated by the drying and flooding algorithm will depend on the grid size, the bottom topography and the time step. The disturbances are small if the grid size is small and the bottom has smooth gradients. If the bottom has steep gradients across a large area on a tidal flat, a large area may be taken out of the flow domain in just one half integration time step. This will produce short oscillations. You can avoid this by smoothing the bottom gradients. Flooding is an explicit process. The boundary of the wet area can only move one grid cell per time step. If the time step is too large an unphysical water level gradient at the wet-dry interface is built up, which will generate a shock wave after flooding.

In the first stage of the ADI-method the drying and flooding algorithm in Delft3D-FLOW consists of the following checks:

- 1 Drying check for velocity points in x -direction ($H_{m,n}^U < 0.5\delta$ and $\Delta z_{m,n,k}^U < z_{min}$) and flooding check for velocity points in x -direction ($H_{m,n}^U > \delta$ and $\Delta z_{m,n,k}^U > z_{min}$). These checks are based on the water level of the previous half time step.
- 2 Drying check for velocity points in x -direction ($H_{m,n}^U < 0.5\delta$) during iterative solution for new water level.
- 3 Drying check (negative volumes) for water level points ($H_{m,n}^\zeta < 0.0$).

In the second stage of the ADI-method, the directions are interchanged.

The threshold δ is specified at input. The total water depth at velocity points is computed by the upwind approach.

When a computational cell is dry, the thickness of the water layer is below the drying threshold. If the computational cell is flooded, the water layer may be very thin and may cause problems in combination with online salt transport or off-line water quality simulations. In Delft3D-FLOW the computational part is protected against “dividing by zero” by assuming that the total water depth is at least 10% of the drying and flooding threshold $Dryflc$, which is also the minimum layer thickness Δz_{min} .

You may define in velocity points so-called weirs or spillways. Weirs are hydraulic structures causing energy losses, see [section 10.9](#). For a *2D weir* the height of the crest, $HKRU$, is taken into account in the drying and flooding algorithm.

The drying check for a *2D weir* point at a U -point is given by:

$$H_{m,n}^U < \frac{1}{2}\delta \text{ or } \max(\zeta_{m-1,n}, \zeta_{m,n}) + HKRU_{m,n} < \frac{1}{2}\delta, \quad (12.36)$$

and the flooding check:

$$H_{m,n}^U > \delta \text{ or } \max(\zeta_{m-1,n}, \zeta_{m,n}) + HKRU_{m,n} > \delta. \quad (12.37)$$

The weir acts as a thin dam for water levels lower than the crest height.

12.10 Cut-cell and 45 degrees closed boundaries

This feature applies for the Z -model only.

12.10.1 Introduction

A staircase closed boundary is sometimes unavoidable when schematising the land-water interface, even in curvilinear co-ordinates. To avoid inaccuracies introduced by the staircase closed boundaries in the Z -model, two methods for removal of such inaccuracies are implemented in Delft3D-FLOW. One is derived from the so-called Cut Cell method, for general curved closed boundaries which do not coincide with a gridline. The second one is derived for 45 degrees boundaries (1 to 1).

12.10.2 Cut Cells

The Cut Cell approach involves truncating the Control Volumes at the boundary surface to create new cells which conform to the shape of the boundary, see [Figure 12.8](#) (left). In the grid generator corner points are shifted, [Figure 12.8](#) (right) to remove the staircase. This approach is called a “Cut Cell” method; see [Kirkpatrick et al. \(2003\)](#).

We discuss the approximation of the fluxes through the boundary cells and the pressure gradients. It is necessary to relocate the velocity nodes associated with the cut boundary cells. The velocity nodes are placed at the centre of the cut face of the Control Volume. It allows the mass conservation to be discretised in the same manner as for a standard cell. The only difference is that the horizontal area (volume) is recalculated for the truncated cells. In 3D the horizontal area is the same for all layers. The pressure (water level) points are left in the original position, even though this may mean that they are physically outside the boundaries of the associated Control Volume. In the present implementation, the advection terms and the wall shear stresses are not corrected for the cut cells.

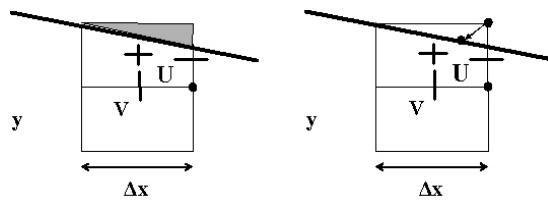


Figure 12.8: left: Cut Cell (definition) and right: defined by shifting (exaggerated) the corner point to boundary.

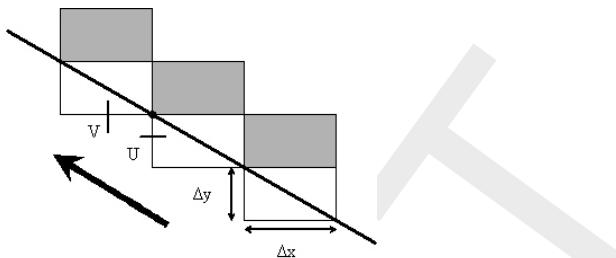


Figure 12.9: Flow along staircase boundary.

The spatial approximation of the advection terms was not adapted. Only for Cut Cells the curvature term was removed because the grid cells are no longer orthogonal.

On the staggered grid you need averaging to determine the V -velocity in a U -velocity point. For Cut cells we changed the averaging procedure, to reduce the effect of the zero velocities at the closed boundaries on the bottom stress. In the averaging procedure only velocity points which are not at closed boundaries are taken into account. This reduces the artificial boundary layer along closed “staircase” boundaries.

12.10.3 45 degrees closed boundary

For a staircase boundary of 45 degrees (1-1), a special approach is implemented for the advection terms. The velocities are reflected in the boundary line, taking into account the aspect ratio of the grid cells.

$$V_{i,j} = -\frac{\Delta y}{\Delta x} U_{i,j} \quad (12.38)$$

The advection terms for these 45 degree boundary cells are discretised with an explicit first order upwind scheme using the velocities at the boundaries.

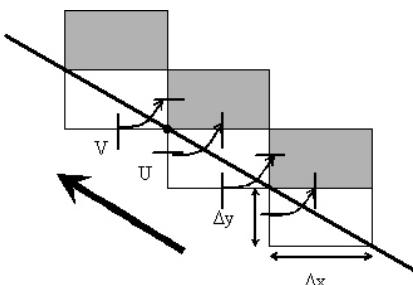


Figure 12.10: Reflection of velocities

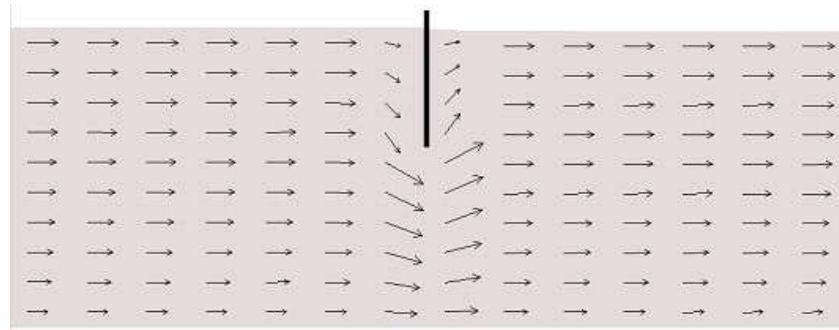


Figure 12.11: Example of a 3D Gate (vertical cross-section)

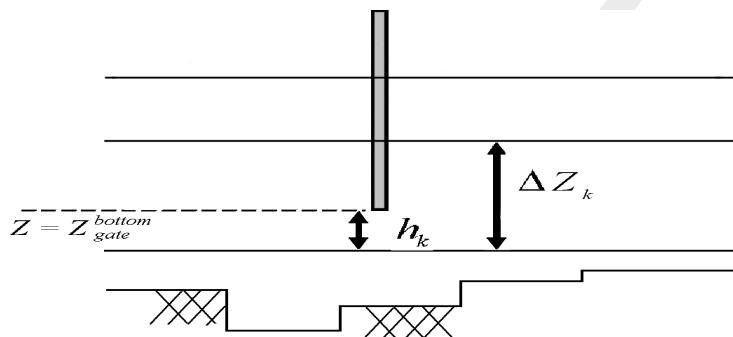


Figure 12.12: Computational layer partially blocked at the bottom of the 3D gate

12.11 Hydraulic structures

Also in the Z -model, the so-called hydraulic structures can be defined to model the effect of obstructions in the flow that cannot be resolved on the horizontal grid (sub-grid) or where the flow is locally non-hydrostatic. Examples of hydraulic structures in civil engineering are gates, sills, sluices, barriers, porous plates, bridges, groynes, weirs. A hydraulic structure generates a loss of energy apart from the loss by bottom friction. At these points an additional force term is added to the momentum equation, to parameterise the extra loss of energy. The term has the form of a friction term with a contraction or discharge coefficient.

In [section 10.9](#), the mathematical formulations and implementation of the hydraulic structures available in Delft3D-FLOW has been described thoroughly. In this section only the parts that deviate will be discussed.

12.11.1 3D Gate

A 3D gate is in fact a thin dam with a limited height/depth (and position in the vertical). It is located at a velocity point and its width is assumed to be zero, so it has no influence on the water volume in the model area. The flow at all intermediate layers of the gate is set to zero. The layer near the top and the layer near the bottom of the gate may be partially blocked. Upstream of the structure the flow is accelerated due to contraction and downstream the flow is decelerated due to expansion.

A 3D gate may be used to model a vertical constriction of the horizontal flow such as near barriers, sluices and Current deflection walls. The vertical constriction of the flow may vary in time by the lowering or raising of the gate. The implementation of the 3D gate has been described in [subsection 10.9.1](#). In this section, the figures showing the 3D gate as defined in the Z -model will be shown.

For more details on the data input requirements for the different type of 3D gates, we refer to Appendix [B.3.1](#).

12.11.2 Quadratic friction

The only hydraulic structure where quadratic friction is applied that is not available within the Z -model is the so-called 2D Weir. The remaining structure may be defined in the Z -model in a similar manner as the σ -model; see Eqs. [10.70](#) to [10.76](#).

12.11.3 Linear friction

The resistance force that is assumed to be linearly dependent on the flow is applied for the rigid sheet. Rigid sheet in the Z -model is treated in a similar manner as in the σ -model; see [Equation 10.98](#).

12.11.4 Floating structure

Floating structures can also be modelled in the Z -model. It is treated in a similar manner as in the σ -model; see [Equation 10.99](#).

12.12 Assumptions and restrictions

The solution of the discretised equations is just an approximation of the exact solution. The accuracy of the solution depends not only on the numerical scheme, but also on the way in which the bottom topography, the geographical area, and the physical processes (turbulence, wave-current interaction) are modelled.

The time integration method strongly influences the wave propagation when applying a large time step. The assumption is made that, by restricting the computational time step, the free surface waves can be propagated correctly.

The open boundaries in a numerical flow model are artificial in the sense that they are introduced to limit the computational area that is modelled. The free surface waves should pass these boundaries completely unhindered. In the numerical model, wave reflections may occur at the open boundaries. These reflections will be observed as spurious oscillations superimposed on the physical results. In Delft3D-FLOW weakly-reflective boundaries are available which diminish these effects.

The open boundary can be divided into segments (sections). The boundary conditions in Delft3D-FLOW are specified for these segments, two values per segment are required, one for each segment end. The boundary condition at internal points within this segment is obtained by linearly interpolation between the end points. Therefore, if the phase variation of the tidal forcing along an open boundary segment is non-linear then the number of open boundary segments should be increased so that the phases at all the segments can be specified. Phase errors may generate an artificial re-circulation flow (eddy) near the open boundary. For steady-state simulations, a similar effect may be observed near the open boundaries if the effect of the Coriolis force on the water level gradient along the open boundary is not taken into account in the boundary conditions.

Care must be taken when time-series of measurements are directly prescribed as forcing functions at the open boundaries. Measurements often contain a lot of undesired noise, due to meteorological or other effects. For tidal flow computations, calibration on processed field data obtained from a tidal analysis or Fourier analysis, avoids this problem.

The following functionalities can not be used in combination with the Z -model:

- ◊ Parallel calculation
- ◊ Roller model
- ◊ Real Time Control
- ◊ Morphology
- ◊ Secondary flow
- ◊ Internal waves
- ◊ Fluid mud
- ◊ Gauss Seidel solver
- ◊ Q2E 2D turbulence model

The following functionalities are not fully tested in combination with the Z -model:

- ◊ Heat model other than option 5
- ◊ Evaporation model
- ◊ Tide generating forces
- ◊ Q-H boundary
- ◊ Drogues
- ◊ Spherical coordinates
- ◊ HLES
- ◊ Structures
- ◊ Domain decomposition
- ◊ Wall friction

DRAFT

References

- Adcroft, A., C. Hill and J. Marshall, 1997. "Representation of topography by shaved cells in a height coordinate ocean model." *Monthly Weather Review* 125 (9): 2293–2315.
- Andrews, D. G. and M. E. McIntyre, 1978. "An exact theory of nonlinear waves on a Lagrangian-mean flow." *Journal of Fluid Mechanics* 89 (4): 609–646.
- Asaeda, T. and J. Imberger, 1992. "Structure of bubble plumes in linearly stratified medium." *Journal of Fluid Mechanics* 249: 35–37.
- Bagnold, R. A., 1966. *An approach to the sediment transport problem from general physics*. US government Print Office.
- Bailard, J. A., 1981. "An Energetics Total Load Sediment Transport Model for Plane Sloping Beaches." *Journal of Geophysical Research* 86 (C11): 10938–10954.
- Bakhmetev, B. A., 1932. *Hydraulics of open channels*, Eng. Soc. Monograph. McGraw-Hill.
- Baptist, M. J., 2005. *Modelling floodplain biogeomorphology*. Ph.D. thesis, Delft University of Technology.
- Barenblatt, G. I., M. Bertsch, R. Dal Passo, V. M. Prostokishen and M. Ughi, 1993. "A mathematical model of turbulent heat and mass transfer in stably stratified shear flow." *Journal of Fluid Mechanics* 253: 341–358.
- Baumert, H. and G. Radach, 1992. "Hysteresis of Turbulent Kinetic Energy in Non-rotational Tidal Flows: A Model Study." *Journal of Geophysical Research* 97 (C3): 3669–3677.
- Beckers, J. M., H. Burchard, J. M. Campin, E. Deleersnijder and P. P. Mathieu, 1998. "Another reason why simple discretizations of rotated diffusion operators cause problems in ocean models: comments on "isoneutral diffusion in a z co-ordinate ocean model""." *American Meteorological Society* 28: 1552–1559.
- Bendat, J. S. and A. G. Piersol, 1971. *Random data: analysis and measurement procedures*. John Wiley & Sons, Inc. New York.
- Benqué, J. P., J. A. Cunge, J. Feuillet, A. Hauguel and F. M. Holly, 1982. "New method for tidal current computation." *Journal of the Waterway, Port, Coastal and Ocean Division, ASCE* 108: 396–417.
- Bijker, E. W., 1967. *Some considerations about scales for coastal models with moveable bed*. Tech. Rep. 50, WL | Delft Hydraulics, Delft, The Netherlands.
- Bijvelds, M. D. J. P., 1997. *Recirculating steady flows in harbours: comparison of numerical computations to laboratory experiments*. Tech. Rep. 1-97, Delft University of Technology, Delft, The Netherlands.
- Bijvelds, M. D. J. P., 2001. *Numerical modelling of estuarine flow over steep topography*. Ph.D. thesis, Delft University of Technology.
- Blumberg, A. F. and G. L. Mellor, 1985. "Modelling vertical and horizontal diffusivities with the sigma co-ordinate system." *Monthly Weather Review* 113 (8): 1379.
- Bulson, P., 1961. "Currents produced by an air curtain in deep water." *Dock and Harbour Authority* 42: 15–22.
- Burchard, H. and H. Baumert, 1995. "On the performance of a mixed-large model based on the k-epsilon turbulence closure." *Journal of Geophysical Research* 100 (C5): 8523–8540.

- Busch, N. E., 1972. "On the mechanics of atmospheric turbulence." In *Workshop on Micrometeorology*, pages 1–65. American Meteorological Society.
- Busnelli, M. M., 2001. *Numerical simulation of free surface flows with steep gradients*. WL | Delft Hydraulics, Delft, The Netherlands. Ph.D. thesis.
- Casulli, V., 1999. "A semi-implicit finite difference method for non-hydrostatic, free surface flows." *International Journal Numerical Methods In Fluids* 30: 425–440.
- Casulli, V. and R. T. Cheng, 1992. "Semi-implicit finite difference methods for three-dimensional shallow water flow." *International Journal Numerical Methods In Fluids* 15 (6): 629–648.
- Christoffersen, J. B. and I. G. Jonsson, 1985. "Bed friction and dissipation in a combined current and wave motion." *Ocean Engineering* 12 (5): 387–423.
- Courant, R. and D. Hilbert, 1962. *Methods of mathematical physics*. Interscience, New York.
- Davies, A. G., R. L. Soulsby and H. L. King, 1988. "A numerical model of the combined wave and current bottom boundary layer." *Journal of Geophysical Research* 93 (C1): 491–508.
- Davies, A. M. and H. Gerritsen, 1994. "An intercomparison of three-dimensional tidal hydrodynamic models of the Irish Sea." *Tellus* 46A: 200–221.
- Dean, R. G. and R. A. Dalrymple, 1991. *Water wave mechanics for engineers and scientists*, vol. 2 of *Advanced series on ocean engineering*. World Scientific Publishing Company.
- Deigaard, 1986. *Not yet known*.
- Deleersnijder, E. and P. Luyten, 1994. "On the practical advantages of the quasi-equilibrium version of the Mellor and Yamada level 2.5 turbulence closure applied to marine modelling." *Appl. Math. Modelling* 18: 281–287.
- Delft3D-GISVIEW UM, 2013. *Delft3D-GISVIEW User Manual*. Deltires, 1.04 ed.
- DIDO UM, 2013. *Delft3D-DIDO User Manual*. Deltires, Delft, 4.00 ed.
- Dingemans, M. W., 1997. *Water Wave Propagation over Uneven Bottoms, Vol. 1 and 2*. Advanced Series on Ocean Engineering, Vol. 13. World Scientific, London.
- Dingemans, M. W., A. C. Radder and H. J. de Vriend, 1987. "Computation of the driving forces of wave-induced currents." *Coastal Engineering* 11: 539–563.
- Dongeren, A. R. van and I. A. Svendsen, 1997. "Absorbing-generating boundary conditions for shallow water models." *Journal of the Waterway, Port, Coastal and Ocean Division, ASCE* 123 (6): 303–313.
- Eckart, C., 1958. "Properties of water, Part II. The equation of state of water and sea water at low temperatures and pressures." *American Journal of Science* 256: 225–240.
- Eijkeren, J. C. H. van, B. J. de Haan, G. S. Stelling and T. L. van Stijn, 1993. "Notes on Numerical Fluid Mechanics, Linear upwind biased methods." In C. B. Vreugdenhil and B. Koren, eds., *Numerical Methods for Advection-Diffusion Problems*, vol. 45, chap. 3, pages 55–91. Vieweg Verlag, Braunschweig. ISSN 0179-9614, ISBN 3-528-07645-3.
- Engelund, F. and E. Hansen, 1967. *A monograph on Sediment Transport in Alluvial Streams*. Teknisk Forlag, Copenhagen.
- Engquist, B. and A. Majda, 1977. "Absorbing boundary conditions for the numerical simulation of waves." *Mathematics of Computation* 31 (139): 629–651.

- Engquist, B. and A. Majda, 1979. "Radiation boundary conditions for acoustic and elastic wave calculations." *Communications on Pure & Applied Mathematics* 32: 313–357.
- Fannelop, T. and K. Sjoen, 1980. "Hydrodynamics of air-bubble plumes." In *Proc. AIAA 18-th Aerospace Sci. Meeting*.
- Farraday, R. V. and F. G. Charlton, 1983. *Hydraulic factors in bridge design*. Tech. rep., Hydraulics Research Station Ltd. Wallingford, Oxfordshire, England.
- Forester, C. K., 1979. "Higher Order Monotonic Convective Difference Schemes." *Journal of Computational Physics* 23: 1–22.
- Fredsøe, J., 1982. "Shape and dimensions of stationary dunes in rivers." *Journal of the Hydraulics Division, ASCE* 108: 932–947.
- Fredsøe, J., 1984. "Turbulent boundary layer in wave-current interaction." *Journal of Hydraulic Engineering* 110: 1103–1120.
- Fredsøe, J. and R. Deigaard, 1992. *Mechanics of Coastal Sediment Transport*. No. 3 in Advanced series on ocean engineering. World Scientific Publishing.
- French, R. H., J. Zseleczky and R. Margolies, eds., 1987. *Open-Channel Hydraulics*. McGraw-Hill, New York. 2nd edition.
- Gaeuman, D., E. Andrews, A. Krause and W. Smith, 2009. "Predicting fractional bed load transport rates: Application of the Wilcock-Crowe equations to a regulated gravel bed river." *Water Resources Research* 45.
- Galappatti, R., 1983. *A depth integrated model for suspended transport*. Tech. rep., Delft University of Technology, Delft, The Netherlands.
- Germano, M., U. Piomelli, P. Moin and W. H. Cabot, 1991. "A dynamic subgrid-scale eddy viscosity model." *Physics of Fluids A* 3 (7): 1760–1765.
- Gill, A. E., 1982. *Atmosphere-Ocean dynamics*, vol. 30 of *International Geophysics Series*. Academic Press.
- Golub, G. H. and C. F. Van Loan, 1989. *Matrix computations*. Baltimore: John Hopkins University Press.
- GPP UM, 2013. *Delft3D-GPP User Manual*. Deltares, 2.14 ed.
- Grant, W. D. and O. S. Madsen, 1979. "Combined wave and current interaction with a rough bottom." *Journal of Geophysical Research* 84 (C1): 1797–1808.
- Grasmeijer, B. and L. Van Rijn, 1998. "Breaker bar formation and migration." *Coastal Engineering* pages 2750–2758. Virginia, USA.
- Gresho, P. M. and R. L. Lee, 1981. "Don't suppress the wiggles, they're telling you something." *Computer and Fluids* 9: 223–253.
- Groeneweg, J., 1999. *Wave-current interactions in a generalized Lagrangian mean formulation*. Delft University of Technology, Delft, The Netherlands. Ph.D. thesis.
- Groeneweg, J. and G. Klopman, 1998. "Changes of the mean velocity profiles in the combined wave-current motion in a GLM formulation." *Journal of Fluid Mechanics* 370: 271–296.
- Haney, R. L., 1991. "On the pressure gradient force over steep topography in sigma coordinate models." *Journal of Physical Oceanography* 21: 610–619.

- Hirsch, C., 1990. *Numerical computation of internal and external flows*. John Wiley & Sons, New York.
- Houwing, E. J., I. C. Tánczos, A. Kroon and M. B. de Vries, 2000. "Interaction of submerged vegetation, hydrodynamics and turbidity; analysis of field and laboratory studies." *Conference INTERCOH2000*.
- Huang, W. and M. Spaulding, 1996. "Modelling horizontal diffusion with sigma coordinate system." *Journal of Hydraulic Engineering* 122 (6): 349–352.
- Huynh-Thanh, S. and A. Temperville, 1991. "A numerical model of the rough turbulent boundary layer in combined wave and current interaction." In R. L. Soulsby and R. Bettes, eds., *Sand transport in rivers, estuaries and the sea*, pages 93–100. Balkema Rotterdam.
- Ikeda, S., 1982. "Incipient Motion of Sand Particles on Side Slopes." *Journal of the Hydraulics Division, ASCE* 108 (1): 95–114.
- Ikeda, S., 1988. *Not yet known*.
- Isobe, M. and K. Horikawa, 1982. "Study on water particle velocities of shoaling and breaking waves." *Coastal Engineering in Japan* 25: 109–123.
- Jagers, H. R. A. and S. A. H. van Schijndel, 2000. *3D computations around structures*. Tech. Rep. Q2487, WL | Delft Hydraulics, Delft, The Netherlands. In Dutch (3D rekenen rondom constructies).
- Kalkwijk, J. P. T. and R. Booij, 1986. "Adaptation of secondary flow in nearly horizontal flow." *Journal of Hydraulic Research* 24 (1): 19–37.
- Kester, J. A. T. M. van, 1994. *Validatie DELFT3D voor menglaag proef. Fase 1: Verbeterde implementatie k-ε model*. Tech. Rep. Z810, WL | Delft Hydraulics, Delft, The Netherlands.
- Kirkpatrick, M. P., S. W. Armfield and J. H. Kent, 2003. "A representation of curved boundaries for the solution of the Navier-Stokes equations on a staggered three-dimensional Cartesian grid." *Journal of Computational Physics* 184: 1–36.
- Klopstra, D., H. J. Barneveld and J. M. Van Noortwijk, 1996. *Analytisch model hydraulische ruwheid van overstroomde moerasvegetatie*. Tech. Rep. PR051, HKV consultants, Lelystad, The Netherlands. Commissioned by Rijkswaterstaat/RIZA, The Netherlands.
- Klopstra, D., H. J. Barneveld, J. M. Van Noortwijk and E. H. Van Velzen, 1997. "Analytical model for hydraulic roughness of submerged vegetation." In *The 27th IAHR Congress, San Francisco, 1997; Proceedings of Theme A, Managing Water: Coping with Scarcity and Abundance*, pages 775–780. American Society of Civil Engineers (ASCE), New York.
- Koch, F. G. and C. Flokstra, 1980. "Bed level computations for curved alluvial channels." In *Proceedings of the XIXth congress of the International Association for Hydraulic Research, 2-7 Feb. 1981, New Delhi, India*, vol. 2, pages 357–364.
- Kolmogorov, A. N., 1942. "Equations of turbulent motion in incompressible fluid." *Izv. Akad. Nauk. SSR, Seria fizicheska Vi No.1 2 (1-2)*: 56–58. English translation: 1968 Imperial College, Mech. Eng. Dept. Rept. ON/6.
- Lane, A., 1989. *The heat balance of the North Sea*. Tech. Rep. 8, Proudman Oceanographic Laboratory.
- Leendertse, J. J., 1967. *Aspects of a computational model for long-period water-wave propagation*. RM-5294-RR. Rand Corporation, Santa Monica. Ph.D. thesis.

- Leendertse, J. J., 1990. "Turbulence modelling of surface water flow and transport: part IVa." *Journal of Hydraulic Engineering* 114 (4): 603–606.
- Leendertse, J. J., R. C. Alexander and S.-K. Liu, 1973. *A three-dimensional model for estuaries and coastal seas*. Volume I: Principles of Computations Report R-1417-OWRT, Rand Corporation, Santa Monica, CA. Volume II: Aspects of computation, R-1764-OWRT, Volume III: The interim program, R-1884-OWRT, Volume IV: Turbulent energy computation, R-2187-OWRT.
- Leendertse, J. J. and E. C. Gritton, 1971. *A water quality simulation model for well mixed estuaries and coastal seas: Vol. II, Computation Procedures*. Tech. Rep. R-708-NYC, Rand Corporation.
- Leer, B. van, 1974. "Towards the ultimate conservative difference scheme II. Monotonicity and conservation combined in a second order scheme." *Journal of Computational Physics* 32: 101–136.
- Lesser, 2003. *Not yet known*.
- Lesser, G., J. van Kester and J. A. Roelvink, 2000. *On-line sediment transport within Delft3D-FLOW*. Tech. Rep. Z2899, wl.
- Lesser, G. R., J. A. Roelvink, J. A. T. M. van Kester and G. S. Stelling, 2004. "Development and validation of a three-dimensional morphological model." *Coastal Engineering* 51: 883–915.
- McDougall, T., 1978. "Bubble plumes in stratified environments." *Journal of Fluid Mechanics* 85: 655–672.
- Meijer, D., 1998. *Modelproeven overstroomde vegetatie*. Tech. rep., HKV Lijn in water. In Dutch.
- Mellor, G. L. and T. Yamada, 1974. "A hierarchy of turbulence closure models for planetary boundary layers." *Journal of Atmospheric Science* 31: 1791–1896.
- Mellor, G. L. and T. Yamada, 1982. "Development of a turbulence closure model for geophysical fluid problems." *Review of Geophysics and Space Physics* 20 (4): 851–875.
- Meyer-Peter, E. and R. Müller, 1948. "Formulas for bed load transport." In *Proceedings of the 2nd Congress IAHR, Stockholm*, vol. 2, pages 39–64.
- Miles, J., 1987. "Richardson's number revisited." In E. J. List and G. H. Jirka, eds., *3rd International Symp. Stratified Flows, February 3-5, Pasadena, California*, pages 1–7. asce.
- Milgram, J., 1983. "Mean flow in round bubble plumes." *Journal of Fluid Mechanics* 133: 345–376.
- Millero, F. J. and A. Poisson, 1981. "International one-atmosphere equation of state of sea water." *Deep-Sea Research* 28A (6): 625–629.
- Munk, W. H. and E. R. Anderson, 1948. "Notes on the theory of the thermocline." *Journal of Marine Research* 7 (3): 276–295.
- Murakami, M., Y. Oonisishi and H. Kunishi, 1985. "A numerical simulation of the distribution of water temperature and salinity in the Seto Inland Sea." *Journal of the Oceanographical Society of Japan* 41: 221–224.
- Myrhaug, D. and O. H. Slaattelid, 1990. "A rational approach to wave-current friction coefficients for rough, smooth and transitional turbulent flow." *Coastal Engineering* 14: 265–293.

- Nikuradse, J., 1933. *Laws of flow in rough pipes*. Tech. Rep. Forschungsheft 361, Forschung auf dem Gebiete des Ingenieurwesens, VDI Verlag, Berlin, Germany. In German (Strömungsgesetze in rauhren rohren) English translation: Laws of flow in rough pipes, NACA TM 1292, 1950.
- Nipius, K. G., 1998. *Transverse transport modelling using Bailard applied to Grevelingen-mouth delta*. Delft University of Technology, Delft, The Netherlands. M.Sc. thesis, in Dutch (Dwarstransportmodellering m.b.v. Bailard toegepast op de Voordelta Grevelingenmonding).
- O'Connor, B. A. and D. Yoo, 1988. "Mean bed friction of combined wave-current flow." *Coastal Engineering* 12: 1–21.
- Oberez, A., 2001. *Turbulence modeling of hydraulic roughness of submerged vegetation*. Master's thesis, UNESCO IHE, Delft. H.E.100.
- Octavio, K. A. H., G. H. Jirka and D. R. F. Harleman, 1977. *Vertical Heat Transport Mechanisms in Lakes and Reservoirs*. Tech. Rep. 22, Massachusetts Institute of Technology.
- Pacanowski, R. C. and A. Gnanadesikan, 1998. "Transient response in a Z -level ocean model that resolves topography with partial cells." *Monthly Weather Review* 126: 3248–3270.
- Parker, G. and E. D. Andrews, 1985. "Sorting of bed load sediment by flow in meander bends." *Water Resources Research* 21: 1361–1373.
- Partheniades, E., 1965. "Erosion and Deposition of Cohesive Soils." *Journal of the Hydraulics Division, ASCE* 91 (HY 1): 105–139.
- Phillips, N. A., 1957. "A co-ordinate system having some special advantages for numerical forecasting." *Journal of Meteorology* 14: 184–185.
- Platzek, F. W., G. S. Stelling, J. A. Jankowski and R. Patzwahl, 2012. "On the representation of bottom shear stress in z-layer models." In *Hydroinformatics 2012*. Hamburg, Germany.
- Postma, L., G. S. Stelling and J. Boon, 1999. "Three-dimensional water quality and hydrodynamic modelling in Hong Kong. Stratification and water quality." In *Proceedings of the 2nd International Symp. on Environmental Hydraulics, Hong Kong, December 1998*, pages 43–49. Balkema, Rotterdam.
- Prandtl, L., 1945. "Über ein neues Formelsystem für die ausgebildete Turbulenz." *Nachrichten von der Akademie der Wissenschaften in Göttingen. Mathematisch-Physikalische Klasse* pages 6–19.
- RGFGRID UM, 2016. *Delft3D-RGFGRID User Manual*. Deltares, 5.00 ed.
- Richardson, J. F. and W. N. Zaki, 1954. "Sedimentation and fluidization: Part I." *I. Trans. Institution of Chemical Engineers* 32: 35–53.
- Richardson, L. F., 1920. "The supply of energy from and to atmospheric eddies." *Proceedings of the Royal Society London* A97: 345–373.
- Rienecker, M. M. and J. D. Fenton, 1981. "A Fourier approximation method for steady water waves." *Journal of Fluid Mechanics* 104: 119–137.
- Rijn, L. C. van, 1984a. "Sediment transport, Part I: bed load transport." *Journal of Hydraulic Engineering* 110 (10): 1431–1456.
- Rijn, L. C. van, 1984b. "Sediment transport, Part II: suspended load transport." *Journal of Hydraulic Engineering* 110 (11): 1613–1640.

- Rijn, L. C. van, 1984c. "Sediment transport, Part III: bed form and alluvial roughness." *Journal of Hydraulic Engineering* 110 (12): 1733–1754.
- Rijn, L. C. van, 1990. *Principles of fluid flow and surface waves in rivers, estuaries, seas and oceans*. Aqua Publications, The Netherlands.
- Rijn, L. C. van, 1993. *Principles of Sediment Transport in Rivers, Estuaries and Coastal Seas*. Aqua Publications, The Netherlands.
- Rijn, L. C. van, 2001. *General view on sand transport by currents and waves : data analysis and engineering modelling for uniform and graded sand (TRANSPOR 2000 and CROS-MOR 2000 models)*. Z2899.20 / Z2099.30 / Z2824.30. WL | Delft Hydraulics, Delft, The Netherlands.
- Rijn, L. C. van, 2003. "Sediment transport by currents and waves; general approximation formulae Coastal Sediments." In *Corpus Christi, USA*.
- Rijn, L. C. van, 2007. "Unified View of Sediment Transport by Currents and Waves. I: Initiation of Motion, Bed Roughness, and Bed-Load Transport." *Journal of Hydraulic Engineering* 133 (6): 649–667.
- Rijn, L. C. van, J. A. Roelvink and W. T. Horst, 2000. *Approximation formulae for sand transport by currents and waves and implementation in DELFT-MOR*. Tech. Rep. Z3054.40, WL | Delft Hydraulics, Delft, The Netherlands.
- Rijn, L. C. van, D. R. Walstra and M. v. Ormondt, 2004. *Description of TRANSPOR2004 and implementation in Delft3D-ONLINE*. Tech. Rep. Z3748.10, WL | Delft Hydraulics, Delft, The Netherlands.
- Rijn, L. van, D. Walstra, B. Grasmeijer, J. Sutherland, S. Pan and J. Sierra, 2003. "The predictability of cross-shore bed evolution of sandy beaches at the time scale of storms and seasons using process-based profile models." *Coastal Engineering* 47: 295–327.
- Ris, R. C., 1997. *Spectral Modelling of Wind Waves in Coastal Areas*. Communications on Hydraulic and Geotechnical Engineering, report 97-4. Delft University of Technology, Delft, The Netherlands. Ph.D. thesis.
- Rodi, W., 1984. "Turbulence models and their application in Hydraulics, State-of-the-art paper article sur l'état de connaissance." IAHR Paper presented by the IAHR-Section on Fundamentals of Division II: Experimental and Mathematical Fluid Dynamics, The Netherlands.
- Roelvink, J. A., 1993. "Dissipation in random wave groups incident on a beach." *Coastal Engineering* 19: 127–150.
- Roelvink, J. A., 2003. *Not yet known*.
- Roelvink, J. A. and M. J. F. Stive, 1989. "Bar-generating cross-shore flow mechanisms on a beach." *Journal of Geophysical Research* 94 (C4): 4785–4800.
- Roelvink, J. A. and D. J. R. Walstra, 2004. "Keeping it simple by using complex models." In *Proceedings of the 6th International Conference on Hydro-Science and Engineering. Advances in Hydro-Science and Engineering*, vol. VI, page p. 12. Brisbane, Australia.
- Ruessink, G. and J. A. Roelvink, 2000. *Validation of On-line Mud Transport within Delft3D-FLOW*. Tech. rep., WL | Delft Hydraulics, Delft, The Netherlands.
- Ryan, P. J., D. R. F. Harleman and K. D. Stolzenbach, 1974. "Surface Heat Loss From Cooling Ponds." *Water Resources Research* 10 (5): 930–938.

- Schwiderski, E. W., 1980. "On Charting Global Ocean Tides." *Review of Geophysics and Space Physics* 18 (10): 243–268.
- Simonin, O., R. E. Uittenbogaard, F. Baron and P. L. Viollet, 1989. "Possibilities and limitations to simulate turbulence fluxes of mass and momentum, measured in a steady stratified mixing layer." In *XXIII IAHR Congress, Ottawa, August 21-25*, pages A55–A62. National Research Council, Canada.
- Slørdal, L. H., 1997. "The pressure gradient force in sigma-co-ordinate ocean models." *International Journal Numerical Methods In Fluids* 24: 987–1017.
- Smith, S. D. and E. G. Banke, 1975. "Variation of the sea surface drag coefficient with wind speed." *Quarterly Joournal of the Royal Meteorological Society* 101: 665–673.
- Soulsby, R., 1997. *Dynamics of marine sands, a manual for practical applications*. Thomas Telford, London.
- Soulsby, R. L., A. G. Davies, J. Fredsøe, D. A. Huntley, I. G. Jonnson, D. Myrhaug, R. R. Simons, A. Temperville and T. J. Zitman, 1993a. "Bed shear stresses due to combined waves and currents." In *Abstracts-in-depth of the Marine Science and Technology G8-M overall workshop, Grenoble.*, pages 2.1-1/2.1–4.
- Soulsby, R. L., L. Hamm, G. Klopman, D. Myrhaug, R. R. Simons and G. P. Thomas, 1993b. "Wave-current interaction within and outside the bottom boundary layer." *Coastal Engineering* 21: 41–69.
- Stelling, G. S., 1984. *On the construction of computational methods for shallow water flow problems*. Tech. Rep. 35, Rijkswaterstaat.
- Stelling, G. S. and S. P. A. Duinmeijer, 2003. "A staggered conservative scheme for every Froude number in rapidly varied shallow water flows." *International Journal Numerical Methods In Fluids* 43: 1329–1354.
- Stelling, G. S. and J. A. T. M. van Kester, 1994. "On the approximation of horizontal gradients in sigma co-ordinates for bathymetry with steep bottom slopes." *International Journal Numerical Methods In Fluids* 18: 915–955.
- Stelling, G. S. and J. J. Leendertse, 1992. "Approximation of Convective Processes by Cyclic AOI methods." In M. L. Spaulding, K. Bedford and A. Blumberg, eds., *Estuarine and coastal modeling, Proceedings 2nd Conference on Estuarine and Coastal Modelling, ASCE*, pages 771–782. Tampa.
- Stelling, G. S., A. K. Wiersma and J. B. T. M. Willemse, 1986. "Practical aspects of accurate tidal computations." *Journal of Hydraulic Engineering* 112 (9): 802–817.
- Stive, M. J. F., 1986. "A model for cross-shore sediment transport." In *Proceedings 20th International Coastal Engineering Conference*, pages 1550–1564. American Society of Civil Engineers, New York.
- Swart, 1974. *Offshore sediment transport and equilibrium beach profiles*. Ph.D. thesis, Delft University of Technology, Delft, The Netherlands. Delft Hydraulics Publ. 131.
- Sweers, H. E., 1976. "A nomogram to estimate the heat exchange coefficient at the air-water interface as a function of windspeed and temperature; a critical survey of some literature." *Journal of Hydrology* 30: –.
- Talmon, A. M., N. Struiksma and M. C. L. M. van Mierlo, 1995. "Laboratory measurements of the direction of sediment transport on transverse alluvial-bed slopes." *Journal of Hydraulic Research* 33 (4): 495–517.

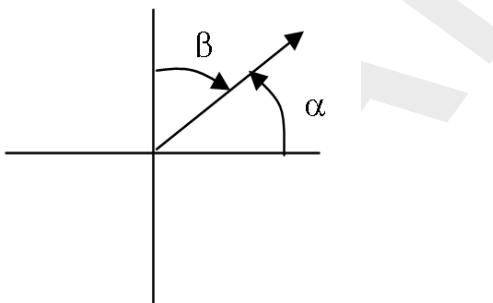
- Taylor, G. I., 1931. "Effect of variation in density on the stability of superposed streams of fluid." *Proceedings Camb. Phil. Soc.* 23: 730–731.
- Thatcher, M. L., 1981. *Not yet known.*
- Thatcher, M. L. and D. R. F. Harleman, 1972. *A mathematical model for the prediction of unsteady salinity intrusion in estuaries.* Report no. 144, MIT School of Engineering Massachusetts Institute of Technology, Department of Civil Engineering.
- TIDE UM, 2013. *Delft3D-TIDE User Manual.* Deltares, 3.00 ed.
- Uittenbogaard, R. E., 1995. *The importance of internal waves for mixing in a stratified estuarine tidal flow.* Ph.D. thesis, Delft University of Technology, Delft, The Netherlands.
- Uittenbogaard, R. E., 1998. *Model for eddy diffusivity and viscosity related to sub-grid velocity and bed topography.* Tech. rep., WL | Delft Hydraulics, Delft, The Netherlands.
- Uittenbogaard, R. E., dec. 2000. *Hydrodynamics of flow over plants, internal communication.* Tech. rep., WL | Delft Hydraulics, Delft, The Netherlands.
- Uittenbogaard, R. E. and F. Baron, 1989. "A proposal: extension of the q2e-eps model for stably stratified flows with transport of Internal Wave Energy." In *7th Turbulent Shear Flows Symp., Stanford, August*, pages 21–23. Sanford University.
- Uittenbogaard, R. E., J. A. T. M. van Kester and G. S. Stelling, 1992. *Implementation of three turbulence models in 3D-TRISULA for rectangular grids.* Tech. Rep. Z81, WL | Delft Hydraulics, Delft, The Netherlands.
- Uittenbogaard, R. E. and B. Van Vossen, 2003. "Subgrid-scale model for Quasi-2D turbulence in shallow water." In *In: Shallow flows: proceedings of the international symposium*, pages 575–582.
- UNESCO, 1981a. *Background papers and supporting data on the international equation of state 1980.* Tech. Rep. 38, UNESCO.
- UNESCO, 1981b. *The practical salinity scale 1978 and the international equation of state of seawater 1980.* Tech. Rep. 36, UNESCO. Tenth report of the Joint Panel on Oceanographic Tables and Standards (1981), (JPOTS), Sidney, B.C., Canada.
- Verboom, G. K. and A. Segal, 1986. "Weakly reflective boundary conditions for shallow water equations." In *25th Meeting Dutch Working group on Numerical Flow Simulations, Delft.* Delft University of Technology, Delft, The Netherlands.
- Verboom, G. K. and A. Slob, 1984. "Weakly-reflective boundary conditions for two-dimensional water flow problems." *Advances in water resources* 7 (4): 192–197. Also presented at "5th International Conference on Finite Elements in Water Resources, Burlington, Vermont, June 1984".
- Vermaas, H., 1987. *Energylosses due to weirs.* Tech. Rep. Q92, WL | Delft Hydraulics, Delft, The Netherlands. In Dutch (Energieverliezen door overlaten: Een gewijzigde berekeningsprocedure voor WAQUA-rivieren versie).
- Vossen, B. van, 2000. *Horizontal Large Eddy simulations; evaluation of flow computations with Delft3D-FLOW.* Tech. Rep. MEAH-197, WL | Delft Hydraulics, Delft, The Netherlands.
- Vreugdenhil, C. B., 1989. *Computational Hydraulics: an Introduction.* Springer-Verlag.
- Vreugdenhil, C. B., 1994. *Numerical methods for shallow-water flow.* Water Science and technology library. Kluwer Academic Publishers.

- Walstra, D. J. R. and J. A. Roelvink, 2000. "3D Calculation of Wave Driven Cross-shore Currents." In *Proceedings 27th International Conference on Coastal Engineering, Sydney, Australia, July 16-21, 2000*, pages 1050–1063.
- Walstra, D. J. R., J. A. Roelvink and J. Groeneweg, 2000. "Calculation of Wave-Driven Currents in a 3D Mean Flow Model." In *Proceedings 27th International Conference on Coastal Engineering, Sydney, Australia, July 16-21, 2000*.
- Wang, Z. B. and J. C. Winterwerp, 1992. *A model to simulate the transport of fluid mud*. Tech. Rep. Z163, WL | Delft Hydraulics, Delft, The Netherlands.
- WAVE UM, 2013. *Delft3D-WAVE User Manual*. Deltares, 3.03 ed.
- Weare, T. J., 1979. "Errors arising from irregular boundaries in ADI solutions of the shallow water equations." *International Journal Numerical Methods Engineering* 14: 921–931.
- Wijbenga, J. H. A., 1990. *Representation of extra energy losses in RIVCUR*. Tech. Rep. Q910, WL | Delft Hydraulics, Delft, The Netherlands. In Dutch (Weergave van extra energieverlies in RIVCUR), research for Rijkswaterstaat, Dienst Binnenwateren/RIZA.
- Wilcock, P. and J. Crowe, 2003. "Surface-based transport model for mixed-size sediment." *Journal of Hydraulic Engineering* 129 (2): 120–128.
- Winterwerp, J. C. and R. E. Uittenbogaard, 1997. *Sediment transport and fluid mud flow*. Tech. Rep. Z2005, WL | Delft Hydraulics, Delft, The Netherlands.
- Winterwerp, J. C., Z. B. Wang, J. A. T. M. van Kester and F. J. Verweij, 1999. "On the far-field impact of Water Injection Dredging." *J. Waterway, Port, Coastal and Ocean Engineering* submitted: –.
- WL | Delft Hydraulics, 1998. *Interaction of submerged vegetation and physical processes; Delft, The Netherlands*. Research report, WL | Delft Hydraulics, Delft, The Netherlands. (in Dutch).

Glossary of terms

area	Geometric domain for the models.
astronomical tidal constituent	Amplitude and phase of the tidal constituents which represent the tide. See also Delft3D-TIDE.
bathymetry	The measurement of depths of the model area. Represented in Delft3D-FLOW by a matrix of depth values; each value is defined in the right upper corner of the corresponding grid cell. Prepared as an attribute file for Delft3D-FLOW or entered in the MDF-file as a uniform value for the whole grid. Default: 0.0. Unit: meter.
bottom roughness coefficient	Measure of the resistance of the flow to the bottom. Physical parameter for Delft3D-FLOW. Defined in the middle of grid sides. Prepared as an input file for Delft3D-FLOW (containing non-uniform values in U and V direction) or entered in the MDF-file as a uniform value; Presentation: table or single value. Given according to: Manning, Chézy [$m^{1/2}/s$], White Colebrook.
boundary section	Boundaries are separation lines between the model area and the outside world. Boundaries can be divided in open boundaries, located in open water, and closed boundaries representing the land-water interface. A boundary section is a part of a boundary on which boundary conditions are prescribed.
boundary conditions	Boundary conditions describe the influence of the outside world on the inside of the model area.
calibration	Tuning of model parameters such that the simulation results match an observed data set within a prescribed accuracy interval.
coupling program	Program which performs some operations on Delft3D-FLOW output files in order to create input files for D-Water Quality or D-Waq PART. With the coupling program, aggregation in time and/or space is possible. Delft3D-FLOW itself can now also perform these actions (See B.18).
cross-section	A line defined along a fixed ξ - or η -co-ordinate, where the sum of computed fluxes, flux rates, fluxes of matter (if exist) and transport rates of matter (if exists) are stored sequentially in time at a prescribed interval.
current	In case of 2D computation: Speed and direction of the hydrodynamic depth-averaged flow. In case of 3D computation: Speed and direction of the hydrodynamic flow in a layer.
cyclic tide	Tidal condition in which the tide repeats itself (e.g. cyclic semi-diurnal tide which repeats itself after its period of approximately 24 hours and 50 minutes).
default	Initial value for a parameter, to be used in the FLOW Input Processor at simulation start time if no MDF-file has been opened.
GPP	Program for the visualisation and animation of results of Delft3D modules.
QUICKIN	Program for the generation and manipulation of grid related data, such as bathymetry, or initial conditions.
Delft3D-QUICKPLOT	A second program for the visualisation and animation of results of Delft3D modules.
RGFGRID	Program for the generation of orthogonal curvilinear grids.
Delft3D-TRIANA	Program for the analysis of observed and simulated tide, used during calibration.
Delft3D-FLOW	Simulation program for tidal and wind driven flow, including the effect of density differences due to a non-uniform heat and salt concentration distribution. Previously known as TRISULA.



depth averaged speed	Hydrodynamic speed, averaged over the depth.
depth contours	Presentation of the bathymetry by iso-lines of depth values.
discharge	Location where water and possibly constituents dissolved in the water are released into or subtracted from the model area.
discharge rate	The amount of water and possibly constituents dissolved in the water released into or subtracted from the model area per unit of time.
domain	Range of values a parameter can have for meaningful results. A domain is represented by its lower and upper limit.
drogues	Floating objects moving with the flow.
dry area/dry point	Part of the model area that is not flooded. Represented by a collection of cells in the grid (dry points) which are either temporarily or permanently dry.
drying and flooding	A process in which points or sub-areas of the model area are becoming dry or wet depending on the local water depth.
flow rate	Volume of fluid passing a cross-section per unit of time.
flow velocity	Speed and direction of a water particle. In the staggered grid the speed is computed at the water level point using only the velocity components with the same grid co-ordinates as the water level point, i.e. the velocity components are not averaged over the velocity points on both sides of the water level point before being used.
flow direction	Direction in which the flow of a water particle is moving. North = 0° ; East = 90° ; South = 180° and West = 270° . The convention is:  A diagram illustrating the relationship between flow direction and mathematical angles. It shows a horizontal axis and a vertical axis forming a coordinate system. A vector representing flow direction makes an angle β with the positive vertical axis. Another angle, α , is shown between the positive horizontal axis and the projection of the flow vector onto the horizontal plane. Arrows indicate the direction of increasing angles. flow direction $\beta = 90^\circ - \alpha$ with: α mathematical angle β angle according Nautical convention
grid (horizontal)	Structured set of virtual points covering the model area in the horizontal direction on which the simulation results are obtained. In Delft3D-FLOW two types of horizontal grid co-ordinate systems can be applied: a Cartesian or a spherical co-ordinate system. In both systems the grid is curvilinear and orthogonal.
grid (vertical)	Structured set of virtual points covering the model area in the vertical direction on which the simulation results are obtained. In Delft3D-FLOW two types of vertical grids can be distinguished: a σ -grid and a Z -grid.
harmonic components	Amplitudes and phases that constitute a time dependent signal, such as a tide. Generally these components are obtained as a result of Fourier analysis. Fourier components are often prescribed as model forcing when a cyclic tidal movement is required.
high water	Time of the tide with maximum water level.
hydrostatic pressure	Pressure exerted by a fluid due to its weight. When the vertical motion of fluid is small compared to the motion in the horizontal direction we may still apply this principle in the computation.
history file	File that contains the results of a simulation in monitoring stations as

horizontal velocity	a function of time. In case of 2D computation: Speed and direction of the hydrodynamic depth-averaged flow. In case of 3D computation: Speed and direction of the hydrodynamic flow in a layer.
	Remark:
	◊ the velocity in a layer is in a σ -layer and is not in a horizontal plane.
hydrodynamic conditions	Set of data which determines the input for Delft3D-FLOW.
integrity check	Verification of input and output data on both its domain and its internal consistency.
logarithmic speed/velocity profile	An expression for the velocity distribution over the depth. The velocity is assumed to be a logarithmic function of the water depth and depends on the depth averaged speed.
lon/lat	Longitudinal and latitudinal co-ordinates in degrees, minutes and seconds.
map file	File that contains the results of a simulation in all grid points at specific instances of time.
model area	A part of the physical space (the world) that is (schematically) represented in the simulation. The model area is connected to the outer world through closed and open boundaries. The forcing of the outer world on the model area is described by boundary conditions and external forces such as wind.
monitoring station	Virtual point in the model area, where computational results, such as the current, the water level and/or the concentration of constituents are monitored as a function of time. Also called observation point.
morphological dynamics	Change of bathymetry due to sedimentation and erosion. Computed with the module 3DMOR feature of FLOW.
neap tide	Tide with a small tidal range.
non-hydrostatic pressure	Pressure exerted by the fluid due to its (relatively large) vertical motion. The vertical motion, that may be in the order of magnitude of (or greater than) the horizontal motion, may be induced by buoyancy or by obstacles or a hydraulic jump.
observation station	Monitoring point for current, water level and/or temperature and salinity. Observation points are defined at the centre of grid cells. Delft3D-FLOW writes the results of the simulation in this point to a history file.
print file	File that contains a sub-set of the results in ASCII format that can be listed on a printer.
restart	Start of a simulation using the results of a previous run as initial conditions.
restart file	File with the simulation results at the last time step of a previous simulation, to be used as initial conditions in a restart run.
river outflow/run-off	Amount of water flowing from the river into the model area.
roughness	Equivalent to 'bottom roughness'. Recommended term is bottom roughness.
salinity	Constituent of (sea) water. Salinity causes a density induced flow additional to the hydrodynamic flow.
scenario	Set of conditions which determine the hydrodynamic simulation completely. A scenario is defined (stored) in an MDF-file and its attribute files.
simulation time	Time period between simulation start and stop time, expressed as real time or in the number of time steps.
spin-up time	Time required by the model to adjust itself to match the prescribed



	boundary and initial conditions. Also known as initial period, transient time or warming-up time.
spring tide	Tide with a large tidal range.
staggered grid	Grid in which the water level, velocity components and depth points are defined at different locations.
steady state	Equilibrium situation; all time variations are absent.
temperature	Thermodynamic condition of the material concerned. In Delft3D-FLOW the temperature is treated as a constituent. Temperature causes a density induced flow additional to the hydrodynamic flow.
test run	Simulation run to check the hydrodynamic behaviour of the model as described by the scenario.
thin dam	A virtual dam along the side of a grid cell across which no flow exchange is possible. Thin dams are defined in the middle of the grid sides.
threshold	Water depth above which a dry grid cell is becoming wet.
tidal constants	Amplitude and phase of the tidal constituents which represent the tide. See also Delft3D-TIDE.
tidal cycle	Time period of the dominant tidal component; about 12 hours and 25 minutes for a semi-diurnal tide and about 24 hours and 50 minutes for a diurnal tide.
TIDE	Program for the analysis of observed or simulated water levels or flows in terms of astronomical tidal components, in order to calculate a geometric series to represent the tide.
time frame	Start and stop time of the simulation and the forcing of all (sub-)processes in the simulation, including writing the results to file.
time history	Sequence of numbers giving the value of one or more parameters in an observation point at sequential moments.
time-series	Sequence of numbers giving the value of one or more parameters used in the model input, or of the simulation results in an observation point at sequential moments.
time step	Time interval at which the results of the simulation are computed.
vector field	Flow data in all grid cells expressed as arrows at a certain instance in time. The arrow represents the speed and direction of the flow.
verification	Evaluation of the quality of simulated results by comparison with observed data. Also called validation.
viscosity	Measure of the resistance of the fluid to the flow.
water level	Defined as a uniform value in the MDF-file or as non-uniform values in every grid point in an attribute file.
wind velocity	Elevation of the free water surface above some reference level.
Z-grid	Magnitude and direction of wind; usually defined relative to true North, positive angle measured clock wise.
Z-model	Vertical grid in a 3D model that is strictly horizontal. Index $k = 1$ in the Z-grid refers to the bottom layer and index $k = k_{max}$ to the surface layer.
σ -grid	A 3D model that has been defined with a vertical grid that is strictly horizontal.
σ -model	Vertical grid in a 3D model that follows the depth profile, i.e. the so-called σ -plane. Index $k = 1$ in the σ -grid refers to the surface layer and index $k = k_{max}$ to the bottom layer.
	A 3D model that has been defined with a vertical grid using σ -planes.

A Files of Delft3D-FLOW

A.1 MDF-file

A.1.1 Introduction

File contents	The Master Definition FLOW file (MDF-file) is the input file for the hydrodynamic simulation program.
Filetype	ASCII
File format	Free formatted
Filename	<name.mdf>
Generated	FLOW-GUI or manually offline

The Master Definition FLOW file (MDF-file) is the input file for the hydrodynamic simulation program. It contains all the necessary data required for defining a model and running the simulation program. In the MDF-file you can define attribute files in which relevant data (for some parameters) are stored. This is especially useful when parameters contain a large number of data (e.g. time-dependent or space varying data). The user-definable attribute files are listed and described in [section A.2](#).

The MDF-file has the following general characteristics:

- ◊ Each line contains a maximum of 300 characters.
- ◊ Each (set of) input parameter(s) is preceded by a *Keyword*.
- ◊ A *Keyword* is at most 6 characters long (a combination of numerical and alpha-numerical characters, but starting with an alpha-numeric character), followed by an equal sign “=”.

The MDF-file is an intermediate file between the FLOW-GUI and the hydrodynamic simulation program. As it is an ASCII-file, it can be transported to an arbitrary hardware platform. Consequently, the hydrodynamic simulation program and the FLOW-GUI do not necessarily have to reside on the same hardware platform.

Generally, you need not to bother about the internal layout or content of the MDF-file. It is, however, sometimes useful to be able to inspect the file and/or make small changes manually. Therefore the MDF-file is an ordinary ASCII-file which you can inspect and change with your favourite ASCII-editor.

The MDF-file is self contained, i.e. it contains all the necessary information about the model concerned. It can therefore be used as model archive by printing the file.

A.1.2 Example

In this section an example MDF-file is listed and described. The same area is used as in Chapter 5, but the scenario applies to a 3D computation. The left column contains the keyword and its value(s); the central column contains the dimension (if useful) and the right column contains a short description.

Record description:

Keyword and value	Format	Description
Ident = #Delft3D-FLOW .03.02 3.39.12#	C*28	Identification string FLOW-GUI
Commnt=		Comment line (not used)
Runtxt= #Demonstration model Delft3D #	C*20	Up to ten lines of free text to

continued on next page

continued from previous page

Keyword and value	Format	Description
#Friesian Tidal Inlet model #		clarify the purpose of the simulation
#3 layers #		and to indicate specific parameter selections.
Filcco= #fti_02.grd#	C*256	Name of the grid file <*.grd>
Fmtcco= #FR#	C*2	Format of grid file
Anglat= 5.500000e+001	1 R	Latitude of model centre
Grdang= 0.000000e+000	1 R	Angle between true North and y-axis
Filgrd= #fti_02.enc#	C*256	Grid enclosure file <*.enc>
Fmtgrd= #FR#	C*2	Format of grid enclosure file
MNKmax= 15 22 3	3 I	Number of grid points 3D simulation
Thick = 50.0000	1 R	Layer thickness from top to bottom
35.0000	1 R	in percentage of total water depth
15.0000	1 R	
Fildep= #fti_ch02.dep#	C*256	File with depth values <*.dep>
Fmtdep= #FR#	C*2	Format of depth file
Commnt=		
Fildry= #tut_fti_1.dry#	C*256	File with indices of dry points <*.dry>
Fmtdry= #FR#	C*2	Format of dry points file
Filtdd = #tut_fti_1.thd#	C*256	File with indices of thin dams <*.thd>
Fmttd = #FR#	C*2	Format of thin dam file
Commnt=		
Itdate= #1990-08-05#	C*10	Reference date for all time functions
Tunit = #M#	C*1	Time unit of time dependent data
Tstart= 0.000000	1 R	Start time after Itdate in Tunits
Tstop = 1.500000e+003	1 R	Stop time after Itdate in Tunits
Dt = 5.00000	1 R	Time step in Tunits
Tzone = 0	1 R	Local time zone
Commnt=		
Sub1 = #S W #	C*4	Flags to activate the processes in two
Sub2 = #PC #	C*3	groups of four and three characters
Namcl = #Conservative Spill #	C*20	Name of first constituents
Wnsvwp= #N#	C*1	Flag for space varying wind and pressure
Filwnd= #tut_fti_1.wnd#	C*256	File with wind data <*.wnd>
Fmtwnd= #FR#	C*2	Format of wind data file
Wndint= #Y#	C*1	Wind data interpolation flag
Commnt=		
Zeta0 = 1.90000	1 R	Initial condition water level
U0 = [.]	1 R	Initial condition x-velocity
V0 = [.]	1 R	Initial condition y-velocity
S0 = 3.000000e+001	1 R	Initial condition salinity, one value for each layer
C01 = 1.000000e+000	1 R	Initial concentration constituent 1
Commnt=		
Filbnd= #tut_fti_1.bnd#	C*256	File with boundary locations <*.bnd>
Fmtbnd= #FR#	C*2	Format of boundary file
Filbch= #tut_fti_1.bch#	C*256	File with harmonic flow boundary conditions <*.bch>
Fmtbch= #FR#	C*2	Format of harmonic conditions file
Filbcc= #tut_fti_1.bcc#	C*256	File with transport boundary conditions <*.bcc>
Fmtbcc= #FR#	C*2	Format of transport conditions file
Rettis= 1.000000e+002 1.000000e+002	x R	Thatcher-Harleman return time at surface (for x open boundaries)
Rettib= 1.000000e+002 1.000000e+002	x R	Thatcher-Harleman return time at bed level (for x open boundaries)

continued on next page

continued from previous page

Keyword and value	Format	Description
Commnt=		
Ag = 9.81300	1 R	Gravitational acceleration
Rhow = 1030.00	1 R	Density of water at background temperature and salinity
Alph0 = [.]		Reflection coefficient
Tempw = 15.0000	1 R	Background water temperature
Salw = 3.100000e+001	1 R	Background salinity
Rouwav= # #	C*4	Bottom stress formulation due to wave action
Wstres= 0.00250000 0.000000 0.00250000 100.000 0.00250000 0.000000	6 R	Wind stress and wind speed coefficients
Rhoa = 1.00000	1 R	Air density
Betac = 0.500000	1 R	Parameter spiral motion
Equili= #N#	C*1	Flag for computation spiral motion
Tkmod= #Algebraic #	C*12	Type of turbulence closure model
Ktemp = 0	1 I	Selection flag for heat model
Fclou = 0.000000	1 R	Percentage sky cloudiness
Sarea = 0.000000	1 R	Surface area in heat model
Ivapop = 0	1 I	Vapour pressure user specified. Only for heat model 4.
Temint= #Y#	C*1	Interpolation flag for temperature data
Commnt=		
Roumet= #C#	C*1	Type of bottom friction formulation
Ccofu = 45.0000	1 R	Uniform bottom roughness in u-dir.
Ccofv = 45.0000	1 R	Uniform bottom roughness in v-dir.
Xlo = 0.000000	1 R	Ozmidov length scale
Htur2d= #N#	C*2	Flag for HLES sub-grid model
Vicouv= 2.00000	1 R	Uniform horizontal eddy viscosity
Dicouv= 10.0000	1 R	Uniform horizontal eddy diffusivity
Vicoww= 1.00000e-006	1 R	Uniform vertical eddy viscosity
Dicoww= 1.00000e-006	1 R	Uniform vertical eddy diffusivity
Irov = 0	1 I	Flag to activate partial slip conditions
Z0v = [.]	1 R	Roughness length vertical side walls
Commnt=		
Iter = 2	1 I	Number of iterations in cont.eq.
Dryflp= #YES#	C*3	Flag for extra drying and flooding
Dpsopt= #MAX#	C*3	Option for check at water level points
Dpuopt= #MEAN#	C*4	Option for check at velocity points
Dryflc= 0.0500000	1 R	Threshold depth drying and flooding
Dco = -999.999	1 R	Marginal depth in shallow area's
Tlfsmo= 0.000000	1 R	Time interval to smooth the hydrodynamic boundary conditions
ThetQH= 0.0000000e+000	1 R	Relaxation parameter for QH relation (0 = no relaxation)
ThetQT= 0.0000000e+000	1 R	Relaxation parameter for total discharge boundary (0 = no relaxation)
Forfuv= #Y#	C*1	Flag horizontal Forester filter
Forfwv= #Y#	C*1	Flag vertical Forester filter
Sigcor= #N#	C*1	Flag to activate anti-creep
Trasol= #Cyclic-method#	C*13	Numerical method for advective terms
Momsol= #Cyclic#	C*6	Numerical method for momentum terms
Commnt=		
Filsrc= #tut_fti_1.src#	C*256	File with discharge locations <*.src>
Fmtsrc= #FR#	C*2	Format of discharge locations file
Fieldis= #tut_fti_1.dis#	C*256	File with discharge data <*.dis>
Fmtdis= #FR#	C*2	Format of discharge data file
Commnt= no. observation points: 5		

continued on next page

continued from previous page

Keyword and value	Format	Description
Filsta= #tut_fti_1.obs#	C*256	File with observation points <*.obs>
Fmtsta= #FR#	C*2	Format of observation points file
Filpar= #tut_fti_1.par#	C*256	File with drogues <*.par>
Fmtpar= #FR#	C*2	Format of drogues file
Commnt=		
Eps = [.]		
Commnt=		
Commnt= no. cross sections: 4		
Filcrs= #tut_fti_1.crs#	C*256	File with cross-sections <*.crs>
Fmtcrs= #FR#	C*2	Format of cross-sections file
Commnt=		
Filpla= #name.pla#	C*256	File with vegetation description <*.pla>
Commnt=		
PMhydr= #YYYYYYY#	C*6	Flags print map output hydrodynamic quantities
PMproc= #YYYYYYYYYYYY#	C*10	Flags print map output constituents and turbulence
PMderv= #YYY#	C*3	Flags print map output derived quantities
PHhydr= #YYYYYYY#	C*6	Flags print time history hydrodynamic quantities
PHproc= #YYYYYYYYYYYY#	C*10	Flags print time history constituents and turbulence
PHderv= #YYY#	C*3	Flags print time history derived quantities
PHflux= #YYYY#	C*4	Print flags time history fluxes through cross-sections
SMhydr= #YYYYYY#	C*5	Flags store map output hydrodynamic quantities
SMproc= #YYYYYYYYYYYY#	C*10	Flags store map output constituents and turbulence
SMderv= #YYYYYY#	C*6	Flags store map output derived quantities
SHhydr= #YYYY#	C*4	Flags store time history output hydrodynamic quantities
SHproc= #YYYYYYYYYYYY#	C*10	Flags store time history output constituents and turbulence
SHderv= #YYYYYY#	C*5	Flags store time history output derived quantities
SHflux= #YYYY#	C*4	Flags store time history output fluxes through cross-sections
Commnt= attribute file Fourier analysis		
Filfou= # #	C*256	File with quantities to be Fourier analysed <*.fou>
Online= #YES#	C*3	Flag for online visualisation
Prmap = [.]	1 R	Time instances to print map output
Prhis = 750.000 0.000000 60.0000	3 R	Time information to print history output
Flmap = 2190.00 120.000 2940.00	3 R	Time information to store map output
Flhis = 0.000000 5.000000 2940.00	3 R	Time information to store history output
Flpp = 2190.00 10.0000 2940.00	3 R	Time information to write to the communication file
Flrst = -999.999	1 R	Time interval to write restart file
Commnt=		
BarocP = #Y#	C*1	Flag for activating barocline pressure term at open boundaries. Default = #Y#

A.1.3 Physical parameters

For Tide Generating forces and Thatcher-Harleman return times there is an extra explanation on the use of the keywords in the following sections

A.1.3.1 Tide Generating Forces

For tide generating forces you can specify the tidal components that are taken into account.

The layout of `Tidfor` is defined as follows:

```
Tidfor = #M2 S2 N2 K2#
      #K1 O1 P1 Q1#
      #Mf Mm Ssa  #
```

A.1.3.2 Thatcher-Harleman Conditions

The Thatcher-Harleman conditions are by default specified by a return time (in minutes) for the surface and the bed level for each open boundary. Example with 3 open boundaries:

```
Rettis = 1.0000000e+002
        0.0000000e+000
        5.0000000e+001
Rettib = 1.0000000e+002
        0.0000000e+000
        5.0000000e+001
```

It is also possible to specify return times for individual constituents, using the following keywords:

<code>RetsS</code>	Return time for Salinity at surface
<code>RetbS</code>	Return time for Salinity at bed level
<code>RetsT</code>	Return time for Temperature at surface
<code>RetbT</code>	Return time for Temperature at bed level
<code>Rets01</code>	Return time for Constituent number 1 at surface
<code>Retb01</code>	Return time for Constituent number 1 at bed level

Idem for constituents with other numbers.

Remarks:



- ◊ `Rettis` and `Rettib` can be combined with the additional keywords to specify return times for all constituents, with varying values for specific constituents. Here is an example specifying a return time of 100 minutes for all constituents, except for temperature, having a return time of 200 minutes:

```
Rettis = 1.0000000e+002
        1.0000000e+002
Rettib = 1.0000000e+002
        1.0000000e+002
RetsT  = 2.0000000e+002
        2.0000000e+002
RetbT  = 2.0000000e+002
        2.0000000e+002
```

- ◊ `Rettis` and `Rettib` are supported by FLOW-GUI; the constituent specific variants are not; you have to add them with a text editor to the mdf-file.

A.1.4 Output options

The MDF-file contains a number of keywords for selecting output options to print (ascii) files and binary map-, his-, dro-, and com-files.

The keywords related to 2D/3D ascii output are `PMhydr`, `PMproc`, `PMderv` to select which quantities to write and `Prmap` to indicate the time steps at which to write these fields. The

keywords related to history station and cross-section ascii output are PHhydr, PHproc, PHderv, PHflux to select the quantities and Prhis to select the output times. The ascii output options are generally not used except for debugging purposes.

The keywords related to 2D/3D binary map-file output are SMhydr, SMproc, SMderv to select the quantities and Flmap to select the output times. The keywords related to history station and cross-section binary output are SHhydr, SHproc, SHderv, SHflux to select the quantities again and Flhis to select the times. The format used for the binary map-, his-, dro-files is by default NEFIS; this can be switched to NetCDF by means of the keyword FlNcdf. The keyword Flpp specifies the output times for the com-file; the quantities written to the com-file and the file format of the com-file cannot be changed.

The values to be specified for the PM*, PH*, SM* and SH* keywords are strings of Y and N characters representing flags for output of different quantities. These quantities differ slightly for storing results to file and for printing, so the flags for each option are described in the tables below. Finally [Table A.6](#) lists a couple of options for additional output to map- and his-files and the tri-diag file which can be switched on in the *Additional parameters* section of the user interface.

Table A.2: Print flags for map-data

Keyword and value	Number/Description	
PMhydr = #YYYYYY#	1 2–5 6	Water level U and V-velocities, magnitude and direction ω -velocities relative to σ -plane and w-velocities
PMproc = #YYYYYYYYYY#	1 2 3–7 8 9 and 10	Concentration salinity Temperature Concentration constituents Intensity spiral motion Turbulent energy and dissipation
PMderv=#YYY#	1 and 2 3	Vertical eddy viscosity and vertical eddy diffusivity. Richardson number if either or both are selected Density

Table A.3: Print flags for history-data

Keyword and value	Number/Description	
PHhydr = #YYYYYY#	1 2–5 6	Water level U- and V-velocities, magnitude and direction ω -velocities relative to σ -plane and w-velocities
PHproc = #YYYYYYYYYY#	1 2 3–7 8 9 and 10	Concentration salinity Temperature Concentration constituents Intensity spiral motion Turbulent energy and dissipation

Table A.3: Print flags for history-data

Keyword and value	Number/Description	
PHderv=#YYY#	1 and 2	Vertical eddy viscosity and vertical eddy diffusivity. Richardson number if either or both are selected
	3	Density
PHflux = #YYYYY#	1	Total flux through cross-sections
	2	Momentary flux through cross-sections
	3	Advection transport through cross-sections
	4	Dispersive transport through cross-sections

Table A.4: Storage flags for map-data

Keyword and value	Number/Description	
SMhydr = #YYYYYY#	1	Water level
	2 and 3	U- and V-velocities
	4 and 5	ω -velocities relative to σ -plane and w-velocities
SMproc = #YYYYYYYYYY#	1	Concentration salinity
	2	Temperature
	3–7	Concentration constituents
	8	Intensity spiral motion
	9 and 10	Turbulent energy and dissipation
SMderv=#YYYYYY#	1 and 2	U- and V- bed stress components
	3 and 4	Vertical eddy viscosity and vertical eddy diffusivity. Richardson number if either or both are selected
	5	Density
	6	Filtered U- and V-velocities of HLES model and horizontal eddy viscosity

Table A.5: Storage flags for history-data

Keyword and value	Number/Description	
SHhydr = #YYY#	1	water level
	2 and 3	U- and V-velocities
	4	W-velocities
SHproc = #YYYYYYYYYY#	1	Concentration salinity
	2	Temperature
	3–7	Concentration constituents
	8	Intensity spiral motion
	9	Turbulent kinetic energy
	10	Turbulent dissipation

Table A.5: Storage flags for history-data

Keyword and value	Number/Description	
SHderv=#YYYYY#	1 and 2 3 and 4 5	U- and V- bed stress components Vertical eddy viscosity and vertical eddy diffusivity. Richardson number if either or both are selected Density
SHflux=#YYYY#	1 2 3 4	Flux in U- and V-direction in ζ -point Total flux through cross-sections ¹⁾ Momentary flux through cross-sections ¹⁾ Advective flux through cross-sections ¹⁾ Dispersive flux through cross-sections ¹⁾

¹⁾ if cross-sections are defined.

The layout of the time information to store results, such as Prhis and Flmap, is defined as:

Item	Description
First real number	Start time in Tunit after the computation start time
Second real number	Time interval in Tunit
Third real number	Stop time in Tunit after the computation start time

Table A.6: Optional output flags under Additional parameters

Keyword	Value	Description	Default
AddTim	#Y# or #N#	Flag for additional performance timing output to <tri-diag> file.	#N#
AdvFlx	#Y# or #N#	Flag for output of instantaneous horizontal fluxes of the advection diffusion equation per substance. Note: anticreep and Forester filter fluxes are not included in these fluxes.	#N#
AirOut	#Y# or #N#	Write meteo input to map- and his-file. This can be used to check the meteorological model forcing. The following quantities will be written to the map-file: wind velocity vector, pressure, cloud coverage (if space varying), air humidity (if space varying), air temperature (if space varying), precipitation rate (if space varying), and evaporation rate (if computed). The following quantities will be written for the station locations to the his-file: wind velocity vector, pressure, precipitation rate, and evaporation rate.	#N#

continued on next page

Table A.6 – continued from previous page

Keyword	Value	Description	Default
CflMsg	#Y# or #N#	Write warnings for too high advective Courant numbers during whole simulation (instead of a maximum of 100 warnings).	#N#
Chezy	#Y# or #N#	Write roughness as Chézy values to map-file.	#N#
CumAFl	#Y# or #N#	Flag for output of cumulative horizontal fluxes of the advection diffusion equation per substance. Note 1: anticreep and Forester filter fluxes are not included in these fluxes. Note 2: AdvFlx will be automatically switched on if CumAFl is activated.	#N#
Filbal	#filename#	Triggers mass balance output for the polygons on the history-file: volume, area, average concentration and average bed level data as well as total water, constituent and sediment flux data. The polygon file should match the file format described in subsection A.2.25. The values will be accumulated over grid cells for which the centre point is located within the polygon. If the centre point is located in multiple polygons, the grid cell will be assigned to the first polygon only.	-
F1Ncdf	#string#	The value string lists the binary file types that should use NetCDF instead of NEFIS (or ascii in case of the Fourier file) as file format. The program will check whether the following substrings occur in the value string: map, his, dro, and fou for map-, history-, drogue-, and Fourier-files respectively.	-
HisBar	#Y# or #N#	Flag for output of barrier height data to history file	#N#
HisDis	#Y# or #N#	Flag for output of flow rate data of discharges and culverts to history file	#N# (#Y# in case of culverts)
HdtOut	#Y# or #N#	Flag for output at every half time step to investigate fluctuations in the results due to the two steps of the ADI solution. Both time steps will be labelled with the same (end of full time step) time.	#N#
HeaOut	#Y# or #N#	Flag for output of heat fluxes determined by temperature models.	#N#
continued on next page			

Table A.6 – continued from previous page

Keyword	Value	Description	Default
KfMnMx	#Y# or #N#	Flag for output of kfumin, kfumax, kfvmin, kf vmax, kfsmin, and kfsmax arrays to the map-file. These administrative arrays indicate which layers of a Z-model are active at velocity (U, V) and water level points.	#N#
LayOut	#Y# or #N#	Flag for output of vertical coordinates of layer interfaces to the map-file.	#N#
MapWav	#Y# or #N#	Write wave data to map-file. This can be used to check that the wave parameters have been passed correctly to the flow computation.	#N#
MergeMap	#Y# or #N#	Flag for merging the results of a parallel simulation when writing the map and Fourier files. If MergeMap equals #Y# the simulation will create one map file and one Fourier file for the whole domain (if output of these files is requested). If MergeMap equals #N# the simulation will create one map file and one Fourier file per partition (if output of these files is requested). Please note that a split (non-merged) map-file cannot be used for restarting.	#Y#
MomTrm	#Y# or #N#	Flag for output of momentum terms to the map-file (see details in Section A.1.4.1).	#N#
PrecOut	4 or 8	Precision of the data written to the output (history, map, drogue, Fourier) files: single (4) or double (8) precision. If the simulation runs in single precision, then PrecOut can only be set to 4.	4
Rough	#Y# or #N#	Write roughness in input unit (as specified Roumet keyword) to map-file. This is particularly useful when using the water depth dependent roughness formulations offered by trachytopes.	#N#
SgrThr	10^{-3} to 10^3	Threshold value in [m] for reporting water level change messages to <tri-diag> file.	25 m
SHlay	1 3 6	Specification of the layers for which the output parameters are written to history-file. By default all layers are written. This specification may be useful when output reduction is needed. Postprocessing (e.g. with Delft3D-QUICKPLOT) may be unable/awkward when necessary information is missing, (e.g. depth averaged velocity, vertical profiles).	-
continued on next page			

Table A.6 – continued from previous page

Keyword	Value	Description	Default
SMlay	2 5 7 8	Specification of the layers for which the output parameters are written to map-file. By default all layers are written. This specification may be useful when output reduction is needed. Postprocessing (e.g. with Delft3D-QUICKPLOT) may be unable/awkward when necessary information is missing, (e.g. depth averaged velocity, vertical profiles).	-
SMVelo	#glm# or #euler#	Flag for velocity output to his- and map-files.	#euler#
UgrThr	10^{-3} to 10^3	Threshold value in [m/s] for reporting velocity component change messages to <tri-diag> file.	5 m/s
Vortic	#Y# or #N#	Flag for output of vorticity and enstrophy.	#N#

A.1.4.1 Momentum terms output

The momentum Eqs. 9.7 and 9.8 in subsection 9.3.1 can be combined to the following vectorized form in which everything except for the partial velocity derivatives in time has been moved to the right hand side:

$$\begin{aligned}
 \underbrace{\begin{bmatrix} \frac{\partial u}{\partial t} \\ \frac{\partial v}{\partial t} \end{bmatrix}}_A = & \underbrace{\begin{bmatrix} -\frac{u}{\sqrt{G_{\xi\xi}}} \frac{\partial u}{\partial \xi} - \frac{uv}{\sqrt{G_{\xi\xi}} \sqrt{G_{\eta\eta}}} \frac{\partial \sqrt{G_{\xi\xi}}}{\partial \eta} \\ -\frac{v}{\sqrt{G_{\eta\eta}}} \frac{\partial v}{\partial \eta} - \frac{uv}{\sqrt{G_{\xi\xi}} \sqrt{G_{\eta\eta}}} \frac{\partial \sqrt{G_{\eta\eta}}}{\partial \xi} \end{bmatrix}}_B + \underbrace{\begin{bmatrix} -\frac{v}{\sqrt{G_{\eta\eta}}} \frac{\partial u}{\partial \eta} + \frac{v^2}{\sqrt{G_{\xi\xi}} \sqrt{G_{\eta\eta}}} \frac{\partial \sqrt{G_{\eta\eta}}}{\partial \xi} \\ -\frac{u}{\sqrt{G_{\xi\xi}}} \frac{\partial v}{\partial \xi} + \frac{u^2}{\sqrt{G_{\xi\xi}} \sqrt{G_{\eta\eta}}} \frac{\partial \sqrt{G_{\xi\xi}}}{\partial \eta} \end{bmatrix}}_C \\
 & + \underbrace{\begin{bmatrix} -\frac{\omega}{d+\zeta} \frac{\partial u}{\partial \sigma} \\ -\frac{\omega}{d+\zeta} \frac{\partial v}{\partial \sigma} \end{bmatrix}}_D + \underbrace{\begin{bmatrix} fv \\ -fu \end{bmatrix}}_E + \underbrace{\begin{bmatrix} -\frac{1}{\rho_0 \sqrt{G_{\xi\xi}}} P_\xi \\ -\frac{1}{\rho_0 \sqrt{G_{\eta\eta}}} P_\eta \end{bmatrix}}_F + \underbrace{\begin{bmatrix} F_\xi \\ F_\eta \end{bmatrix}}_G \\
 & + \underbrace{\begin{bmatrix} \frac{1}{(d+\zeta)^2} \frac{\partial}{\partial \sigma} (\nu_V \frac{\partial u}{\partial \sigma}) \\ \frac{1}{(d+\zeta)^2} \frac{\partial}{\partial \sigma} (\nu_V \frac{\partial v}{\partial \sigma}) \end{bmatrix}}_H + \underbrace{\begin{bmatrix} M_\xi \\ M_\eta \end{bmatrix}}_I \quad (A.1)
 \end{aligned}$$

When the keyword `MomTrm` is activated, then most of the marked terms will be written to the map-file. This feature is available as of Delft3D-FLOW version 6.01.01 for sigma-layer models only. The variables are added as elements to the map-series group. The field contains the terms valid for the last whole time step (i.e. averaged over the two half time steps of the ADI solver) for regular output, and terms valid for each half time step in case of half time step output (`HdtOut` active). The element names for the M- and N-components respectively, as well as a textual description of the meaning of the terms is given in the list below.

A elements MOM_DUDT and MOM_DVDT: acceleration (in GLM coordinates)

- B elements MOM_UDUDX and MOM_VDVDY: acceleration due to streamwise momentum transport
- C elements MOM_VDUDY and MOM_UDVDX: acceleration due to lateral momentum transport
- D vertical advection of momentum (not yet included in output file)
- E elements MOM_UCORIOLIS and MOM_VCORIOLIS: acceleration due to Coriolis force
- F the pressure term is split into acceleration due to barotropic pressure gradients as given by the first term in Eqs. 9.15 and 9.16 including the atmospheric pressure gradient (elements MOM_UPPRESSURE and MOM_VPRESSURE) and the baroclinic pressure gradient (elements MOM_UDENSITY and MOM_VDENSITY) as given by the second term in the aforementioned equations
- G elements MOM_UVISCO and MOM_VVISCO: acceleration due to viscosity as given by Eqs. 9.26 and 9.27
- H vertical diffusion of momentum (not yet included in the output file)
- I all other terms are further subdivided into flow resistance for instance due to vegetation (elements MOM_URESISTANCE and MOM_VRESISTANCE), tide generating forces (elements MOM_UTIDEGEN and MOM_VTIDEGEN), wind force in top layer (elements MOM_UWINDFORCE and MOM_VWINDFORCE), bed shear in bottom layer (elements MOM_UBEDSHEAR and MOM_VBEDSHEAR), and waves forces (elements MOM_UWAVES and MOM_VWAVES).

A.2 Attribute files

A.2.1 Introduction

In the following sections we describe the attribute files used in the input file (MDF-file) of Delft3D-FLOW. Most of these files contain the quantities that describe one specific item, such as the location of open boundaries, or time dependent data of fluxes discharged in the model area by discharge stations.

Most of the attribute files can be generated by the FLOW-GUI after defining an input scenario. Some files can almost only be generated by utility programs such as the curvilinear grid generated by RGFGRID. Still, we describe both type of files as it might be useful to know how the input data is structured to be able to generate (large) files, such as astronomic boundary conditions, or time-series for wind speed and direction by client specific tools.

For each file we give the following information:

- ◊ File contents.
- ◊ Filetype (free formatted, fix formatted or unformatted).
- ◊ Filename and extension.
- ◊ Generated by (i.e. how to generate the file).
- ◊ Restrictions on the file contents.
- ◊ Example(s).



Remarks:

- ◊ The access mode of all attribute files is sequential.
- ◊ In the examples the file content is printed in font Courier and comment (not included in the file) between curly brackets font, unless explicitly stated differently.

A.2.2 Orthogonal curvilinear grid

The orthogonal curvilinear grid file can be specified in the FLOW-GUI in Data group *Domain - Grid parameters*.

File contents	The co-ordinates of the orthogonal curvilinear grid at the depth points.
Filetype	ASCII
File format	Free formatted
Filename	<name.grd>
Generated	RGFGRID

Record description:

Record	Record description
	Preceding description records, starting with an asterisk (*), will be ignored.
1	Record with Co-ordinate System = Cartesian or value Spherical
2	Record with Missing Value = -9.999990000000000E+02. If this record is not given 0.0 will be assumed as missing value.
3	The number of grid <i>points</i> in m- and n-direction (2 integers).
4	Three real values (not used).
5 to K+5	A label and record number, the <i>x</i> -component of the world co-ordinates of all points in m-direction, starting with row 1 to row <i>nmax</i> , with as many continuation records as required by <i>mmax</i> and the number of co-ordinates per record. The label and record number are suppressed on the continuation lines. This set of records is repeated for each row until <i>n</i> = <i>nmax</i> .
K+5 to 2K+4	A similar set of records for the <i>y</i> -component of the world co-ordinates.

K is the number of records to specify for all grid points a set of *x*- and *y*-co-ordinates.

Restrictions:

- ◊ The grid must be orthogonal.
- ◊ Input items in a record are separated by one or more blanks.



Example:

```

*
* Deltares, Delft3D-RGFGRID Version 4.16.01.4531, Sep 30 2008, 23:32:27
* File creation date: 2008-10-01, 23:19:22
*
Coordinate System = Cartesian
      9       7
 0 0 0
Eta=    1    0.000000000000000E+00   1.000000000000000E+02   2.000000...
                  5.000000000000000E+02   6.000000000000000E+02   7.000000...

```

Eta=	2	0.000000000000000E+00	1.000000000000000E+02	2.000000...
		5.000000000000000E+02	6.000000000000000E+02	7.000000...
Eta=	3	0.000000000000000E+00	1.000000000000000E+02	2.000000...
		5.000000000000000E+02	6.000000000000000E+02	7.000000...
Eta=	4	0.000000000000000E+00	1.000000000000000E+02	2.000000...
		5.000000000000000E+02	6.000000000000000E+02	7.000000...
Eta=	5	0.000000000000000E+00	1.000000000000000E+02	2.000000...
		5.000000000000000E+02	6.000000000000000E+02	7.000000...
Eta=	6	0.000000000000000E+00	1.000000000000000E+02	2.000000...
		5.000000000000000E+02	6.000000000000000E+02	7.000000...
Eta=	7	0.000000000000000E+00	1.000000000000000E+02	2.000000...
		5.000000000000000E+02	6.000000000000000E+02	7.000000...
Eta=	1	1.000000000000000E+02	1.000000000000000E+02	1.000000...
		1.000000000000000E+02	1.000000000000000E+02	1.000000...
Eta=	2	2.000000000000000E+02	2.000000000000000E+02	2.000000...
		2.000000000000000E+02	2.000000000000000E+02	2.000000...
Eta=	3	3.000000000000000E+02	3.000000000000000E+02	3.000000...
		3.000000000000000E+02	3.000000000000000E+02	3.000000...
Eta=	4	4.000000000000000E+02	4.000000000000000E+02	4.000000...
		4.000000000000000E+02	4.000000000000000E+02	4.000000...
Eta=	5	5.000000000000000E+02	5.000000000000000E+02	5.000000...
		5.000000000000000E+02	5.000000000000000E+02	5.000000...
Eta=	6	6.000000000000000E+02	6.000000000000000E+02	6.000000...
		6.000000000000000E+02	6.000000000000000E+02	6.000000...
Eta=	7	7.000000000000000E+02	7.000000000000000E+02	7.000000...
		7.000000000000000E+02	7.000000000000000E+02	7.000000...

A.2.3 Computational grid enclosure

The computational grid enclosure file need to be specified in the FLOW-GUI in Data Group *Domain - Grid parameters*, the file itself is generated by RGFGRID, see [RGFGRID UM \(2016\)](#).

File contents	The indices of the external computational grid enclosure(s) and optionally one or more internal computational grid enclosures that outlines the active computational points in a Delft3D-FLOW computation. The file is strongly related to the curvilinear grid file.
Filetype	ASCII
File format	Free formatted
Filename	<name.enc>
Generated	RGFGRID

Record description:

Record	Record description
All	One pair of M and N indices representing the grid co-ordinates where a line segment of the computational grid enclosure (polygon) changes direction.



Restrictions:

- ◊ A polygon must be closed. The first point of the polygon is repeated as last point.
- ◊ A line segment may not intersect or touch any other line segment.
- ◊ The angle formed by consecutive line segments (measured counter clock-wise) can have a value of: 45, 90, 135, 225, 270 or 315 degrees, but not 0, 180 and 360 degrees.
- ◊ In a row or column there should be at least two active computational grid cells.
- ◊ Input items in a record are separated by one or more blanks.

Example:

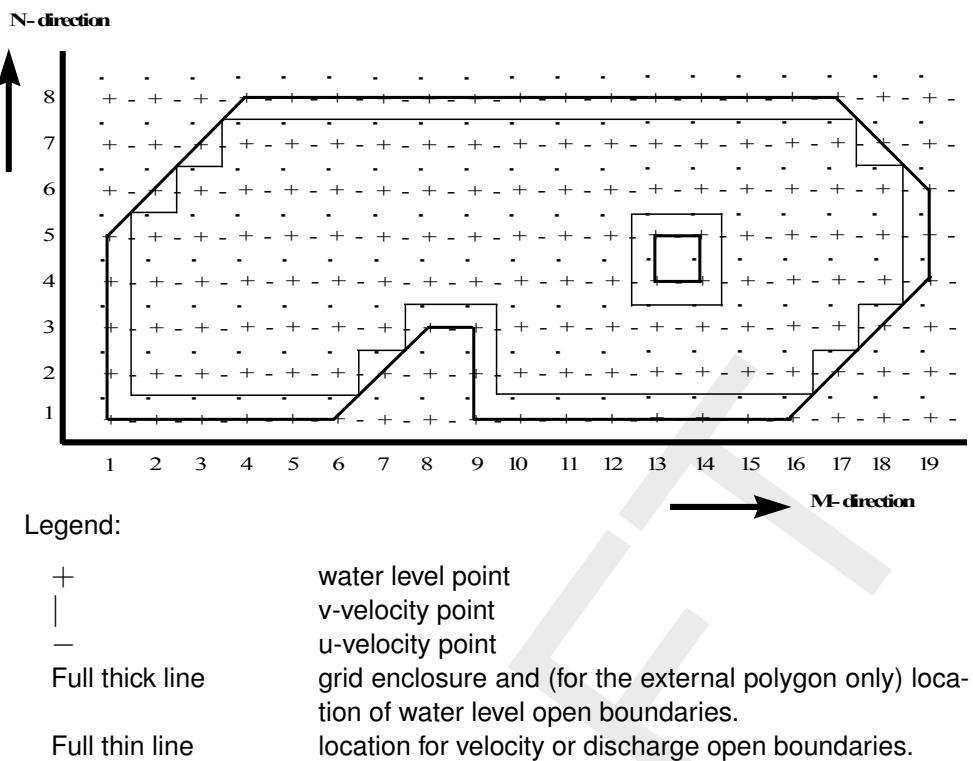


Figure A.1: Example of computational grid enclosures

Model area with (one) external and one internal polygon, see [Figure A.1](#).

```

1 1 begin external polygon
6 1
8 3
9 3
9 1
16 1
19 4
19 6
17 8
4 8
1 5
1 1 end external polygon
13 4 begin internal polygon
14 4
14 5
13 5
13 4 end internal polygon

```

A.2.4 Bathymetry

The bathymetry file can be specified in the FLOW-GUI in Data Group *Domain - Bathymetry*.

File contents	The bathymetry in the model area, represented by depth values (in metres) for all grid points.
Filetype	ASCII
File format	Free formatted or unformatted
Filename	<name.dep>
Generated	FLOW-GUI (only for uniform depth values). Offline with QUICKIN and data from digitised charts or GIS-database.

Record description:

Filetype	Record description
Free formatted	Depth values per row, starting at N = 1 to N = Nmax, separated by one or more blanks. The number of continuation lines is determined by the number of grid points per row (Mmax) and the maximum record size of 132.
Unformatted	Mmax depth values per row for N = 1 to N = Nmax.

**Restrictions:**

- ◊ The file contains one M and N line more than the grid dimension.
- ◊ The maximum record length in the free formatted file is 132.
- ◊ Depth values from the file will not be checked against their domain.
- ◊ The input items are separated by one or more blanks (free formatted file only).
- ◊ The default missing value is: -999.0

Example:

File containing 16 * 8 data values for a model area with 15 * 7 grid points (free formatted file).

```

1.0   2.0   3.0   4.0   -5.0   -5.0   -5.0   8.0   9.0   10.0   11.0
12.0  13.0  14.0  -5.0  -999.0
3.0   4.0   5.0   6.0   7.0   -6.0   -6.0   10.0  11.0  12.0  13.0
14.0  15.0  16.0  17.0  -999.0
5.0   6.0   7.0   8.0   9.0   10.0  -7.0   12.0  13.0  14.0  15.0
16.0  17.0  18.0  19.0  -999.0
7.0   8.0   9.0   10.0  11.0  12.0  13.0  14.0  15.0  16.0  17.0
18.0  19.0  -7.0  19.0  -999.0
9.0   10.0  11.0  12.0  13.0  14.0  15.0  16.0  17.0  18.0  19.0
20.0  19.0  18.0  17.0  -999.0
-7.0  12.0  13.0  14.0  15.0  16.0  17.0  18.0  19.0  20.0  19.0
18.0  17.0  16.0  15.0  -999.0
-8.0  -8.0  15.0  16.0  17.0  18.0  19.0  20.0  19.0  18.0  17.0
16.0  15.0  14.0  13.0  -999.0
-999.0 -999.0 -999.0 -999.0 -999.0 -999.0 -999.0 -999.0 -999.0 -999.0
-999.0 -999.0 -999.0 -999.0 -999.0

```

A.2.5 Thin dams

The thin dams file can be specified in the FLOW-GUI in Data Group *Domain - Thin dams*.

File contents	Location and type of thin dams.
Filetype	ASCII
File format	Free formatted
Filename	<name.thd>
Generated	QUICKIN or FLOW-GUI

Record description:

Record	Record description
each record	The grid indices of the begin and end point of a line of thin dams (4 integers). A character indicating the type of thin dams (U or V).

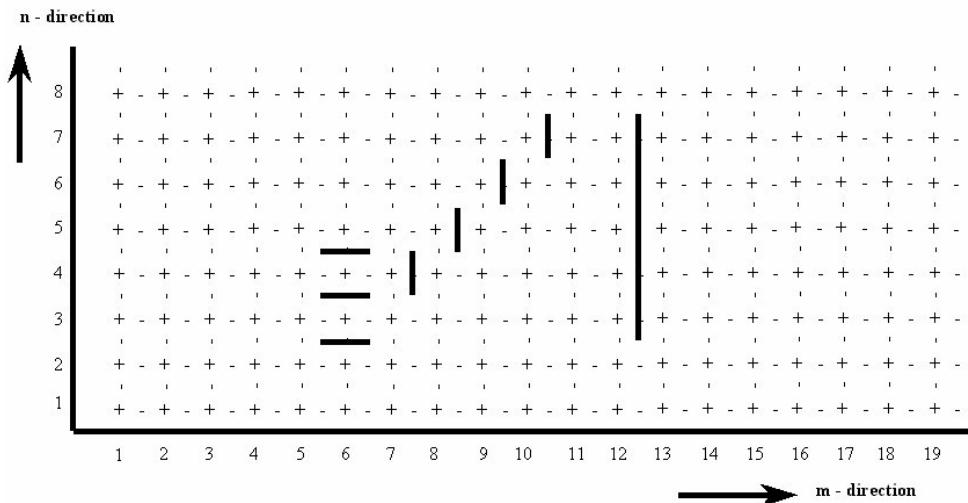


Figure A.2: Example of thin dams in a model area

Restrictions:

- ◊ The angle of the line segment and the horizontal numerical grid axis may be an integer multiple of 45 degrees.
- ◊ Thin dams can not be defined along the model boundaries (which by default lie along the lines $M = 1$, $N = 1$, $M = M_{max}$ or $N = N_{max}$). Therefore, the indices of thin dams must lie between $M = 2$ and $M_{max}-1$ and $N = 2$ and $N_{max}-1$ respectively.
- ◊ Input items are separated by one or more blanks.
- ◊ The direction of the dam is perpendicular to the velocity direction over which the dams are superimposed!



Example:

Three (sets of) thin dams in model area of $19 * 8$ grid points, see [Figure A.2](#).

```
6   2   6   4 V
7   4   10  7 U
12  3   12  7 U
```

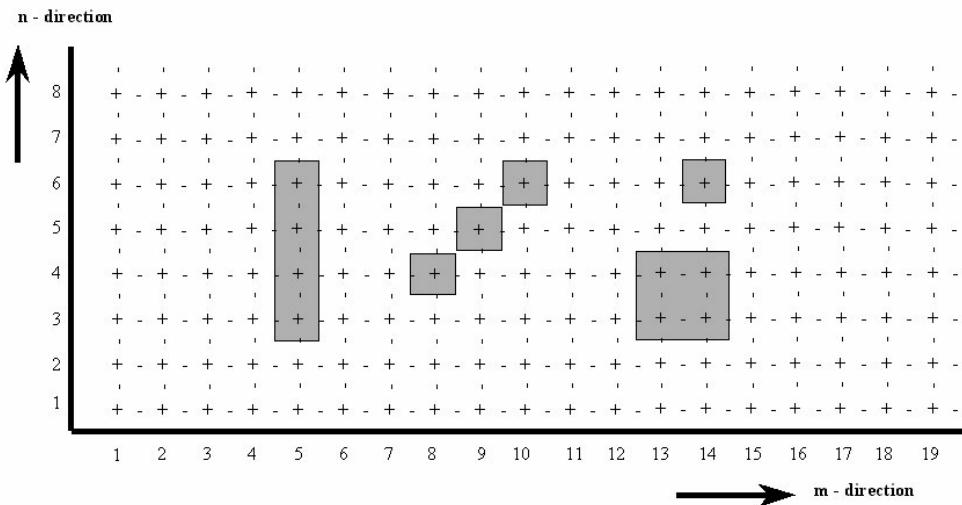
A.2.6 Dry points

The dry points file can be specified in the FLOW-GUI in Data Group *Domain - Dry points*.

File contents	Index location of (permanently) dry points.
Filetype	ASCII
File format	Free formatted
Filename	<name.dry>
Generated	QUICKIN or FLOW-GUI

Record description:

Record	Record description
each record	The grid indices of the begin and end point of a dry section (4 integers).

**Figure A.3:** Dry points in model area**Restrictions:**

- ◊ The angle of a line of dry points and the horizontal numerical grid axis can be an integer multiple of 45 degrees.
- ◊ Dry points may not be defined along the model boundaries (which by default lie along the lines $M = 1$, $N = 1$, $M = M_{max}$ or $N = N_{max}$). Therefore, the indices of these points must lie between $M = 2$ and $M_{max}-1$ and $N = 2$ and $N_{max}-1$, respectively.
- ◊ The input items are separated by one or more blanks.
- ◊ The most lower-left dry point has indices (2, 2).

Example:

Five sets of dry points in a model area of $19 * 8$ grid points, see [Figure A.3](#).

```

5   3   5   6
8   4   10  6
13  3   14  3
13  4   14  4
14  6   14  6
  
```

A.2.7 Time-series uniform wind

Time-series for wind speed and direction for a uniform wind can be specified in the FLOW-GUI in Data Group *Physical parameters - Wind*.

File contents

Time-series for wind speed and direction for a uniform wind. The wind direction is defined according to the nautical convention, i.e. relative to north and positive in clock-wise direction, see [Figure A.4](#).

Filetype

ASCII or binary

File format

Free formatted or unformatted

Filename

$<name.wnd>$

Generated

FLOW-GUI, WAVE-GUI, or manually offline

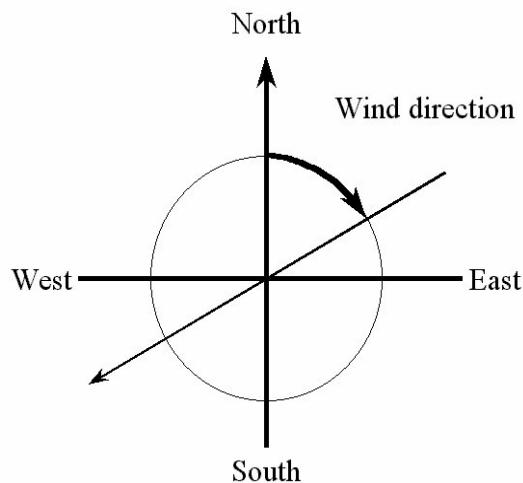


Figure A.4: Definition sketch of wind direction according to Nautical convention

Record description:

Record	Record description
each record	The time in minutes after the Reference date 00:00:00 hr (1 real). The wind speed in m/s and the wind direction relative to north, positive clockwise (2 reals).

Restrictions:

- ◊ Times must be an integer multiple of the simulation time step.
- ◊ The contents of the file will not be checked on its domain.
- ◊ The input items are separated by one or more blanks.



Example:

Time-series for uniform wind field, starting as a north wind, turning to south-west and back to north, see [Figure A.4](#). The wind direction in [Figure A.4](#) is about +60 degrees.

```

0.    0.0    0.
10.   1.2    315.
30.   3.7    270.
150.  4.2    225.
600.  5.7    225.
610.  4.0    235.
900.  0.0    0.

```

A.2.8 Space varying wind and pressure

In this section the different options are described for specifying space varying meteo data (wind, atmospheric pressure, etc.) as input for your simulation. The descriptions in this section are applicable to meteo input files of a certain version. More details on the compatibility of the meteo input files and the conversion from one version to another can be found in Appendix [B.7](#).

A.2.8.1 Defined on the computational grid

Time-series for space varying wind velocity components (east-west and south-north) and atmospheric pressure can be specified in the FLOW-GUI in Data Group *Physical parameters - Wind*.

File contents	Time-series for space varying wind velocity components (east-west and south-north) and atmospheric pressure, defined on the computational grid. The file consists of a header, followed by datablocks containing the wind and pressure fields at times specified using a standardised time definition above each datablock. The header specifies the type of file and the input it contains using a number of keywords. The keywords are case insensitive and the order of the keywords is not fixed.
Filetype	ASCII or binary.
File format	Free formatted or unformatted, keyword based.
Filename	<name.wnd>
Generated	Some offline program.

Header description:

Keywords	Value	Description
FileVersion	1.03	version of file format
Filetype	meteo_on_computational_grid	meteo input on computational grid
NODATA_value	free	value used for input that is to be neglected
n_quantity	3	number of quantities specified in the file
quantity1	x_wind	wind in <i>x</i> -direction
quantity2	y_wind	wind in <i>y</i> -direction
quantity3	air_pressure	air pressure
unit1	m s-1	unit of quantity1, meters/second
unit2	m s-1	unit of quantity2, meter/second
unit3	Pa or mbar	unit of quantity3, Pa or millibar

Time definition and data block description

Keywords	Value	Description
Time	<i>fixed format described below</i>	time definition string

The time definition string has a fixed format, used to completely determine the time at which a dataset is valid. The time definition string has the following format:

TIME *minutes/hours since YYYY-MM-DD HH:MM:SS TIME ZONE*, e.g.

360 minutes since 2008-07-28 10:55:00 +01:00

The format of the string is completely fixed. No extra spaces or tabs can be added between the different parts of the definition. The time definition is followed by the datablock of input values corresponding to the specified time. The data block consists of three subsequent blocks containing the velocity component in M-direction, the velocity component in N-direction and the atmospheric pressure, respectively. All three quantities are given for *Nmax* by *Mmax* points, where the first value in the dataset corresponds to cell (1, 1) on the grid. Every next line in the dataset then corresponds to a row on the grid. The time definition and the data block — for all three quantities — are repeated for each time instance of the time-series.

File version and conversion

The current description holds for FileVersion 1.03. The table below shows the latest modifications in the file format (and version number).

FileVersion	Modifications
1.03	No changes for this meteo input type, but for the meteo types <i>meteo_on_equidistant_grid</i> and <i>meteo_on_curvilinear_grid</i>
1.02	No changes for this meteo input type, but for the meteo type <i>meteo_on_spider_web_grid</i>
1.01	Changed keyword MeteoType to FileType Changed fixed value of input type (Keyword FileType) from <i>Svwp</i> to <i>meteo_on_computational_grid</i> (<i>meteo_on_flow_grid</i> is also allowed)

Restrictions:

- ◊ Keywords are followed by an equal sign '=' and the value of the keyword.
- ◊ When a keyword has value *free* the value of this keyword is free to choose by the user. When only one value is given for a keyword, this keyword has a fixed value and when 2 or more options are shown, the user can choose between those values.
- ◊ Times must be specified exactly according to the time definition. See the examples shown in this section.
- ◊ The contents of the file will not be checked on its domain.
- ◊ The wind components are specified at the cell centres (water level points) of the computational grid.
- ◊ Input items in a data block are separated by one or more blanks (free formatted file only).



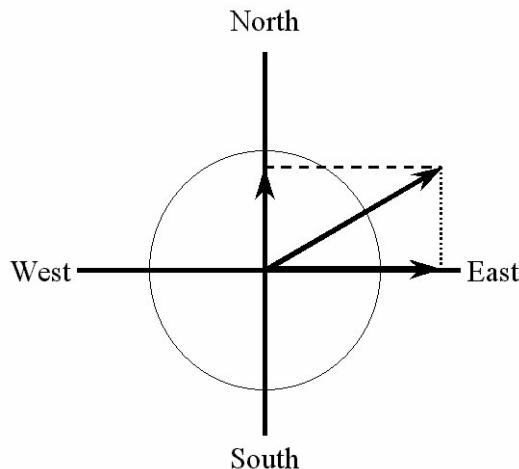


Figure A.5: Definition wind components for space varying wind

Remarks:

- ◊ The time definition in the meteorological file contains the number of minutes or hours since a reference date and time in a certain time zone. The reference time and time zone may differ from those of the simulation. The computational engine will search in the meteo file for the simulation time and interpolate between neighbouring times if necessary. Possible differences in time zone will be accounted for by shifting the meteo input data with the difference. The reference times within the time definition string may vary in a meteo file, i.e. it is possible to attach new input with a different reference time, behind the last data block.
- ◊ Comments can be added after #'s.

Example

Model area of $25 * 33$ grid points ($Mmax = 25$; $Nmax = 33$). The input data is printed in Courier, comments are printed behind #'s.

```

Time = 0.0 minutes since 2008-09-20 10:30:00 +01:00      # Time definition
{33 records with 25 values each}                          # Wind component west to east
{33 records with 25 values each}                          # Wind component south to north
{33 records with 25 values each}                          # Atmospheric pressure
Time = 340.0 minutes since 2008-09-20 10:30:00 +01:00    # Time definition
{33 records with 25 values each}                          # Wind component west to east
{33 records with 25 values each}                          # Wind component south to north
{33 records with 25 values each}                          # Atmospheric pressure
Time = 600.0 minutes since 2008-09-20 10:30:00 +01:00    # Time definition
{33 records with 25 values each}                          # Wind component west to east
{33 records with 25 values each}                          # Wind component south to north
{33 records with 25 values each}                          # Atmospheric pressure
Time = 1240.0 minutes since 2008-09-20 10:30:00 +01:00   # Time definition
{33 records with 25 values each}                          # Wind component west to east
{33 records with 25 values each}                          # Wind component south to north
{33 records with 25 values each}                          # Atmospheric pressure

```



Remarks:

- ◊ To obtain the wind direction according to the nautical convention, the wind direction is reversed.
- ◊ The wind is specified in terms of its components along the west-east (x_wind) and south-north (y_wind) co-ordinate system, see [Figure A.5](#). These definitions differ from the nautical convention (used for uniform wind), which is specified relative to true

North, see [Figure A.4](#).

Remark:

- ◊ On open boundaries, an input signal is prescribed that is consistent with some average pressure. Usually the signal corresponds to Mean Sea Level. One actually wants to prescribe an input signal corresponding to the local pressure prescribed by the space varying meteo input. To this end, it is possible to specify an average pressure - which should correspond to your input signal on the open boundaries - which is then used to determine local pressure gradients that need to be applied along the open boundaries to obtain an input signal that is consistent with the local atmospheric pressure. This functionality used to be specified in the wind file but it should now be specified in the Master Definition File in the Data Group *Additional parameters*, using `PavBnd: Average Pressure on Boundaries`. Using this keyword one can specify an average pressure that is used on all open boundaries, independent of the type of wind input. The pressure must be specified in N/m². An example:

```
PavBnd= 101300.0
```



A.2.8.2 Defined on an equidistant grid

Time-series for space varying wind velocity components (east-west and south-north) and atmospheric pressure on an equidistant grid (other than the computational grid) can be specified in the FLOW-GUI in Data Group *Additional parameters*.

File contents	Time-series of a space varying wind and atmospheric pressure defined on an equidistant rectilinear or spherical grid (other than the computational grid).
Filetype	ASCII.
File format	Free formatted, keyword based.
Filename	<name.amp> for atmospheric pressure in [Pa] or [millibar], <name.amu> for the wind speed component in east-west-direction in [m/s], <name.amv> for the wind speed component in north-south direction in [m/s].
Generated	Some offline program.



Remark:

- ◊ Space varying wind and pressure on an equidistant grid is implemented as a special feature. You must specify some additional keywords and values in Data Group *Additional parameters* or in the MDF-file; see [subsection B.7.1](#) for details.

A.2.8.3 Defined on a curvilinear grid

Time-series for space varying wind velocity components (east-west and south-north) and atmospheric pressure on a separate curvilinear grid (other than the computational grid) can be specified in the FLOW-GUI in Data Group *Additional parameters*.

File contents	Time-series of a space varying wind and atmospheric pressure defined on a curvilinear (Cartesian or spherical) grid (other than the computational grid).
Filetype	ASCII.
File format	Free formatted, keyword based.
Filename	<name.amp> for atmospheric pressure in [Pa] or [millibar], <name.amu> for the wind speed component in east-west-direction in [m/s],

Generated	$\langle name.amv \rangle$ for the wind speed component in north-south direction in [m/s], $\langle name.grd \rangle$ for the curvilinear grid on which the wind and pressure are specified.
	Some offline program.

**Remark:**

- ◊ Space varying wind and pressure on a curvilinear grid is implemented as a special feature. You must specify some additional keywords and values in Data Group *Additional parameters* or in the MDF-file; see [subsection B.7.2](#) for details.

A.2.8.4 Defined on a Spiderweb grid

Time-series for space varying wind velocity components (east-west and south-north) and atmospheric pressure on a Spiderweb grid can be specified in the FLOW-GUI in Data Group *Additional parameters*.

File contents	Time-series of a space varying wind and atmospheric pressure defined on a Spiderweb grid. The grid can be rectilinear or spherical. This type of wind input is used to describe (rotating) cyclone winds.
Filetype	ASCII.
File format	Free formatted, keyword based.
Filename	$\langle name.web \rangle$ containing the wind speed in [m/s], wind (from) direction in [degree] and atmospheric pressure in [Pa] or [millibar].
Generated	Some offline program.

**Remark:**

- ◊ Space varying wind and pressure on a Spiderweb grid is implemented as a special feature. You must specify some additional keywords and values in Data Group *Additional parameters* or in the MDF-file; see [subsection B.7.3](#) for details.

A.2.9 Initial conditions

File contents	Initial conditions for the hydrodynamics, the transported constituents (if any) and the secondary flow intensity (if any) at all points (Data Group <i>Initial conditions</i>).
File format	Free formatted or unformatted.
Filename	$\langle name.ini \rangle$
Generated	Offline by some external program.

Initial conditions for separate quantities can be generated with the program QUICKIN.

Record description:

Record	Record description
each record	A record contains a row of Mmax values (Mmax reals).

The file contains a matrix of dimensions (Mmax*Nmax) for each quantity and if relevant for each layer for which an initial condition is required.

The matrices are given in the following order:

- 1 Water elevation (one matrix).

- 2 U-velocities (Kmax matrices).
- 3 V-velocities (Kmax matrices).
- 4 Salinity, *only if selected as an active process* (Kmax matrices).
- 5 Temperature, *only if selected as an active process* (Kmax matrices).
- 6 Constituent number 1, 2, 3 to the last constituent chosen, *only if selected* (Kmax matrices).
- 7 Secondary flow (for 2D simulations only), *only if selected as an active process* (one matrix).

In total there will be $(C_{\text{max}}+2) \times K_{\text{max}} + 1$ matrices of $M_{\text{max}} \times N_{\text{max}}$ where:

- | | |
|------------------|---|
| M_{max} | number of grid points in M (U or ξ) direction. |
| N_{max} | number of grid points in N (V or η) direction. |
| K_{max} | number of layers. |
| C_{max} | number of constituents (including temperature, salinity, secondary flow). |

Restrictions:

- ◊ The maximum record length in the free formatted file is 132
- ◊ The contents of the file will not be checked on its domain.
- ◊ Input items in a record are separated by one or more blanks (free formatted file only).



Example 1:

A model of 1 layer, $M_{\text{max}} = 25$ and $N_{\text{max}} = 33$, with two constituents, TEST1 and TEST2, and secondary flow.

The input data is printed in Courier; comment (not in the file) is printed between brackets.

```
{33 records with 25 values each} {Water elevation}
{33 records with 25 values each} {U-velocity component}
{33 records with 25 values each} {V-velocity component}
{33 records with 25 values each} {Concentrations constituent TEST1}
{33 records with 25 values each} {Concentrations constituent TEST2}
{33 records with 25 values each} {Secondary flow intensity}
```

Example 2:

A model with 2 layers, $M_{\text{max}} = 25$ and $N_{\text{max}} = 33$, with salinity, temperature and one constituent, denoted by TEST.

The input data is printed in Courier; comment (not in the file) is printed between brackets.

```
{33 records with 25 values each} {Water elevation}
{33 records with 25 values each} {U-velocity component in layer 1}
{33 records with 25 values each} {U-velocity component in layer 2}
{33 records with 25 values each} {V-velocity component in layer 1}
{33 records with 25 values each} {V-velocity component in layer 2}
{33 records with 25 values each} {Salinity in layer 1}
{33 records with 25 values each} {Salinity in layer 2}
{33 records with 25 values each} {Temperature in layer 1}
{33 records with 25 values each} {Temperature in layer 2}
{33 records with 25 values each} {Concentrations constituent TEST in layer 1}
{33 records with 25 values each} {Concentrations constituent TEST in layer 2}
```



Remarks:

- ◊ A record for the free formatted file is to be interpreted as a logical record. The length of a physical record is limited to 132, so a logical record consists of as many physical records as required by M_{max} and the number of values per (physical) record.

- ◊ The velocity components are the computational U and V components, not the east and north components.

A.2.10 Open boundaries

The open boundaries file(s) can be specified in the FLOW-GUI in Data Group *Boundaries*.

File contents	The location and description of open boundaries.
Filetype	ASCII
File format	Fix formatted for text variables; free formatted for real and integer values.
Filename	< <i>name.bnd</i> >
Generated	FLOW-GUI

Record description:

Record	Record description																							
each record	<p>Name of the open boundary section (20 characters). Type of boundary (1 character).</p> <table style="margin-left: 40px;"> <tr><td>Z</td><td>water level</td></tr> <tr><td>C</td><td>current</td></tr> <tr><td>N</td><td>Neumann</td></tr> <tr><td>Q</td><td>discharge per grid cell</td></tr> <tr><td>T</td><td>total discharge for boundary section</td></tr> <tr><td>R</td><td>Riemann</td></tr> </table> <p>Type of data (1 character).</p> <table style="margin-left: 40px;"> <tr><td>A</td><td>astronomic</td></tr> <tr><td>H</td><td>harmonic</td></tr> <tr><td>Q</td><td>QH tables (only for water level boundaries)</td></tr> <tr><td>T</td><td>time-series</td></tr> </table> <p>Grid indices of the begin and end point of the boundary section (4 integers). Reflection coefficient (1 real), not for Neumann or Riemann. Vertical profile (three strings); <i>only for 3D simulations and velocity type boundaries (C, Q, T and R)</i>.</p> <table style="margin-left: 40px;"> <tr><td>Uniform</td></tr> <tr><td>Logarithmic</td></tr> <tr><td>3D profile</td></tr> </table> <p>Two labels (each 12 characters, no blanks in the label name) referencing to the blocks in the amplitude and phase file <*.bca>; <i>only if the type of data is A</i>.</p>	Z	water level	C	current	N	Neumann	Q	discharge per grid cell	T	total discharge for boundary section	R	Riemann	A	astronomic	H	harmonic	Q	QH tables (only for water level boundaries)	T	time-series	Uniform	Logarithmic	3D profile
Z	water level																							
C	current																							
N	Neumann																							
Q	discharge per grid cell																							
T	total discharge for boundary section																							
R	Riemann																							
A	astronomic																							
H	harmonic																							
Q	QH tables (only for water level boundaries)																							
T	time-series																							
Uniform																								
Logarithmic																								
3D profile																								



Restrictions:

- ◊ Maximum record length in the free formatted file is 132.
- ◊ The boundary section name must start at position one in a record.
- ◊ The value of the reflection coefficient will not be checked on its domain.
- ◊ All input items in a record must be separated by one or more blanks.

- ◊ Astronomic and harmonic forced boundaries must be specified before QH-relation forced boundaries, which in turn should be specified before time-series forced boundaries.
- ◊ Astronomic and harmonic forced boundaries cannot be combined.

Example:

Two boundary sections with data type A(stronomic) and one with type T(ime series).

```
Paradise Bay 1      Q A   1   1   1   5 0.0 Uniform    Paradise_1A  Paradise_1B
Paradise Bay 2      C A   16  3   16  6 0.0 Logarithmic Paradise_2A  Paradise_2B
Sea Entrance        Z T   4   8   14  8 0.0
```

Remarks:



- ◊ A label may not contain blanks between non-blank characters.
- ◊ For the labels 12 characters are read. Be sure the second label starts at least 13 positions after the start of the first.

A.2.11 Astronomic flow boundary conditions

File contents	Boundary conditions for open boundary sections of type <i>Astronomic</i> (Data Group <i>Boundaries - Flow conditions</i>) in terms of amplitudes and phases for the astronomic components.
Filetype	ASCII
File format	Fix formatted for text variables, free formatted for real and integer values.
Filename	<name.bca>
Generated	FLOW-GUI or offline by program Delft3D-TRIANA.

Record description:

Record	Record description
1	Label for end point A of open boundary section (12 characters, no blanks)
2 to 2+NCOM-1	For each component its name (8 characters), amplitude and phases (2 reals).
2+NCOM	Label for end point B of open boundary section (12 characters, no blanks).
2+NCOM+1 to 2+2*NCOM	For each component its name (8 characters), amplitude and phases (2 reals).

where:

NCOM number of tidal components.

These records are repeated for each open boundary section.

Restrictions:



- ◊ The labels for the end points of an open boundary section are defined in the <name.bnd> file.
- ◊ The name of the label may not contain blanks between non-blank characters.

- ◊ All component names are written in upper case.
- ◊ The label and component name must start in position one.
- ◊ The number of components and the components used may differ between boundary sections.
- ◊ At both ends of a section the same set of components must be defined. Between sections these sets may differ.

Example:

A model with 3 open boundary sections (with astronomical boundary conditions).

The input data is printed in Courier; comment (not in the file) is printed between brackets.

```

East_bound_A {section name, end point A}
A0 0.02 0.0 {mean value}
M2 1.87 314.3 {component name, amplitude and phase}
S2 0.32 276.4
O1 0.21 14.3

East_bound_B {section name, end point B}
A0 0.03 0.0 {mean value}
M2 1.89 264.7 {component name, amplitude and phase}
S2 0.29 220.9
O1 0.19 38.3

West_bound_A {section name, end point A}
A0 0.06 0.0 {mean value}
M2 1.71 122.5 {component name, amplitude and phase}
S2 0.18 46.4

West_bound_B {section name, end point B}
A0 0.06 0.0 {mean value}
M2 1.69 110.3 {component name, amplitude and phase}
S2 0.19 22.4

Sea_bound_A {section name, end point A}
A0 0.07 0.0 {mean value}
M2 1.67 300.9 {component name, amplitude and phase}
S2 0.32 76.2
K1 0.05 33.1

Sea_bound_B {section name, end point B}
A0 0.07 0.0 {mean value}
M2 1.69 324.1 {component name, amplitude and phase}
S2 0.29 110.1
K1 0.09 6.1

```

A.2.12 Astronomic correction factors

File contents	The file contains <i>corrections</i> to the astronomical components of the open boundary sections with data type astronomic. These corrections may be applied during calibration. This file avoids a large processing effort of your tidal data. Just specify the appropriate boundary section and the component(s) that you wish to alter and how much it needs be changed (Data Group <i>Boundaries - Flow conditions</i>). The amplitude factor is a multiplicative factor and the phase factor is an additive factor.
Filetype	ASCII
File format	Fix formatted for text variables, free formatted for real and integer values.
Filename	<name.cor>
Generated	FLOW-GUI

Record description:

Record	Record description
1	Label for begin point (12 characters).
2 to 1+NCOMR	Component name (8 characters), amplitude and phase correction factors (2 reals).
2+NCOMR	Label for end point (12 characters).
3+NCOMR to 2+2*NCOMR	Component name (8 characters), amplitude and phase correction factors (2 reals).

where:

NCOMR the number of tidal components that requires corrections.

The records (1 to 2+2*NCOMR) may be repeated for the number of open boundary sections.

Restrictions:

- ◊ The name of the label may not contain blanks between non-blank characters.
- ◊ Label names for the begin and end points are defined in the <name.bnd> file.
- ◊ All component names are written in upper case.
- ◊ The label and component name must start in record position one.
- ◊ The number of components and the components used may differ per boundary section.
- ◊ The astronomical component A0 cannot be corrected.

Example:

Model for which two open boundary sections, with astronomical boundary conditions, need corrections.

The input data is printed in Courier; comment (not in the file) is printed between brackets.

```

East_bound_A {section name, end point A}
M2 0.90 10.0 {component name, amplitude and phase}
S2 0.95 -7.5
East_bound_B {section name, end point B}
M2 0.90 10.0 {component name, amplitude and phase}
S2 0.95 -7.5
Sea_bound_A {section name, end point A}
M2 0.95 7.0 {component name, amplitude and phase}
S2 0.90 -3.0
Q1 1.10 15.0
K1 1.10 10.0
Sea_bound_B {section name, end point B}
M2 0.95 7.0 {component name, amplitude and phase}
S2 0.90 -3.0
Q1 1.10 15.0
K1 1.10 10.0

```

Remark:

- ◊ In the example the correction factors are the same for both end points; but this is not mandatory.

A.2.13 Harmonic flow boundary conditions

File contents	The frequencies, amplitudes and phases for all open boundary sections with data type equal to H(armonic) (Data Group <i>Boundaries - Flow conditions</i>).
Filetype	ASCII
File format	Free formatted
Filename	<name.bch>
Generated	FLOW-GUI or manually offline

Record description:

Record	Record description
1	Frequencies (including 0.0 for the mean value) (reals).
2	Blank
3 to NTOH+2	Amplitudes at the begin of each boundary section for all frequencies (reals).
NTOH+3 to 2*NTOH+2	Amplitudes at the end of each boundary section for all frequencies (reals).
2*NTOH+3	Blank
2*NTOH+4 to 3*NTOH+3	Phases at the begin of each boundary section for all frequencies (blanks for mean value) (reals).
3*NTOH+4 to 4*NTOH+3	Phases at the end of each boundary section for all frequencies (blanks for mean value) (reals).

where:

NTOH number of open boundary sections driven with harmonic frequencies.



Restrictions:

- ◊ Maximum record length is 132.
- ◊ The input items will not be checked on their domain.
- ◊ Input items in a record are separated by one or more blanks.



Remark:

- ◊ The phase values at intermediate points are interpolated from the values specified at the begin and end of the opening section. You should take care for a good representation of phases at transition points (e.g. an interpolation between 356 and 13 degrees).

Example:

Model area with 3 open boundary sections (NTOH = 3) with H data type and two harmonic frequencies including the mean value (which has frequency 0.0).

The input data is printed in Courier; comment (not in the file) is printed between brackets.

```

0.0    30.0000 {KC = 2, including zero frequency for the mean value}
{blank record}
-1.6   2.1   {amplitudes at end A, section 1}
-1.5   2.3   {amplitudes at end A, section 2}
1.6    2.2   {amplitudes at end A, section 3}
-1.7   2.2   {amplitudes at end B, section 1}
-1.7   2.1   {amplitudes at end B, section 2}
1.8    2.2   {amplitudes at end B, section 3}
{blank record}
120.5  {phase at end A, 1-st component, section 1}
79.6   {phase at end A, 1-st component, section 2}
245.8  {phase at end A, 1-st component, section 3}
125.6  {phase at end B, 1-st component, section 1}
88.9   {phase at end B, 1-st component, section 2}
283.7  {phase at end B, 1-st component, section 3}

```

A.2.14 QH-relation flow boundary conditions

File contents	QH relations at water level boundaries for boundary sections with the data type Q (Data Group <i>Boundaries - Flow conditions</i>).
File format	Fix format for header information; free format for time series data.
Filename	<name.bcq>
Generated	FLOW-GUI or manually offline

Record description:

For each water level boundary segment with data type Q a data block must be prescribed consisting of:

- ◊ Header records containing a number of compulsory and optional keywords accompanied by their values.
- ◊ A set of records containing the discharge/water level data.
- ◊ Each record contains a discharge in [m³/s] and a water level in [m].

Restrictions:



- ◊ Maximum record length is 5000.
- ◊ Position, format of the keywords and the format of keyword-values in the header are fixed (see example).
- ◊ All keywords have a length of 20 characters.
- ◊ Header in each block must be ended with the (compulsory) keyword: 'records in table' accompanied by the number of data records to follow.
- ◊ Positive discharges indicate flow in positive M/N direction. If the model flows in negative M/N direction negative discharges must be specified.
- ◊ Discharges must be specified in increasing order. That is, $Q = 100.0$ should be specified before $Q = 200.0$. For negative discharges $Q = -200.0$ must be specified before $Q = -100.0$.
- ◊ QH boundaries should only be specified as outflow boundaries.
- ◊ The sequence of blocks must be consistent with the sequence of water level boundary sections with data type Q.

Example:

Model with open boundary sections with data type Q. Flow in negative M or N direction.

table-name	Boundary Section :	2	a20,1x,a40
contents	uniform		a20,1x,a20
location ¹	open boundary number	2	a20,1x,a20
xy-function	'equidistant'		
interpolation ²	linear		a20,1x,free
parameter	'total discharge (t)' unit '[m**3/s]'		a20,1x,a20,1x,a20,free
parameter	'water elevation (z)' unit '[m]'		a20,1x,a20,1x,a20,free
records in table	5		a20,1x,i6
-2000.0000	2.1455		
-1500.0000	1.7711		
-1000.0000	1.3516		
-500.0000	0.8514		
-300.0000	0.6057		

A.2.15 Time-series flow boundary conditions

File contents	Time-series for flow boundary conditions of all open boundary sections with data type T (Data Group <i>Boundaries - Flow conditions</i>).
Filetype	ASCII
File format	Fix format for header information; free format for time-series data.
Filename	<name.bct>
Generated	FLOW-GUI, program Delft3D-NESTHD or manually offline

Record description:

For each open boundary segment with boundary data of type T (time-series) the data is given in two related blocks:

- 1 A header block containing a number of compulsory and optional keywords accompanied by their values.
- 2 A data block containing the time dependent data.

Description header block:

¹sequence must follow sequential order of open boundaries in pre-processor

²Optional Keywords and values

Record	Text	Format	Value	Format
1	table-name	a20, 1x	'Boundary Section: ##'	a
2	contents	a20,1x	'Logarithmic' 'Uniform' '3d-profile'	a
3	location	a20, 1x	'name boundary section' (see <name.bnd> file)	a
4	time-function	a20, 1x	'non-equidistant'	a
5	reference-time	a20, 1x	yyyymmdd (must be equal to it- date)	i8
6	time-unit	a20, 1x	'minutes'	a
7	interpolation	a20, 1x	'linear'	a
8 to 8+NPAR+1	parameter	a20, 1x	'parameter name and location', 'layer and location', units '[ccc]'	a, 'units', a
10+NPAR	records-in-table	a20, 1x	number of records in the data block	integer

Remark:

- ◊ NPAR is the number of parameters for which a time varying boundary condition is being specified.



Description data block:

Record	Record description
each record	Time in minutes after the Reference Date and NPAR*{2 of Kmax} values representing the parameter for which a time varying boundary condition is being specified (all reals). The number of values to be specified for each end point of a boundary section depends on the type of profile in the vertical: 2 real values for either a uniform or logarithmic profile Kmax*2 real values for 3D-profile where the end points for each layer are prescribed.

Restrictions:

- ◊ Maximum record length is 5000.
- ◊ Position, format of the keywords and the format of keyword-values in the header are fixed (see example).
- ◊ All keywords have a length of 20 characters.
- ◊ Header in each block must be ended with the (compulsory) keyword: 'records in table' accompanied by the number of data records to follow.
- ◊ Times must be multiples of the integration time step; the times specified will be checked on their domain.
- ◊ The sequence of blocks must be consistent with the sequence of open boundary secti-



ons with data type T(ime).

- ◊ All open boundaries that have '3D-profile' must precede other open boundaries.

Example 1:

Model with 2 open boundary sections with time-series as boundary conditions.

The first boundary section concerns a discharge boundary for which the boundary condition is given at two time breakpoints, i.e. at 0.0 and 8 000.0 minutes after the Reference Date. The vertical profile is logarithmic, the interpolation method linear, the time-series is assumed to be non-equidistant and the time is given in minutes. The second boundary section concerns a current boundary for which the boundary condition is given at two time breakpoints, i.e. at 0.0 and 8 000.0 minutes after the Reference Date. The vertical profile is uniform, the interpolation method linear, the time series is assumed to be non-equidistant and the time is given in minutes.

```

table-name      'Boundary Section : 1'
contents        '
location        'East Boundary'
time-function   'non-equidistant'
reference-time  19941001
time-unit       'minutes'
interpolation   'linear'
parameter       'time           , unit , , '
parameter       'flux/discharge (q) end A' unit , '
parameter       'flux/discharge (q) end B' unit , '
records in table 2
      0.0000 50000.0    100000.
      8000.0000 75000.0    133000.

table-name      'Boundary Section : 2'
contents        '
location        'West boundary'
time-function   'non-equidistant'
reference-time  19941001
time-unit       'minutes'
interpolation   'linear'
parameter       'time           , unit '[min]'
parameter       'current        (c) end A' unit '[m/s]'
parameter       'current        (c) end B' unit '[m/s]'
records in table 2
      0.0000 1.50000     1.60000
      8000.0000 1.75000     1.80000

```

Example 2:

The second example concerns an open sea boundary consisting of three sections, one section is velocity controlled, one section is Riemann controlled and one section is water level controlled. The computation is depth averaged. See [subsection A.2.17](#) for the corresponding bcc-file.

```

table-name      'Boundary Section : 1'
contents        '
location        'west-vel-1'
time-function   'non-equidistant'
reference-time  19960718
time-unit       'minutes'
interpolation   'linear'
parameter       'time           , unit '[min]'
parameter       'current        (c) end A' unit '[m/s]'
parameter       'current        (c) end B' unit '[m/s]'

```

```

records-in-table      185
 840.0  0.000  0.000
 855.0  0.120  0.116
  .....
  .....
 3555.0 -0.651 -0.845
 3570.0 -0.589 -0.783
table-name          'Boundary Section : 2'
contents            'Uniform           '
location            'west-vel-2        '
time-function      'non-equidistant'
reference-time    19960718
time-unit          'minutes'
interpolation     'linear'
parameter          'time             , unit '[min]'
parameter          'riemann          (r) end A' unit '[m/s]'
parameter          'riemann          (r) end B' unit '[m/s]'
records-in-table   185
 840.0  0.000  0.000
 855.0  0.113  0.104
  .....
 3555.0 -0.815 -0.341
 3570.0 -0.756 -0.301
table-name          'Boundary Section : 3'
contents            'Uniform           '
location            'west-wl           '
time-function      'non-equidistant'
reference-time    19960718
time-unit          'minutes'
interpolation     'linear'
parameter          'time             , unit '[min]'
parameter          'water elevation (z) end A' unit '[m]'
parameter          'water elevation (z) end B' unit '[m]'
records-in-table   185
 840.0  1.899  1.858
 855.0  1.848  1.808
  .....
 3585.0  2.197  2.192
 3600.0  2.186  2.183

```

Example 3:

The third example concerns the same open boundary of the second example, but now for a 3D computation with 5 layers in the vertical. See [subsection A.2.17](#) for the bcc-file.

```

table-name          'Nested flow BC west-vel-1           '
contents            '3d-profile          '
location            'west-vel-1          '
time-function      'non-equidistant'
reference-time    19960718
time-unit          'minutes'
interpolation     'linear'
parameter          'time             , unit '[min]'
parameter          'current          (c) end A layer: 1' unit '[m/s]'
parameter          'current          (c) end A layer: 2' unit '[m/s]'
parameter          'current          (c) end A layer: 3' unit '[m/s]'
parameter          'current          (c) end A layer: 4' unit '[m/s]'
parameter          'current          (c) end A layer: 5' unit '[m/s]'
parameter          'current          (c) end B layer: 1' unit '[m/s]'
parameter          'current          (c) end B layer: 2' unit '[m/s]'
parameter          'current          (c) end B layer: 3' unit '[m/s]'
parameter          'current          (c) end B layer: 4' unit '[m/s]'
parameter          'current          (c) end B layer: 5' unit '[m/s]'
records-in-table   185

```

```

840.0 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000
855.0 .124 .118 .109 .109 .106 .126 .119 .115 .112 .107
.....
3585.0 -.583 -.641 -.736 -.788 -.776 -.444 -.529 -.651 -.766 -.841
3600.0 -.499 -.549 -.636 -.691 -.682 -.374 -.447 -.557 -.668 -.741
table-name      'Nested flow BC west-vel-2'
contents        ,
location        'west-vel-2'
time-function   'non-equidistant'
reference-time  19960718
time-unit       'minutes'
interpolation   'linear'
parameter       'time'          ' unit '[min]'
parameter       'current'       (c) end A layer: 1' unit '[m/s]'
parameter       'current'       (c) end A layer: 2' unit '[m/s]'
parameter       'current'       (c) end A layer: 3' unit '[m/s]'
parameter       'current'       (c) end A layer: 4' unit '[m/s]'
parameter       'current'       (c) end A layer: 5' unit '[m/s]'
parameter       'current'       (c) end B layer: 1' unit '[m/s]'
parameter       'current'       (c) end B layer: 2' unit '[m/s]'
parameter       'current'       (c) end B layer: 3' unit '[m/s]'
parameter       'current'       (c) end B layer: 4' unit '[m/s]'
parameter       'current'       (c) end B layer: 5' unit '[m/s]'
records-in-table 185
840.0 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000
855.0 .119 .113 .109 .106 .103 .104 .104 .101 .999 .102
.....
3585.0 -.403 -.491 -.612 -.728 -.804 -.450 -.454 -.454 -.422 -.365
3600.0 -.339 -.415 -.524 -.634 -.705 -.409 -.407 -.390 -.349 -.286
table-name      'Nested flow BC west-wl'
contents        ,
location        'west-wl'
time-function   'non-equidistant'
reference-time  19960718
time-unit       'minutes'
interpolation   'linear'
parameter       'time'          ' unit '[min]'
parameter       'water elevation (z)' end A ' unit '[m]'
parameter       'Water elevation (z)' end B ' unit '[m]'
records-in-table 185
840.0 1.899 1.856
855.0 1.883 1.857
.....
3585.0 2.186 2.181
3600.0 2.185 2.177

```

A.2.16 Time-series correction of flow boundary conditions

File contents	Time-series corrections for flow boundary conditions of all open boundary sections with data type T (Data Group <i>Boundaries - Flow conditions</i> in GUI).
Filetype	ASCII
File format	Fix format for header information; free format for time-series data.
	Same format as .bct file described above.
Filename	<name.bc0>
MDF file example line	Filbc0 = # name.bc0 #
Generated	program Delft3D-NESTHD or manually offline

Record description:

For each open boundary segment with boundary data of type T (time-series) the data is given in two related blocks:

- 1 A header block containing a number of compulsory and optional keywords accompanied by their values.
- 2 A data block containing the time dependent data.

Description header block:

Record	Text	Format	Value	Format
1	table-name	a20, 1x	'Boundary Section: ##'	a
2	contents	a20,1x	'Logarithmic' 'Uniform' '3d-profile'	a
3	location	a20, 1x	'name boundary section' (see <name.bnd> file)	a
4	time-function	a20, 1x	'non-equidistant'	a
5	reference-time	a20, 1x	yyyymmdd (must be equal to it- date)	i8
6	time-unit	a20, 1x	'minutes'	a
7	interpolation	a20, 1x	'linear'	a
8 to 8+NPAR+1	parameter	a20, 1x	'parameter name and location', 'layer and location', units '[ccc]'	a, 'units', a
10+NPAR	records-in-table	a20, 1x	number of records in the data block	integer

Remark:

- ◊ NPAR is the number of parameters for which a time varying boundary condition is being specified.



Description data block:

Record	Record description
each record	Time in minutes after the Reference Date and NPAR*{2 of Kmax} values representing the parameter for which a time varying boundary condition is being specified (all reals). The number of values to be specified for each end point of a boundary section depends on the type of profile in the vertical: 2 real values for either a uniform or logarithmic profile Kmax*2 real values for 3D-profile where the end points for each layer are prescribed.

**Restrictions:**

- ◊ Maximum record length is 5000.
- ◊ Position, format of the keywords and the format of keyword-values in the header are fixed (see example).
- ◊ All keywords have a length of 20 characters. The keywords are **case-sensitive**.
- ◊ Header in each block must be ended with the (compulsory) keyword: 'records in table' accompanied by the number of data records to follow.
- ◊ Times must be multiples of the integration time step; the times specified will be checked on their domain.
- ◊ The sequence of blocks must be consistent with the sequence of open boundary sections with data type T(ime).
- ◊ All open boundaries that have '3D-profile' must precede other open boundaries.

Example 1:

One of the open boundary sections with time-varying boundary correction in addition to the boundary condition.

A water level correction at the east boundary for which the boundary condition is given at two time breakpoints, i.e. at 40.0 and 50.0 minutes after the Reference Date. The vertical profile is logarithmic, the interpolation method linear, the time-series is assumed to be non-equidistant and the time is given in minutes.

```

table-name      'Boundary Section : 1'
contents        ''
location        'East Boundary'
time-function   'non-equidistant'
reference-time  19941001
time-unit       'minutes'
interpolation   'linear'
parameter       'time'           ,      unit  ,  ,
parameter       'water elevation (z)' end A' unit '[m]' 
parameter       'water elevation (z)' end B' unit '[m]' 
records in table 2
 4.0000000e+001 -9.0207780e-005 1.0587220e-004
 5.0000000e+001 -1.9945080e-004 7.0667030e-005
...

```

A.2.17 Time-series transport boundary conditions

File contents	Time-series for transport boundary conditions of all open boundary sections (Data Group <i>Boundaries - Transport conditions</i>).
Filetype	ASCII
File format	Fix format for header information; free format for time-series data
Filename	<name.bcc>
Generated	FLOW-GUI, program Delft3D-NESTHD or manually offline

Record description:

For each open boundary segment the data is given in two related blocks:

- 1 A header block containing a number of compulsory and optional keywords accompanied by their values.
- 2 A data block containing the time dependent data.

Description header block:

Record	Text	Format	Value	Format
1	table-name	a20, 1x	'Boundary Section : ##'	a
2	contents	a20,1x	'Logarithmic' 'Uniform' 'Step' '3d-profile'	a
3	location	a20, 1x	'name boundary section' (see <name.bnd> file)	a
4	time-function	a20, 1x	'non-equidistant'	a
5	reference-time	a20, 1x	yyyymmdd (must be equal to it- date)	i8
6	time-unit	a20, 1x	'minutes'	a
7	interpolation	a20, 1x	'linear'	a
8 to 8+NPAR+1	parameter	a20, 1x	'parameter name and location', 'layer and location', units '[ccc]'	a, 'units', a
10+NPAR	records in table	a20, 1x	number of records in the data block	integer

Remark:

- ◊ NPAR is the number of parameters for which a time-varying boundary condition is being specified.



Description data block:

Record	Record description
each record	Time in minutes after the Reference Date and NPAR values representing the parameter for which a time varying boundary condition is being specified (all reals). The number of values to be specified for each end point of a boundary section depends on the type of profile in the vertical: uniform: 2 real values for each parameter specified linear: 4 real values for each parameter specifying the parameter in the surface and in the bottom layer for both section end points. step: 5 real values; 4 reals for each parameter specifying the parameter in the surface and in the bottom layer for both section end points and the location of the discontinuity/jump in the vertical in metres below the surface 3D-profile: Kmax*2 real values for each layer in both end points of the boundary section being prescribed.

**Restrictions:**

- ◊ Maximum record length is 5000.
- ◊ Position, format of the keywords and the format of keyword-values in the header are fixed (format in example).
- ◊ All keywords have a length of 20 characters.
- ◊ Header in each block must be ended with the (compulsory) keyword: 'records in table' accompanied by the number of data records to follow.
- ◊ Times must be multiples of the integration time step; the times specified will be checked on their domain.
- ◊ Not all values of other parameters from the file will be checked on their domain.
- ◊ The sequence of blocks must be consistent with the sequence of open boundary sections and the sequence of constituents.
- ◊ All open boundaries that have '3D-profile' must precede other open boundaries.

Example 1:

Model area with two open boundary sections. A 3D simulation ($KMAX > 1$) has been specified; the profiles in the vertical are prescribed as functions. Salinity and one constituent are included in the simulation. The boundary condition is specified as follows:

- ◊ For a boundary section called West BOUNDARY, the salinity has a 'Step' profile with the discontinuity occurring at 2.5 metres below the surface and a constituent called 'test constituent' has a Uniform profile in the vertical. The boundary conditions are given for two time breakpoints, i.e. at 0.0 and 8000.0 minutes after the Reference Date 00:00:00 hr.
- ◊ For a boundary section called East BOUNDARY, the salinity has a linear profile in the vertical and a constituent called 'test constituent' has a Uniform profile in the vertical. The boundary conditions are given for two time breakpoints, i.e. at 0.0 and 8000.0 minutes after midnight on the Reference Date.

```

table-name      'T-series BC for process run: 123'
contents        'step'
location        'West BOUNDARY'
time-function   'non-equidistant'
reference-time  19941001
time-unit       'minutes'
parameter       'time'                                ' unit '[min]'
parameter       'salinity'                            ' end A surface' unit '[ppt]'
parameter       'salinity'                            ' end A bed    ' unit '[ppt]'
parameter       'salinity'                            ' end B surface' unit '[ppt]'
```

```

parameter          'salinity'           end B bed      ' unit '[ppt]'
parameter          'discontinuity'       ' unit '[m]'
records in table  2
    0.0000   33.0000   35.000   34.0000   36.0000   2.5000
    8000.0000  32.7000   34.000   35.0000   36.2000   2.5000
table-name        'T-series BC for flow run: 123'
contents          'uniform'
location          'West BOUNDARY'
time-function     'non-equidistant'
reference-time   19941001
time-unit         'minutes'
interpolation    'linear'
parameter          'time'               ' unit '[min]'
parameter          'test constituent'  end A          ' unit '[ - ]'
parameter          'test constituent'  end B          ' unit '[ - ]'
records in table 2
    0.0000   1.50000   1.60000
    8000.0000  1.75000   1.80000
table-name        'T-series BC for process run: 123'
contents          'linear'
location          'East BOUNDARY'
time-function     'non-equidistant'
reference-time   19941001
time-unit         'minutes'
interpolation    'linear'
parameter          'time'               ' unit '[min]'
parameter          'salinity'          end A surface ' unit '[ppt]'
parameter          'salinity'          end A bed   ' unit '[ppt]'
parameter          'salinity'          end B surface ' unit '[ppt]'
parameter          'salinity'          end B bed   ' unit '[ppt]'
records in table 2
    0.0000   33.0000   35.000   34.0000   36.0000
    8000.0000  32.7000   34.000   35.0000   36.2000
table-name        'T-series BC for flow run: 123'
contents          'uniform'
location          'East BOUNDARY'
time-function     'non-equidistant'
reference-time   19941001
time-unit         'minutes'
interpolation    'linear'
parameter          'time'               ' unit '[min]'
parameter          'test constituent'  end A          ' unit '[ - ]'
parameter          'test constituent'  end B          ' unit '[ - ]'
records in table 2
    0.0000   1.50000   1.60000
    8000.0000  1.75000   1.80000

```

Example 2:

The second example concerns an open sea boundary consisting of three sections for which the salinity concentrations are prescribed. The computation is depth averaged; see [subsection A.2.15](#) for the corresponding bct-file.

```

table-name        'Boundary Section : 1'
contents          'Uniform '
location          'west-vel-1 '
time-function    'non-equidistant'
reference-time   19960718
time-unit         'minutes'
interpolation    'linear'
parameter          'time'               ' unit '[min]'
parameter          'Salinity'          end A uniform ' unit '[ppt]'
parameter          'Salinity'          end B uniform ' unit '[ppt]'
records-in-table 185

```

```

840.0 20.9 23.2
855.0 20.9 23.2
870.0 20.9 23.0
885.0 20.9 22.8
900.0 20.9 22.5
.....
3555.0 20.2 21.2
3570.0 20.5 21.2
3585.0 20.7 21.3
3600.0 20.9 21.4

table-name      'Boundary Section : 2'
contents        'Uniform '
location        'west-vel-2          '
time-function   'non-equidistant'
reference-time  19960718
time-unit       'minutes'
interpolation   'linear'
parameter       'time           unit '[min]'
parameter       'Salinity        unit '[ppt]'
parameter       'Salinity        unit '[ppt]'
records-in-table 185
840.0 23.2 21.9
855.0 23.1 21.9
870.0 23.0 21.8
885.0 22.8 21.6
900.0 22.5 21.4
.....
3555.0 21.1 17.8
3570.0 21.1 17.8
3585.0 21.2 17.8
3600.0 21.3 17.9

table-name      'Boundary Section : 3'
contents        'Uniform '
location        'west-vel-3          '
time-function   'non-equidistant'
reference-time  19960718
time-unit       'minutes'
interpolation   'linear'
parameter       'time           unit '[min]'
parameter       'Salinity        unit '[ppt]'
parameter       'Salinity        unit '[ppt]'
records-in-table 185
840.0 20.6 22.2
855.0 20.6 22.1
870.0 20.5 21.8
885.0 20.4 21.4
900.0 20.3 20.8
.....
3555.0 16.7 12.4
3570.0 16.7 13.0
3585.0 16.7 13.5
3600.0 16.8 13.9

```

Example 3:

The third example concerns the same open boundary of the second example, i.e. consisting of three sections for which the salinity concentrations are prescribed. The computation is a 3D computation with 5 layers in the vertical; see [subsection A.2.15](#) for the corresponding bct-file.

```

table-name      'Nested transport bc salinity west-vel-1          '
contents        '3d-profile' # at ends A&B of open boundary segment
location        'west-vel-1          '
time-function   'non-equidistant'
reference-time  19960718

```

```

time-unit          'minutes'
interpolation     'linear'
parameter          'time'                                ' unit '[min]'
parameter          'salinity'   end A layer: 1' unit '[ppt]'
parameter          'salinity'   end A layer: 2' unit '[ppt]'
parameter          'salinity'   end A layer: 3' unit '[ppt]'
parameter          'salinity'   end A layer: 4' unit '[ppt]'
parameter          'salinity'   end A layer: 5' unit '[ppt]'
parameter          'salinity'   end B layer: 1' unit '[ppt]'
parameter          'salinity'   end B layer: 2' unit '[ppt]'
parameter          'salinity'   end B layer: 3' unit '[ppt]'
parameter          'salinity'   end B layer: 4' unit '[ppt]'
parameter          'salinity'   end B layer: 5' unit '[ppt]'
records-in-table  185
 840.0 15.6 16.5 19.3 19.3 20.3 17.1 18.4 21.1 23.0 24.1
 855.0 15.6 16.7 19.4 19.5 20.6 16.9 17.9 20.6 22.6 23.8
  .....
 3585.0 11.7 13.1 15.2 16.7 17.8 10.7 13.1 18.1 19.3 20.6
 3600.0 11.8 13.1 15.2 16.7 17.6 11.1 13.1 18.0 19.1 20.4
table-name        'Nested transport bc salinity west-vel-2'
contents          '3d-profile' # at ends A&B of open boundary segment
location          'west-vel-2'
time-function     'non-equidistant'
reference-time    19960718
time-unit          'minutes'
interpolation     'linear'
parameter          'time'                                ' unit '[min]'
parameter          'salinity'   end A layer: 1' unit '[ppt]'
parameter          'salinity'   end A layer: 2' unit '[ppt]'
parameter          'salinity'   end A layer: 3' unit '[ppt]'
parameter          'salinity'   end A layer: 4' unit '[ppt]'
parameter          'salinity'   end A layer: 5' unit '[ppt]'
parameter          'salinity'   end B layer: 1' unit '[ppt]'
parameter          'salinity'   end B layer: 2' unit '[ppt]'
parameter          'salinity'   end B layer: 3' unit '[ppt]'
parameter          'salinity'   end B layer: 4' unit '[ppt]'
parameter          'salinity'   end B layer: 5' unit '[ppt]'
records-in-table  185
 840.0 17.3 18.4 20.6 22.9 24.2 16.5 16.5 17.9 20.2 23.3
 855.0 17.2 18.0 20.2 22.6 23.9 16.4 16.5 17.8 20.0 23.1
  .....
 3585.0 10.7 13.1 18.3 19.5 20.8 11.2 12.3 15.0 17.4 19.2
 3600.0 11.1 13.2 18.2 19.4 20.7 11.9 12.8 15.4 17.8 19.5
  .....

```

A.2.18 Time-series for the heat model parameters

File contents	Time dependent data for heat model (Data Group <i>Physical parameters - Heat flux model</i>).
Filetype	ASCII
File format	Free formatted
Filename	<name.tem>
Generated	FLOW-GUI, or manually offline

Record description:

The record description depends on the heat flux model selected.

Option 1: Absolute temperature model, total incoming solar radiation

Record	Record description
each record	Time after midnight on the Reference Date [minutes]. Relative humidity [percentage]. Air temperature [$^{\circ}$ C]. Incoming solar radiation for a cloudless sky [J/m 2 s] [4 reals].

Option 2: Absolute temperature model, net solar radiation

Record	Record description
each record	Time after midnight on the Reference Date [minutes]. Relative humidity [percentage]. Air temperature [$^{\circ}$ C]. Net (sum of short and long wave radiation, reflection accounted for) solar radiation [J/m 2 s], [4 reals].

Option 3: Excess temperature model

Record	Record description
each record	Time after midnight on the Reference Date [minutes]. Background temperature [$^{\circ}$ C], [2 reals].

Option 4A: Murakami heat flux model

In this option the vapour pressure will be computed ($\text{IVAPOP} = 0$).

Record	Record description
each record	Time after midnight on the Reference Date [minutes]. Relative humidity [percentage]. Air temperature [$^{\circ}$ C]. Net (short wave) solar radiation [J/m 2 s], [4 reals].

Option 4B: Murakami heat flux model

In this option the vapour pressure is user-defined ($\text{IVAPOP} = 1$).

Record	Record description
each record	Time after midnight on the Reference Date [minutes]. Relative humidity [percentage]. Air temperature [$^{\circ}$ C]. Net (short wave) solar radiation [J/m ² s]. Vapour pressure [mbar], [5 reals].

Option 5A: Ocean model

In this option the solar radiation will be computed from the cloud coverage (`SolRad = #N#`)

Record	Record description
each record	Time after midnight on the Reference Date [minutes]. Relative humidity [percentage]. Air temperature [$^{\circ}$ C]. Fraction cloud coverage [percentage], [4 reals].

Option 5B: Ocean model

In this option the solar radiation is specified directly, additional to the cloud coverage (`SolRad = #Y#`)

Record	Record description
each record	Time after midnight on the Reference Date [minutes]. Relative humidity [percentage]. Air temperature [$^{\circ}$ C]. Fraction cloud coverage [percentage] Net (short wave) solar radiation [J/m ² s], [5 reals].

Remark:

- ◊ The parameter `IVAPOP` is specified in the MDF-file.



Restrictions:

- ◊ Times must be multiples of the integration time step.
- ◊ Relative humidity must be a percentage (0–100).
- ◊ The other parameters will not be checked against their domain.
- ◊ Input items in a record must be separated by one or more blanks.
- ◊ Only for the Ocean model: Using `SolRad` as described for option 5B, one can specify the measured net solar radiation directly as a time series. The specified solar radiation is then only reduced using the Albedo coefficient.



Example 1:

Heat flux model option 3 is selected.

```
0.      0.  
10.     15.  
20.     17.
```

```

30.    17.
150.   16.
600.   16.
610.   15.
900.    0.

```

Example 2:

Heat flux model option 2 is selected.

```

0.    40.    18.    0.
10.   45.    21.    50.
20.   47.    24.   200.
30.   47.    22.   200.
150.  46.    25.   250.
600.  46.    23.   200.
610.  45.    18.    50.
900.  40.    17.    0.

```

Example 3:

Heat flux model option 4B is selected. You have to add `IVAPOP = 1` in the Data Group *Additional parameters*.

```

0.    40.0   18.0   0.    8.3412
10.   45.0   21.0   50.   11.2994
20.   47.0   24.0   200.  14.1575
30.   47.0   22.0   200.  12.5449
150.  46.0   25.0   250.  14.7110
600.  46.0   23.0   200.  13.0460
610.  45.0   18.0   50.   9.3838
900.  40.0   17.0   0.    7.8337

```

A.2.19 Bottom roughness coefficients

File contents	Bottom roughness coefficients in U- and V-direction for all grid points (Data Group <i>Physical parameters - Roughness</i>).
Filetype	ASCII
File format	Free formatted or unformatted
Filename	<code><name.rgh></code>
Generated	Free formatted: FLOW-GUI (only for uniform depth values), QUICKIN or manually offline. Unformatted: by some external program.

Record description:

Two blocks with data are needed: one for U- and one for V-direction.

Filetype	Record description
Free formatted	The bottom roughness is given per row, starting at N = 1 to N = Nmax, separated by one or more blanks. The number of continuation lines is determined by the number of grid points per row (Mmax) and the maximum record size of 132.
Unformatted	Mmax bottom roughness values per row for N = 1 to N = Nmax.

Restrictions:

- ◊ The maximum record length in the free formatted file is 132.
- ◊ The value of these parameters will not be checked against their domain.
- ◊ Input items in a record must be separated by one or more blanks (free formatted file only).

**Example:**

Bottom roughness coefficients (Chézy formulation) for a model containing 11*7 points.

The records are filled with (maximal) 8 reals.

The input data is printed in Courier; comment (not in the file) is printed between brackets.

```
65.0 65.0 65.0 65.0 65.0 65.0 65.0 {start U-roughness coefficient}
65.0 65.0 65.0
65.0 65.0 65.0 65.0 65.0 65.0 65.0
65.0 65.0 65.0
60.0 60.0 60.0 60.0 60.0 60.0 60.0
60.0 60.0 60.0
60.0 60.0 60.0 60.0 60.0 60.0 60.0
60.0 60.0 60.0
55.0 55.0 55.0 55.0 55.0 55.0 55.0
55.0 55.0 55.0
55.0 55.0 55.0 55.0 55.0 55.0 55.0
55.0 55.0 55.0
55.0 55.0 55.0 55.0 55.0 55.0 55.0
55.0 55.0 55.0
60.0 60.0 60.0 60.0 60.0 60.0 60.0 {start V-roughness coefficient}
60.0 60.0 60.0
60.0 60.0 60.0 60.0 60.0 60.0 60.0
60.0 60.0 60.0
65.0 65.0 65.0 65.0 65.0 65.0 65.0
65.0 65.0 65.0
65.0 65.0 65.0 65.0 65.0 65.0 65.0
65.0 65.0 65.0
65.0 65.0 65.0 65.0 65.0 65.0 65.0
65.0 65.0 65.0
65.0 65.0 65.0 65.0 65.0 65.0 65.0
65.0 65.0 65.0
65.0 65.0 65.0 65.0 65.0 65.0 65.0
65.0 65.0 65.0
```

A.2.20 Horizontal eddy viscosity and diffusivity

File contents	Horizontal eddy viscosity and eddy diffusivity for all grid points (Data Group <i>Physical parameters - Viscosity</i>).
Filetype	ASCII or binary
File format	Free formatted or unformatted
Filename	< <i>name.edy</i> >
Generated	Free formatted: FLOW-GUI, QUICKIN or manually offline. Unformatted: by some external program.

Remark:

- ◊ Horizontal eddy diffusivity is only needed in case of salinity, temperature or constituents.

Record description:

Filetype	Record description
Free formatted	The eddy viscosity is given per row, starting at N = 1 to N = Nmax, separated by one or more blanks. The number of continuation lines is determined by the number of grid points per row (Mmax) and the maximum record size of 132. The eddy diffusivity is given per row in a similar fashion as the eddy viscosity is given.
Unformatted	Mmax eddy viscosity values per row for N = 1 to N = Nmax. First the eddy viscosity is given followed by the eddy diffusivity.

**Restrictions:**

- ◊ The maximum record length in the free formatted file is 132.
- ◊ The value of the input parameters will not be checked against their domain.
- ◊ Items in a record must be separated by one or more blanks (free formatted file only).

Example:

Eddy viscosity and eddy diffusivity for a model with (one layer and) m=8, n=12 grid points. A constituent is included in the model.

The input data is printed in Courier; comment (not in the file) is printed between brackets.

```

1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 {start eddy viscosity}
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 {start eddy diffusivity}
10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
10.0 10.0 25.0 25.0 25.0 10.0 10.0 10.0
10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
10.0 10.0 25.0 25.0 25.0 10.0 10.0 10.0
10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
10.0 10.0 25.0 25.0 25.0 10.0 10.0 10.0
10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0

```

A.2.21 Discharge locations

File contents	Description (name, grid indices and interpolation method for discharge rate) for discharge stations (Data Group <i>Operations - Discharges</i>).
Filetype	ASCII
File format	Fix formatted for text variables, free formatted for real and integer values.
Filename	<name.src>
Generated	FLOW-GUI or manually offline

Record description:

Record	Record description																
each record	<p>Name of discharge (20 characters), Type of interpolation (1 character):</p> <table> <tr> <td>y</td> <td>linear,</td> </tr> <tr> <td>n</td> <td>block,</td> </tr> </table> <p>Grid indices of discharge location (3 integers), Character to indicate the type of discharge:</p> <table> <tr> <td>blank n</td> <td>normal discharge,</td> </tr> <tr> <td>b</td> <td>bubble screen, see section B.22,</td> </tr> <tr> <td>c d e u</td> <td>culvert, see subsubsection 10.9.2.6,</td> </tr> <tr> <td>m</td> <td>discharge with momentum,</td> </tr> <tr> <td>p</td> <td>power station,</td> </tr> <tr> <td>w</td> <td>walking discharge,</td> </tr> </table> <p>Grid indices of power station outlet (3 integers).</p>	y	linear,	n	block,	blank n	normal discharge,	b	bubble screen, see section B.22 ,	c d e u	culvert, see subsubsection 10.9.2.6 ,	m	discharge with momentum,	p	power station,	w	walking discharge,
y	linear,																
n	block,																
blank n	normal discharge,																
b	bubble screen, see section B.22 ,																
c d e u	culvert, see subsubsection 10.9.2.6 ,																
m	discharge with momentum,																
p	power station,																
w	walking discharge,																

Restrictions:

- ◊ One record per discharge
- ◊ Maximum record length is 132.
- ◊ The discharge name is read as string of length 20 and must start in position one.
- ◊ If K = 0, the discharge will be distributed over the vertical proportional to the relative layer thickness.
- ◊ A discharge must be located inside the computational grid enclosure.
- ◊ Items in a record must be separated by one or more blanks.



Example:

The application requires 5 discharges. The first two discharges are 'normal' discharges, the first discharge is only in layer 5, the second discharge is distributed over the vertical. The third discharge is a walking discharge; the release point moves with drying and flooding. The last two discharges are so-called power stations. The intake of the first power station is distributed over the vertical and its outlet is in layer 1, while the second power station has its intake in layer 5 and its outlet distributed over the vertical.

```

Discharge Station A Y 10 11 5
Discharge Station B Y 15 5 0
Discharge Station C Y 114 88 3 w
Discharge Station D N 115 93 0 p 110 90 1
Discharge Station E Y 88 119 5 p 80 115 0

```

A.2.22 Flow rate and concentrations at discharges

File contents	Discharge rate and concentrations as a function of time for each discharge location (Data Group <i>Operations - Discharges</i>).
Filetype	ASCII
File format	Fix formatted for text variables, free formatted for real and integer values.
Filename	<name.dis>
Generated	FLOW-GUI, or manually offline

Record description:

For each discharge the data is given in two related blocks:

- 1 A header block containing a number of compulsory and optional keywords with their values.
- 2 A data block containing the time dependent data.

Description header block:

Record	Text	Value
1	table-name	'T-series for discharges run: runid'
2	contents	regular, momentum, walking, power
3	location	name of the discharge
4	time-function	equidistant or non-equidistant
5	reference-time	yyyymmdd
6	time-unit	minutes
7	interpolation	linear or block
8 to 8+NPAR+1	parameter	name of parameter, 'unit', units of parameter
10+NPAR	records in table	the number of records in the data block

where:

NPAR is the number of parameters for which a time discharge is being specified.

Description data block:

Record	Record description
each record	Time in minutes after 00:00:00 on the Reference Date (1 real), NPAR values representing the parameters for which a time varying discharge is being specified (all reals).



Restrictions:

- ◊ The parameters must be given in the following mandatory sequence: discharge flux, salinity, temperature, constituents, velocity and direction relative to the ξ -axis (positive counter-clock wise). Only those constituents are specified in the time dependent data that are specified in the header.
- ◊ The maximum record length in the file is 132.
- ◊ The format of keywords and keyword-values in the header are fixed.
- ◊ All keywords have a length of 20 characters.
- ◊ Header records must start in position one.
- ◊ Header in each block must be ended with the (compulsory) keyword: 'records in table' accompanied by its appropriate value.
- ◊ Times must be a multiple of the integration time step; times will be checked.
- ◊ Not all other parameters will be checked against their domain.
- ◊ The order of the blocks must be consistent with the sequence of discharge locations as specified in the FLOW-GUI.
- ◊ Input items in the data records must be separated by one or more blanks.

Example:

The number of discharges is 3. Each discharge releases salinity and one constituent in addition to the water released.

```

          'T-series for discharges run: 1'
contents
location
time-function
reference-time
19941001
time-unit
'minutes'
interpolation
'linear'
parameter
'flux/discharge rate' unit '[m**3/s]'
parameter
'salinity'           unit '[ppt]'
parameter
'test'              unit '[kg/m**3]'
parameter
'flow magnitude [m/s]' unit '[m/s]'
parameter
'flow direction [deg]' unit '[deg]'
records in table
2
    0.0  90.0  5.0  0.0  45.0  18.0
    1440.0 90.0  0.0  0.0  50.0  20.0
          'T-series for discharges run: 123'
contents
location
time-function
reference-time
19941001
time-unit
'minutes'
interpolation
'linear'
parameter
'flux/discharge rate' unit '[m**3/s]'
parameter
'salinity'           unit '[ppt]'
parameter
'test'              unit '[kg/m**3]'
records in table
2
    0.0  65.0  0.0  1.0
    1440.0 65.0  0.0  1.0
          'T-series for discharges run: 123'
contents
location
time-function
reference-time
19941001
time-unit
'minutes'
interpolation
'linear'
parameter
'flux/discharge rate' unit '[m**3/s]'
parameter
'salinity'           unit '[ppt]'
parameter
'test'              unit '[kg/m**3]'
records in table
2
    0.0  40.0  3.0  0.0

```

1440.0 40.0 3.0 0.0

**Remark:**

- ◊ The flow rate specified in this file may be overruled during the computation by means of the real-time control module. See [B.28](#)

A.2.23 Dredge and dump characteristics

File contents	Characteristics of all dredging, dumping and sediment nourishment activities in the simulation (Data Group <i>Operations - Dredge and dump</i>).
Filetype	ASCII
File format	Fix formatted for text variables, free formatted for real and integer values.
Filename	<name.dad>
Generated	QUICKIN, or manually offline

Record description:

The file consists of a header block (containing information about versions) followed by an optional general block with global settings for all dredge/dump blocks and all domains, one (or more) optional domain blocks with settings for the specified domain, a dredge block for each dredge area, a nourishment block for each sediment nourishment measure and an optional dump block for each dump area. [Table A.8](#) gives an overview of all supported keywords and their purpose. The name of the dredge and dump polygon file should be specified in the general block, but for reasons of backward compatibility it is also accepted that the `PolygonFile` is located in the header block.

Table A.7: Dredge and dump input file with keywords

Keyword	Format	Record description
DredgeFileInfo		
FileCreatedBy	string	program name and version
FileCreationDate	string	creation date and time of the <*.dad> file
FileVersion	string	version number of the <*.dad> file ('01.02')
General		
PolygonFile	string	name of polygon file with polylines defining dump and dredge areas
continued on next page		

R = Real; I = Integer; L = Logical; C = Character

Table A.7 – continued from previous page

Keyword	Format	Record description
InPolygon	1 I	default flag to indicate how to check whether a grid cell is inside or outside a dredge or dump polygon (1–3, default: 2). This option can be used to study the sensitivity of the simulation for changes in polygon sizes. A grid cell is inside the polygon if <ul style="list-style-type: none"> 1 all corners of the grid cell lie within the polygon (minimum area) 2 the centre of the grid cell lies within the polygon (default) 3 one of the corners of the grid cell lies within the polygon (maximum area)
RefPlane	1 R or string	elevation of reference plane above reference level of the simulation for dredging and dumping for this domain: constant value or file name (default: 0 m) Overruled by value for current domain.
DepthDef	1 I	default flag to indicate meaning of depth definitions (1–4; default: 1)) See description in dredge block.
DredgeDepth	1 R	default depth to be maintained by dredging (default: -999.) Set to -999. if dredging is not limited by depth.
MaxVolRate	1 R	default maximum dredging or sediment nourishment rate in m ³ /morphological year (volume including pores) (default: -999.) Set to -999. if dredging rate is unlimited.
DredgeDistr	1 I	default distribution of dredging activities when limited by rate (1–8, default: 3) See description in dredge block.
Clearance	1 R	default additional depth to dredge when dredging occurs (default: 0 m).
TriggerAll	1 L	default flag to indicate that if one point in the dredge area exceeds the dredging threshold, all points should be dredged to clearance level irrespective whether they exceed the trigger level (default: false)
DredgeWhenDry	1 L	default flag to indicate whether dredging should continue at dry points (default: false).
continued on next page		

R = Real; I = Integer; L = Logical; C = Character

Table A.7 – continued from previous page

Keyword	Format	Record description
DredgeWhileMorfac	0 L	default flag to indicate whether dredging activities should occur during morphological startup period or periods with morfac=0 (default: true). This flag affects only dredging activities not limited by dredging rate; activities that are limited by a maximum dredging rate are always inactive during startup and other periods with morfac=0 as the effective morphological time is zero.
ObeyCmp	1 L	default flag to indicate whether dredging should be limited to the amount of sediment available (default: true). If false: excess dredged volume is recorded but cannot be dumped as composition is unknown and considered to be irrelevant for the simulation.
TimeSeries	string, optional	name of file containing time-dependent dredging activities
TS_MorTimeScale	1 L	flag to indicate whether times in TimeSeries file are specified on hydrodynamical or morphological time scale (true = morphological time scale; default: false = hydrodynamical time scale)
DistrOverDump	1 I	default method of distributing sediment over dump areas. (1–3; default: 1) See description in dredge block.
DumpDistr	1 I	default method of distributing sediment within a dump area. (1–4; default: 1) See description in dump block.
MinimumDumpDepth	1 R	default minimum remaining depth after dumping: when specified, dumping continues until the specified level is reached. Any surplus of sediment is removed from the model (default: -999.) Set to -999. if dumping should not be restricted.
DumpWhenDry	1 L	default flag to indicate whether dumping should continue at dry points (default: false)
UseDunes	1 L	flag to indicate whether simulated dune heights should be included in volume computations when dredging and dumping; if included, dune height will be set to 0 after dredging and dumping (default: false)
AlphaDuneHeight	[0 0.5]	if UseDunes is true, then the effective bed level is increased by this fraction of the dune height when checking whether dredging threshold is exceeded (default: 0.5)
continued on next page		

R = Real; I = Integer; L = Logical; C = Character

Table A.7 – continued from previous page

Keyword	Format	Record description
Domain		
Name	string	name of the domain
RefPlane	1 R or string	elevation of reference plane above reference level of the simulation for dredging and dumping for this domain: constant value or file name (default: constant as specified in General block) The format of the file is a QUICKIN compatible file with data at cell centres.
Dredge		
Name	string	name of the dredge area: should correspond to a blockname in the polygon file
InPolygon	1 I	flag to indicate how to check whether a grid cell is inside or outside the dredge polygon (1–3, default: as indicated in General block). See description in General block.
DepthDef	1 I	flag to indicate meaning of depth definitions (1–4; default: as specified in General block) <ul style="list-style-type: none"> 1 relative to reference plane 2 relative to water level 3 relative to maximum of reference plane and water level 4 relative to minimum of reference plane and water level
DredgeDepth	1 R	depth to be maintained by dredging relative (default: as specified in General block) Set to -999. if dredging is not limited by depth.
MaxVolRate	1 R	maximum dredging rate in m ³ /morphological year (volume including pores) (default: as specified in General block) Set to -999. if dredging rate is unlimited. This maximum volume per year will only be reached if dredging takes place year-round and never runs out of available dredging material, i.e. it doesn't increase the dredging rate per time step if the dredging strategy isn't active for a while.
continued on next page		

R = Real; I = Integer; L = Logical; C = Character

Table A.7 – continued from previous page

Keyword	Format	Record description
DredgeDistr	1 I	<p>in case the maximum dredging rate is reached in a certain timestep, the model has to decide what sediment to remove (1–8; default: as specified in General block). Eight rules have been implemented</p> <ul style="list-style-type: none"> 1 remove same sediment thickness from every grid cell 2 remove highest sediment deposits first 3 remove sediment proportionally to the amount of sediment available for dredging (default) 4 remove sediment from highest point first (gives more or less same results as 2 for small time steps, but may give erratic behavior when large time steps are used) 5 remove sediment from lowest point first 6 remove sediment from shallowest areas first 7 remove sediment from shallowest point first (gives more or less same results as 6 for small time steps, but may give erratic behavior when large time steps are used) 8 remove sediment from deepest point first
Clearance	1 R	additional depth to dredge when dredging occurs (default: as specified in General block)
TriggerAll	1 L	flag to indicate that if one point in the dredge area exceeds the dredging threshold, all points should be dredged to clearance level irrespective whether they exceed the trigger level (default: as specified in General block)
DredgeWhenDry	1 L	flag to indicate whether dredging should continue at dry points (default: as specified in General block)
DredgeWhileMorfac \neq 0	1 L	flag to indicate whether dredging activities should occur during morphological startup period or periods with morfac=0 (default: as specified in General block). This flag affects only dredging activities not limited by dredging rate.
ObeyCmp	1 L	flag to indicate whether dredging should be limited to the amount of sediment available (default: as specified in General block). If false: excess dredged volume is recorded but cannot be dumped as composition is unknown and considered to be irrelevant for the simulation.
continued on next page		

R = Real; I = Integer; L = Logical; C = Character

Table A.7 – continued from previous page

Keyword	Format	Record description
DumpLimited	1 L	flag to indicate that dredging is limited by dump capacity (default: false)
DistrOverDump	1 I	method of distributing sediment over dump areas (old keyword DumpDistr still supported). Three methods have been implemented (1–3, default: 1) <ul style="list-style-type: none"> 1 based on user specified percentages 2 fill dump areas in specified order 3 distribute sediment based on dump capacity
Dump	string	name of the dump area: should correspond to a blockname in the polygon file
Sediment	string, optional	name of sediment fraction
Percentage	1 R	percentage of dredged material of the indicated sediment fraction to be dumped in the indicated dump area; if no sediment fraction is indicated: uniform value for all fractions.
Nourishment		
Name	string, optional	name of the nourishment source. Contrary to the Name in the dredge and dump blocks, this name is not associated with a specific polygon.
Volume	1 R	total sediment volume (including pores) to be supplied
MaxVolRate	1 R	maximum sediment nourishment rate in m ³ /morphological year (volume including pores) (default: as specified in General block) This rate will be converted into a rate per time step assuming that nourishment occurs at a constant rate year-round, i.e. it does not try to compensate for periods of limited dumping capacity or periods of inactivity as optionally specified in TimeSeries file.
Sediment	string	name of sediment fraction
SedPercentage	1 R	composition of supplied sediment: volumetric percentage of the indicated sediment fraction
continued on next page		

R = Real; I = Integer; L = Logical; C = Character

Table A.7 – continued from previous page

Keyword	Format	Record description
DistrOverDump	1 I	method of distributing sediment over dump areas (old keyword DumpDistr still supported). Three methods have been implemented (1–3, default: 1) <ul style="list-style-type: none"> 1 based on user specified percentages 2 fill dump areas in specified order 3 distribute sediment based on dump capacity
Dump	string	name of the dump area: should correspond to a blockname in the polygon file
Sediment	string, optional	name of sediment fraction
Percentage	1 R	percentage of dredged material of the indicated sediment fraction to be dumped in the indicated dump area; if no sediment fraction is indicated: uniform value for all fractions.
Dump		
Name	string	name of the dump area: should correspond to a blockname in the polygon file
InPolygon	1 I	flag to indicate how to check whether a grid cell is inside or outside the dump polygon (1–3, default: as indicated in General block). See description in General block.
DepthDef	1 I	flag to indicate meaning of depth definitions (1–4; default: as specified in General block) <ul style="list-style-type: none"> 1 relative to reference plane 2 relative to water level 3 relative to maximum of reference plane and water level 4 relative to minimum of reference plane and water level
MinimumDumpDepth	1 R	minimum remaining depth after dumping: when specified, dumping continues until the specified level is reached. Any surplus of sediment is removed from the model (default: as specified in General block) Set to -999. if dumping should not be restricted.
continued on next page		

R = Real; I = Integer; L = Logical; C = Character

Table A.7 – continued from previous page

Keyword	Format	Record description
DumpDistr	1 I	determines how the sediment is dumped within the dump area. Four options have been implemented (1–4; default as specified in General block) <ul style="list-style-type: none"> 1 dump with uniform thickness 2 fill in lowest parts first (horizontal reference plane) 3 fill in deepest parts first 4 dump proportional to capacity (only for dumping areas with limited capacity)
DumpWhenDry	1 L	flag to indicate whether dumping should continue at dry points (default: as specified in General block)

R = Real; I = Integer; L = Logical; C = Character

Warning:

- ◊ QUICKIN does not yet recognize all keywords and will remove all keywords it does not recognize when you attempt to edit a manually edited dredge and dump file with QUICKIN. Backup old <*.dad> files before you try to do so!



Restrictions:

- ◊ The maximum record length in the file is 256 characters.
- ◊ Dredge and dump activities may be combined with domain decomposition simulations. However, please note that in such cases one should select the same <*.dad> for all domains and use the **domain** block if there are differences between the domains.
- ◊ Dumping conditions (to a certain minimum depth or to flooded points only) may not exactly be satisfied if multiple dredging and dumping overlap. In case of dredging limited by dumping capacity similar interactions may also cause deviations from requested behaviour.



Remarks:

- ◊ The order of the data blocks is arbitrary.
- ◊ To remove sediment from the model, specify one (or more) dump areas outside the grid.
- ◊ Dredging measures are processed in the order in which the **dredge** blocks are specified in the input file.
- ◊ Dumping measures are processed in the order in which they are specified in the **dredge** and **nourishment** blocks of the input file.



Example:

The number of dredge areas is 3. ‘dredge 1’ and ‘dredge 2’ will be dredged to a depth of 10 m. All dredged material is dumped in area ‘dump 1’. Area ‘dredge 3’ is dredged to a depth of 20 m and the material removed from that area is dumped in ‘dump 2’.

```
[DredgeFileInfo]
FileCreatedBy      = Deltares, QUICKIN Version 4.20.00.30984, Nov 12 2013, 09:50:44
FileCreationDate  = 2013-11-28, 16:01:21
FileVersion       = 01.03
[General]
PolygonFile       = dad_areas.pol
```

```
[Dredge]
  Name      = dredge 1
  DredgeDepth =    10.00
  Dump      = dump 1
  Percentage = 100.0
[Dredge]
  Name      = dredge 2
  DredgeDepth =    10.00
  Dump      = dump 1
  Percentage = 100.0
[Dredge]
  Name      = dredge 3
  DredgeDepth =    20.00
  Dump      = dump 2
  Percentage = 100.0
```

A.2.24 Dredge and nourishment time-series

File contents	Time-series for dredge and nourishment activities; filename specified in dredge and dump characteristics file.
Filetype	ASCII
File format	Fix format for header information; free format for time-series data.
Filename	<name.ddt>
Generated	manually offline

Record description:

The file may contain one or more time-series for dredging and sediment nourishment activities. A time-series will be used for all dredging and sediment nourishment activities if the location equals ‘General’. It will be used only for a single dredging or sediment nourishment activity if the location equals the name of the dredging or sediment nourishment activity. The order of the paired header/data blocks is arbitrary.

Description header block:

Keyword	Record description
location	‘General’ or name of dredging or sediment nourishment activity (1 quoted string)
time-function	time function type (quoted string: ‘non-equidistant’)
reference-time	reference time (integer formatted as yyymmdd or quoted string ‘from model’)
time-unit	time unit (quoted string: ‘decades’, ‘years’, ‘days’, ‘hours’, ‘minutes’, ‘seconds’, ‘ddhhmmss’, ‘absolute’)
interpolation	interpolation type (quoted string: ‘block’)
parameter	parameter name (quoted string: ‘time’ or ‘active’) followed by keyword unit and dummy unit specification (quoted string: time-unit or ‘-’)

Description data block:

Record	Record description
each record	time specified in time-unit since reference-time or absolute time specified as an integer formatted as yyyyymmddhhmmss value indicating whether dredging (or sediment nourishment) is active from the specified time onwards (0=inactive, 1=active).

Example:

Model with multiple dredging and nourishment activities. All specified activities will be active during the first two days except for the dredging activity called 'dredge 3' which is active during days two till four.

```

location           'General
time-function     'non-equidistant
reference-time    'from model'
time-unit          'days'
interpolation      'block'
parameter          'time
parameter          'active
      0.0000      1.0
      2.0000      0.0
location           'dredge 3
time-function     'non-equidistant
reference-time    'from model'
time-unit          'days'
interpolation      'block'
parameter          'time
parameter          'active
      0.0000      0.0
      2.0000      1.0
      4.0000      0.0

```

A.2.25 Polygon file

File contents

The co-ordinates of one or more polygons. Each polygon is written in a single block of data

Filetype

ASCII

File format

Free formatted

Filename

<name.pol>

Generated

RGFGRID, QUICKIN, D-Waq DIDO, etc

Record description:

The file may contain one or more polygons. For every polygon the file should contain a line indicating the name of the polygon, followed by a line indicating the number of points making up the polygon and the number of coordinates, i.e. 2, finally followed by the coordinate data.

Record	Record description
	Preceding description records, starting with an asterisk (*), and will be ignored.
1	A non blank character string, starting in column one
2	Two integers representing the numbers of rows and number of columns for this block of data
	Two reals representing the x, y or λ, ϕ -coordinate

**Restriction:**

- ◊ The first record and the last record in the block should be the same

Example:

```

*
* Deltares, Delft3D-DIDO Version 3.39.01.4423:4459, Sep 25 2008, 20:10:54
* 2008-09-25, 22:11:08
*
Observation Area 001
      5   2
1.8768018E+05  6.1708738E+05
1.8996981E+05  6.1001035E+05
1.9746314E+05  6.1266423E+05
1.9480925E+05  6.1838830E+05
1.8768018E+05  6.1708738E+05
Observation Area 002
      5   2
2.0011703E+05  6.1818015E+05
1.9819166E+05  6.1063479E+05
2.0568498E+05  6.0870942E+05
2.0797461E+05  6.1599460E+05
2.0011703E+05  6.1818015E+05
Observation Area 003
      5   2
1.9340425E+05  6.1396516E+05
2.0183425E+05  6.1365294E+05
1.9944054E+05  6.0558720E+05
1.9522555E+05  6.0595146E+05
1.9340425E+05  6.1396516E+05

```

A.2.26 Observation points

The observation points file can be specified in the FLOW-GUI in Data Group *Monitoring - Observations*.

File contents	Description of observation points characterised by their name and grid indices.
Filetype	ASCII
File format	Fix formatted for text variables, free formatted for integer values.
Filename	<name.obs>
Generated	FLOW-GUI or manually offline

Record description:

Record	Record description
each record	Name of observation point (20 characters), Location of the observation point in grid indices (2 integers).

Restrictions:

- ◊ One record per observation point.
- ◊ The name of the observation point must start in position one.
- ◊ The maximum record length is 132.
- ◊ Input items in a record must be separated by one or more blanks.

**Example:**

File containing 5 observation points:

Paradise Bay 1	10	11
Paradise Bay 2	11	11
Sea Entrance	4	8
Port Arthur	18	85
Deep Channel	22	44

Remarks:

- ◊ The observation points are stationary by default. A moving observation point can be specified by defining a stationary observation point in the above-mentioned observation points file and by subsequently specifying its path via a moving observation point file (keyword Filmst) or by linking it to a drogue.
- ◊ The position of an observation point is automatically linked to that of a drogue if the same name is used for both the observation point and the drogue. If an observation point is linked to a drogue, the position of that observation point specified in the observation file described above is overruled.
- ◊ If the path of an observation point is specified via a moving observation point file, the position of that observation point specified in the observation file described above is overruled. For a description of the moving observation point file see Section A.2.27.
- ◊ If an observation point moves out of the simulation domain, a missing value of -999.0 will be returned for all quantities.
- ◊ If the name of an observation point matches both a record in the moving observation point file and the name of a drogue, then the path of the drogue will be used.

**A.2.27 Moving observation points**

The `<name.mst>`-file specifies the path of a moving observation point. Its file format resembles closely the file format of time series conditions at open boundaries. The moving observation points file can be specified in the *Additional Parameters* section of the FLOW-GUI by means of the keyword `Filmst`:

```
Filmst = #name.mst#
```

File contents	Time-series of the paths of moving observation points
Filetype	ASCII
File format	Fix format for header information; free format for time-series data.
Filename	<code><name.mst></code>
Generated	manually offline

Record description:

For each observation point that should be moving along a pre-defined path, the path is specified by means of two related blocks:

- ◊ A header block containing a number of compulsory and optional keywords accompanied by their values.
- ◊ A data block containing the time dependent data.

Description header block:

Text	Required	Value
table-name	no	arbitrary string
location	yes	'name observation point' (see < <i>name.obs</i> > file)
time-function	no	{'non-equidistant'} or 'equidistant'
time-step	yes	time step only in case of time-function 'equidistant'
reference-time	yes	yyyymmdd, yyymdd hhmmss or 'from model'
time-unit	no	'years', 'decades', 'days', 'hours', {'minutes'}, 'seconds', 'ddhhmmss' or 'date'
interpolation	no	{'linear'} or 'block'
extrapolation	no	'periodic', 'constant' or {'none'}
parameter	yes	'time' unit '[min]'
parameter	yes	'x-coordinate' units '[m]' (or '[deg]')
parameter	yes	'y-coordinate' units '[m]' (or '[deg]')
records-in-table	no	number of times/lines in the data block

**Remarks:**

- ◊ Default parameter values are indicated in braces.
- ◊ Reference-time not required if time-unit equals 'date'.
- ◊ Unit strings are currently not interpreted by Delft3D-FLOW.
- ◊ Only tables for which the location name matches the name of an observation point specified in the <*name.obs*> file will be used.

Description data block:

Record	Record description
each record	Time in time-units after the reference-time and followed by <i>x</i> - and <i>y</i> -coordinates (all reals). In case of time-function 'equidistant', the first (time) column should be dropped. In case of time-unit 'date' the date and time should be specified as one string using the format: yyyymmddhhmmss.

Remark:

- ◊ Maximum record length is 512.

**Example:**

The example file specifies the tracks of two moving observation points named 'Steamer 1' and 'Raft 2' respectively. The actual model may include zero, one or two moving observation points depending on the number of location names that match observation point names specified in the <*.obs> file.

```

location          'Steamer 1'
time-function    'non-equidistant'
reference-time   20100118
time-unit        'minutes'
interpolation    'linear'
parameter        'time
parameter        'x-coordinate
parameter        'y-coordinate
records-in-table 3
      0 1750 1450
     1440 2250 1950
    2880 1750 2450
location          'Raft 2'
time-function    'non-equidistant'
reference-time   20100118
time-unit        'minutes'
interpolation    'linear'
parameter        'time
parameter        'x-coordinate
parameter        'y-coordinate
records-in-table 2
      0 2050 1550
    2880 3050 1550
  
```

A.2.28 Drogues

File contents	Description of drogues characterised by their name, release indices, release and recovery time (Data Group <i>Monitoring - Drogues</i>).
Filetype	ASCII
File format	Fix formatted for text variables, free formatted for real and integer values.
Filename	<name.par>
Generated	FLOW-GUI or manually offline

Record description:

Record	Record description
each record	Name of the drogue (20 characters), Release time in minutes relative to 00:00:00 on the Reference Date (1 real), Recovery time in minutes relative to 00:00:00 on the Reference Date (1 real), Grid indices of the release point of the drogue (2 reals).

Remark:

- ◊ Drogues can be released anywhere inside a grid cell, i.e. at fractional grid indices. The



default value, when using the **Visualisation Area** window, is the grid centre; fractional indices must be defined manually.



Restrictions:

- ◊ One record per drogue.
- ◊ The drogue's name must start in position one.
- ◊ Maximum record length is 132.
- ◊ Input items in a record must be separated by one or more blanks.

Example:

File contains 5 drogues. All drogues are released and recovered at the same time: released at 1200 minutes after the midnight on the simulation Reference Date and recovered 300 minutes later. All drogues are released in the grid cells surrounding depth point with grid indices (10,11).

```
Drogue at Station: A    1200.00  1500.00  10.0   11.0
Drogue at Station: B    1200.00  1500.00  10.5   11.0
Drogue at Station: C    1200.00  1500.00  10.0   11.5
Drogue at Station: D    1200.00  1500.00  9.5    11.0
Drogue at Station: E    1200.00  1500.00  10.0   10.5
```

A.2.29 Cross-sections

File contents	The description of cross-sections characterised by their name and grid indices (<i>Data Group Monitoring - Cross-sections</i>).
Filetype	ASCII
File format	Fix formatted for text variables, free formatted for integer values.
Filename	<name.crs>
Generated	FLOW-GUI or manually offline

Record description:

Record	Record description
each record	Name of the cross-section (20 characters), Grid indices of the begin and end point of the cross-section (4 integers).



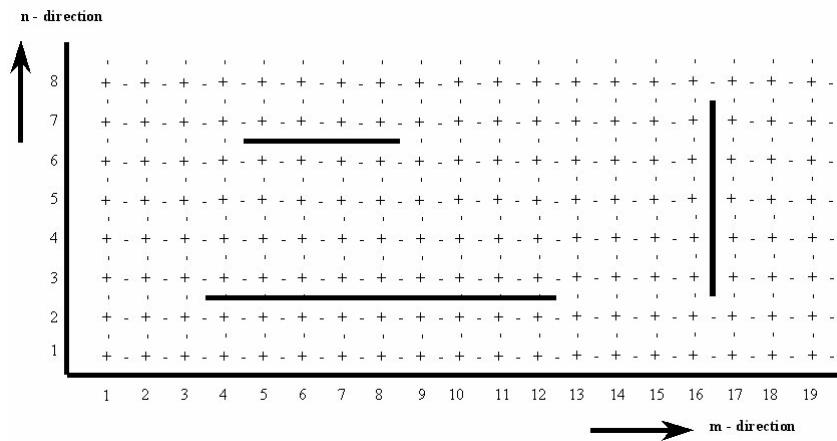
Restrictions:

- ◊ One record per cross-section
- ◊ The cross-section's name must start in position one.
- ◊ Maximum record length is 132.
- ◊ A cross-section must contain at least two consecutive grid cells.
- ◊ Name and indices are separated by one or more blanks.

Example:

File containing 3 cross-sections, two 'V-cross-sections' and one 'U-cross-section', see [Figure A.6](#).

```
Bridge          5     6     8     6
River Edo - Entrance 16     3    16     7
Sea Entrance      4     2    12     2
```

**Figure A.6:** Cross-sections in model area

A.2.30 Fourier analysis

File contents

All parameters required to execute an online Fourier analysis for specified quantities, a specified period and for specified frequencies (Data Group *Output - Storage*).

Filetype

ASCII

File format

Fix formatted for text variables, free formatted for real and integer values.

Filename

<name.fou>

Generated

Manually offline

Record description:

Record	Record description														
each record	<p>Variable on which the Fourier analysis is to be performed (2 characters):</p> <table> <tr><td>wl</td><td>water levels</td></tr> <tr><td>uv</td><td>velocities</td></tr> <tr><td>qf</td><td>mass fluxes of water</td></tr> <tr><td>ct</td><td>temperature</td></tr> <tr><td>cs</td><td>salinity</td></tr> <tr><td>cn</td><td>n-th constituent in the MDF-file</td></tr> <tr><td>bs</td><td>bed stress</td></tr> </table> <p>Analysis start time in minutes after 00:00:00 on the Reference Date, Analysis stop time in minutes after 00:00:00 on the Reference Date, Number of cycles within the analysis time frame, Nodal amplification factor, Astronomical argument, Layer number for the Fourier analysis, Flag for the computation of elliptic parameters: y/n (default no) if number of cycles > 0, min/max if number of cycles = 0.</p>	wl	water levels	uv	velocities	qf	mass fluxes of water	ct	temperature	cs	salinity	cn	n-th constituent in the MDF-file	bs	bed stress
wl	water levels														
uv	velocities														
qf	mass fluxes of water														
ct	temperature														
cs	salinity														
cn	n-th constituent in the MDF-file														
bs	bed stress														

Remarks:

- ◊ If the number of cycles is set equal to 0, the mean level of the variable over the interval specified by the start and stop time is determined.
- ◊ If, in addition, the flag for the computation of elliptic parameters is set to "max" or "min"



the maximal or minimal value is determined of the selected variable as it occurred during the simulation.

- ◊ Computed Fourier amplitudes slightly differ from the amplitude of the corresponding tidal component. When comparisons with co-tidal maps have to be made, this factor can be determined using the subsystem ASCON of Delft3D-TIDE, the tidal analysis package of Deltares.
- ◊ Computed Fourier phases are related to the reference date of the FLOW computation. For comparisons with co-tidal maps a phase shift equal to the astronomical argument has to be applied.
- ◊ The layer number is not relevant for water levels.



Restrictions:

- ◊ Times specified must be a multiple of the computational time step.
- ◊ Times specified must be a valid time, i.e. must fit in the simulation time frame of the FLOW computation.
- ◊ Items in a record must be separated by one or more blanks.
- ◊ The variable's name must start in position one.

Example:

A Fourier analysis is requested for:

- ◊ Water level: mean value and the first two harmonics.
- ◊ U- and V-velocity: first harmonic, in the top layer (1).
- ◊ Mass fluxes of water: first harmonic, in the top layer (1).
- ◊ Temperature: first harmonic, in the third layer.
- ◊ Salinity: mean value of the third layer.
- ◊ Four constituents: mean value for a slightly shifted time period in the top layer for 3 constituents; and the maximum for the fourth constituent.

wl	720.	1440.	2	1.000	0.000	
wl	720.	1440.	1	1.000	0.000	
wl	720	1440.	0	1.000	0.000	
uv	720.	1440.	1	1.000	0.000	1
qf	720.	1440.	1	1.000	0.000	1
ct	720.	1440.	1	1.000	0.000	3
cs	720.	1440.	0	1.000	0.000	3
c1	710.	1430.	0	1.000	0.000	1
c2	710.	1430.	0	1.000	0.000	1
c3	710.	1430.	0	1.000	0.000	1
c4	710.	1430.	0	1.000	0.000	1 max



Remark:

- ◊ A layer value of 0 in a 3D model is not allowed; i.e. this will NOT result in a depth-averaged value.

A.2.31 (Rigid) 3D vegetation model

To include vegetation in Delft3D-FLOW, the keyword `Filpla` may be included in the FLOW input file <*.mdf>, e.g.:

```
Filpla=#example.pla#
```

In the FLOW-GUI this keyword-value pair can be specified in Data Group *Additional Parameters*.

File contents	Characteristics of vegetation.
Filetype	ASCII
File format	Ini formatted.
Filename	<name.pla>
Generated	manually offline

Record description:**Table A.8:** Vegetation input file with keywords

Keyword	Format	Record description
VegetationFileInfo		
FileCreatedBy	string	program name and version
FileCreationDate	string	creation date and time of the <*.pla> file
FileVersion	string	version number of the <*.pla> file ('01.00')
General		
PolygonFile	string	name of polygon file with polylines defining vegetation areas
C1Plant	1 R	The overall turbulence length scale C_l between stems (Equation 9.361).
ItPlant	1 I	Number of time steps between updates of plant arrays
Vegetation		
type	string	Type of the vegetation
Vps	2 R, 1 I, 1 R	Specification of the vertical plant structure (Vps). For each specified plant type, there should be a block in the file containing four columns: 1 height [m] 2 stem diameter [m] 3 number of stems [-] 4 C_d coefficient [-]
Area		
VegetationType	string	name between # must match a vegetation group in this file.
Polygon	string	name between # as specified the polygon file defined by the keyword <code>PolygonFile</code>
continued on next page		

R = Real; I = Integer; L = Logical; C = Character

Table A.8 – continued from previous page

Keyword	Format	Record description
NPlants or NPlantsFile	I string	Number of stems per square metre [m^2] name between # must match an existing file in dep-format (see subsection A.2.4)

R = Real; I = Integer; L = Logical; C = Character

In the plant input file, the overall turbulence length scale C_l can be specified ([Equation 9.361](#)), using the keyword ClPlant.

If more types of plants are specified in the same grid cell, the average stem diameter and the average number of stems per unit area are combined such that the combination gives the total resistance and total occupied plant areas. The number of stems per unit area may vary over time if a sigma layer passes a plant top. Then, the number of stems and their diameters will be redistributed over the vertical. The update may not be necessary every time step if the water level varies slowly. The update interval may be increased to save some computational effort, using the keyword ItPlant.

In the *(Rigid) 3D vegetation model* input file many different plant types may be specified, each type with its own vertical plant structure and horizontal spatial distribution.

The Vertical Plant Structure (VPS) describes the number of stems and their widths per plant as a function of the vertical co-ordinate. For each specified plant type, there should be a block in the file containing four columns:

- 1 the height of the plants,
- 2 number of stems,
- 3 stem diameter and
- 4 the friction coefficient

The horizontal distribution of the number of plants per m^2 can be specified for each plant type separated in two ways.

- 1 by means of a constant value (NPlants)
- 2 by means of reference to a file (NPlantsFile). The file format is according the dep-file format, see [subsection A.2.4](#)

Some example input files are given in Figures [A.7](#) to [A.9](#)

```
[VegetationFileInfo]
  FileCreatedBy      = Delft3D Support
  FileCreationDate  = 29-11-2013
  FileVersion       = 01.00
[General]
  PolygonFile = plants.pol
  ClPlant     = 0.80      [ - ] Turbulence length scale coefficient between stems
  ItPlant     = 50        [ - ] Number of time steps between updates of plant arrays
[Vegetation]
  Type = reed
  *
  *   height [m]      stem diameter [m]      nr of stems [-]      cd-coefficient [-]
  *
  Vps  =  0.0          0.008                1                  1.0
  Vps  =  0.90         0.008                1                  1.0
  Vps  =  0.91         0.001                1                  1.0
[Area]
  VegetationType = #reed#           [ - ] must match a vegetation group in this file
  Polygon       = #reed#           [ - ] must match a polygon in the PolygonFile
  NPlants      = 256             [ 1/m2 ]
```

Figure A.7: Example of the plant input file (<name.pla>) where the areas are defined with a polygon file, see [subsection A.2.25](#)

```
[VegetationFileInfo]
  FileCreatedBy      = Delft3D Support
  FileCreationDate  = 02-07-2004
  FileVersion       = 01.00
[General]
  ClPlant = 0.80      [ - ] Turbulence length scale coefficient between stems
  ItPlant = 50        [ - ] Number of time steps between updates of plant arrays
[Vegetation]
  Type = reed
  *
  *   height [m]      stem diameter [m]      nr of stems [-]      cd-coefficient [-]
  *
  Vps  =  0.0          0.008                1                  1.0
  Vps  =  0.90         0.008                1                  1.0
  Vps  =  0.91         0.001                1                  1.0
[Area]
  VegetationType = #reed#      #      [ - ] must match a vegetation group in this file
  NPlantsFile     = #reed.dep#    [ - ] must match an existing file in dep-format
```

Figure A.8: Example of the plant input file (<name.pla>) where the area is defined with files according the depth-format, see [subsection A.2.4](#)

```

[VegetationFileInfo]
  FileCreatedBy      = Delft3D Support
  FileCreationDate  = 29-11-2013
  FileVersion       = 01.00
[General]
  C1Plant = 0.80      [ - ] Turbulence length scale coefficient between stems
  ItPlant = 50         [ - ] Number of time steps between updates of plant arrays
[Vegetation]
  Type = wier1
  *
  * height [m]      stem diameter [m]      nr of stems [-]      cd coefficient [-]
  *
  Vps = 0             0.003                 1000                  1.0
  Vps = 0.5           0.003                 1000                  1.0
[Vegetation]
  Type = wier2
  *
  * height [m]      stem diameter [m]      nr of stems [-]      cd coefficient [-]
  *
  Vps = 0             0.003                 1000                  1.0
  Vps = 0.8           0.003                 1000                  1.0
[Area]
  VegetationType = #wier1    #
  NPlantsFile   = #vegetatiedichtheid_gemaaid_wier.dep#
[Area]
  VegetationType = #wier2    #
  NPlantsFile   = #vegetatiedichtheid_niet_gemaaid_wier.dep#

```

Figure A.9: Example of the plant input file (<name.pla>) where two different vegetation types are defined.

A.2.32 Space varying subsidence uplift definition

To include subsidence/uplift in Delft3D-FLOW, the keyword Filsdu may be included in the FLOW input file <*.mdf>, e.g.:

```
Filsdu=#example.sdu#
SduS1 =#N#
```

The value of the keyword SduS1 determines whether or not the water level is adjusted as well (default: no). In the FLOW-GUI these keyword-value pairs can be specified in Data Group *Additional Parameters*.

In this section a description of the space-varying subsidence/uplift is given.

File contents	Time-series for space varying uplift (or subsidence), defined on the computational grid. The file consists of a header, followed by data-blocks containing the uplift field at times specified using a standardised time definition above each datablock. The header specifies the type of file and the input it contains using a number of keywords. The keywords are case insensitive and the order of the keywords is not fixed.
Filetype	ASCII or binary.
File format	Free formatted or unformatted, keyword based.
Filename	<name.sdu>
Generated	Some offline program.

Header description:

Keywords	Value	Description
FileVersion	1.03	version of file format
Filetype	field_on_computational_grid	field input on computational grid
NODATA_value	<i>free</i>	value used for input that is to be neglected
n_quantity	1	number of quantities specified in the file
quantity1	bedrock_surface_elevation	bed rock surface elevation
unit1	m	unit of quantity1, meters w.r.t. to arbitrary datum

Time definition and data block description

Keywords	Value	Description
Time	<i>fixed format described below</i>	time definition string

The time definition string has a fixed format, used to completely determine the time at which a dataset is valid. The time definition string has the following format:

TIME *minutes/hours since YYYY-MM-DD HH:MM:SS TIME ZONE*, e.g.

```
360 minutes since 2008-07-28 10:55:00 +01:00
```

The format of the string is completely fixed. No extra spaces or tabs can be added between the different parts of the definition. The time definition is followed by the datablock of input values corresponding to the specified time. The data block consists of a single block containing the bedrock surface elevation in the *Nmax* by *Mmax* water level points, where the first value in the dataset corresponds to cell (1, 1) on the grid. Every next line in the dataset then corresponds to a row on the grid. The time definition and the data block are repeated for each time instance of the time-series.

File version and conversion

The current description holds for FileVersion 1.03.

**Restrictions:**

- ◊ Keywords are followed by an equal sign '=' and the value of the keyword.
- ◊ When a keyword has value *free* the value of this keyword is free to choose by the user. When only one value is given for a keyword, this keyword has a fixed value and when 2 or more options are shown, the user can choose between those values.
- ◊ Times must be specified exactly according to the time definition. See the examples shown in this section.
- ◊ The contents of the file will not be checked on its domain.
- ◊ Input items in a data block are separated by one or more blanks (free formatted file only).

**Remarks:**

- ◊ The time definition in the subsidence/uplift (.sdu) file is handled similarly to space varying wind and pressure defined on the computational grid (Section).
- ◊ Comments can be added after #'s.

Example

Model area of 250 * 33 grid points (Mmax = 250; Nmax = 33). The input data is printed in Courier, comments are printed behind #'s.

```
Time = 0.0 minutes since 2008-09-20 10:30:00 +01:00          # Time definition
{33 records with 250 values each}
Time = 340.0 minutes since 2008-09-20 10:30:00 +01:00          # Bedrock surface elevation w.r.t. to arbitrary datum
{33 records with 250 values each}
Time = 600.0 minutes since 2008-09-20 10:30:00 +01:00          # Time definition
{33 records with 250 values each}
Time = 1240.0 minutes since 2008-09-20 10:30:00 +01:00          # Bedrock surface elevation w.r.t. to arbitrary datum
{33 records with 250 values each}                                # Time definition
{33 records with 250 values each}                                # Bedrock surface elevation w.r.t. to arbitrary datum
```

**Remark:**

- ◊ The subsidence/uplift file (.sdu) is attached to the master definition file (.mdf) using the *FlsSdu* keyword. To include the effect of change in the bedrock surface elevation upon the water level, the *SduS1* keyword should be set to #Y# or #N# (= default).

B Special features of Delft3D-FLOW

B.1 Introduction

In the FLOW-GUI in Data Group *Additional parameters* you can specify certain parameters and values that initiate a special feature to Delft3D-FLOW.

When you want to use such a special feature you must specify its keyword and value in the Data Group *Additional parameters*. The keyword is used in the MDF-file to recognise the special feature and the value is in most cases the name of the file that provides additional input data for that feature. In this section we describe the keywords and files related to the special features that are currently available. Details of the formulations of most special features are given in Chapters 9, 10, 11 and 12. For some special features the details of the formulations are given in this appendix, such as:

- ◊ Horizontal Large Eddy Simulations (HLES) and
- ◊ Domain decomposition.

The currently available special features are:

Table B.1: Special features of Delft3D-FLOW

Functionality	Description
B.2 Decay rate constituents	Define a linear decay rate for each constituent
B.3 Hydraulic structures rigid sheets Floating structures 3D gates local weir porous plate bridge pier 2D weir Upwinding at discharges current deflection wall	Hydraulic structures that influence the flow Impermeable sheets in (part of) the vertical Floating structure at the free surface Gate structures in (part of) the vertical Weirs Porous plate in (part of) the vertical Sub-grid piers extending over full vertical Spillways, weirs and summer dikes (2D) Upwind advection scheme near discharge locations Current deflecting wall
B.4 Space varying Coriolis coefficients	Coriolis coefficient for each grid point
B.5 Temperature modelling	Temperature modelling
B.6 Evaporation and precipitation	User-defined evaporation and rainfall
B.7 Space varying wind and pressure	Space varying wind and pressure input
B.8 Horizontal large eddy simulation	Sub-grid scale HLES model
B.9 Sediment transport and morphology	see also chapter 11
B.10 Fluid mud (2-layer approach)	see section B.10
B.11 Z-model	see also chapter 12 and section B.11
B.12 Non-hydrostatic solver	see also chapter 12 and section B.12
B.14 Domain decomposition	see section B.14

Table B.1: Special features of Delft3D-FLOW

Functionality	Description
B.15 Surfbeat/roller model	see section B.15
B.19 Dry run	see section B.19
B.20 Reuse temporary files	see section B.20
B.21 Change the update frequency of the nodal factors	see section B.21
B.22 Bubble screen	Injection of air to enhance vertical mixing
B.23 1D–3D Coupling	See section B.23
B.24 Output of Courant number messages	See section B.24
B.25 Initialisation of water depth in dry points	See section B.25
B.26 Remapping of near-bottom layers for accurate and smooth bottom shear stress in Z-layers	See section B.26
B.27 Slope Limiter	See section B.27

**Remark:**

- ◊ An important restriction of the current implementation of structures in Delft3D-FLOW is that only one type of structure is accepted at a grid, (m, n) , location. This means you cannot combine a local weir and a porous plate or 3D gate at the same grid location. However, you can combine two 3D gates, one extending over layer 1 and another 3D gate extending over layer 3 to 5, at the same grid, (m, n) , location.

A specific type of special feature is the User Defined Function. It is a more powerfull definition of a special feature but its definition is more complex. These User Defined Functions are described in [section B.13](#).

The currently available User Defined Functions are:

Functionality	Description
BC turbulence model	Boundary conditions for the turbulence model
Diagnostic mode	Specify to use the diagnostic mode
Particle wind factor	An additional wind speed working on particles

B.2 Decay rate constituents

Keyword	Value	Description	Default
Decay#	real number	Linear decay rate of constituent # where # = {1 to 9}	0.0 [1/day]
Dcay##	real number	Linear decay rate of constituent ## where ## = {10 to 99}	0.0 [1/day]

For the (conservative) constituents defined in the Data Group *Processes* → *Pollutants and tracers* you can define a linear decay rate. Decay rates cannot be specified for salinity, temperature and suspended sediment fractions. The number of the constituent following the `Decay` or `Dcay` keyword should match the number of the corresponding `NamC` keyword indicating the name of the constituent. The decay rate is defined *per day*.

The decay rate is implemented as a sink term on the right hand side of the transport equation, i.e.

$$\frac{\partial c}{\partial t} + (\text{advection}) = (\text{dispersion}) - K \cdot c, \quad (\text{B.1})$$

where K is the decay rate.

Remark:

- ◊ The keyword is case sensitive!



B.3 Hydraulic structures

B.3.1 3D gates

Keyword	Value	Description	Default
<code>Filgat</code>	<code>#name.gat#</code>	Filename for 3D gate data	none

A 3D gate is represented by a vertical plate in the flow along one of the grid directions and that covers some or all layers in the vertical, [Figure B.1](#).

There is no momentum exchange across the gate and friction exerted on the flow is not taken into account. The purpose of 3D gates is to represent small objects, e.g. floating or submerged dams or structures. 3D gates can be defined at multiples of 45 degree angles with the grid directions and they must be defined inside the grid enclosure.

File contents	Definition of impermeable sheets blocking the flow in N- or M-direction and extending over one or more layers in the vertical.
File format	Free formatted.
Generated	Manually offline.

Remarks:

- ◊ The keyword is case sensitive!
- ◊ Value *must* be enclosed between two number signs (#).



Record description:

Record	Record description				
each record	<p>Direction in which the flow is blocked (1 character):</p> <table style="margin-left: 40px;"> <tr><td>U</td><td>U-gate</td></tr> <tr><td>V</td><td>V-gate</td></tr> </table> <p>Begin and end indices of the gate, i.e. m1, n1, m2, n2 (4 integers) Layer numbers over which the gate extends, i.e. k1 and k2 (2 integers).</p>	U	U-gate	V	V-gate
U	U-gate				
V	V-gate				

**Restrictions:**

- ◊ The 3D gate type must be given in position one.
- ◊ A multiple gate can be defined along a line segment. The angle of the line segment and the M-direction must be an integer multiple of 45 degrees.
- ◊ Gates may not be defined along the model boundaries (which by default lie along the lines m = 1, n = 1, m = Mmax or n = Nmax). Therefore, the indices of these points must lie between m = 2 and Mmax-1 and n = 2 and Nmax-1, respectively.
- ◊ Input items in a record must be separated by one or more blanks.
- ◊ You *cannot* combine 3D gates with other structures, such as rigid sheets, porous plates or local weirs at the same grid location (m, n).
- ◊ However, gates defined at the same location but extending over different layers can be combined (you can have a gate at the same location in layer 1 and in layers 3 to 5).

Example:

Four 3D gates are defined, see [Figure B.1](#).

Three V-gates extending from:

- ◊ (5, 3) to (7, 3) over layer 4,
- ◊ (5, 4) to (7, 4) over layer 4 and 5,
- ◊ (5, 5) to (7, 5) over layer 4 and 5.

One U-gate extending from:

- ◊ (4, 4) to (5, 4) over layer 4 and 5.

```

V 5   3   7   3   4   4
V 5   4   7   4   4   5
V 5   5   7   5   4   5
U 4   4   4   5   4   5

```

B.3.2 Quadratic friction

B.3.2.1 Barrier

Keyword	Value	Description	Default
Filbar	#name.bar#	Filename for barrier data	none

A barrier is represented by a vertical gate from the water level down to a certain level in the flow along one of the grid directions. The gate blocks the flow in the top grid cells. The number

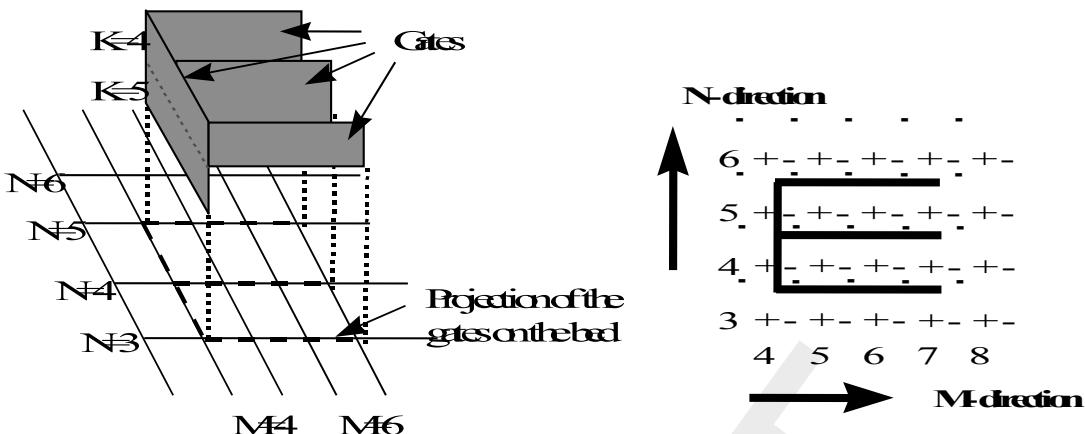


Figure B.1: Example of 3D gates in perspective view (left) and top view (right)

of layers that are blocked will vary with water elevation and the gate level. Partial blocking is accounted for.

File contents	The description of barrier locations and characteristics.
Filetype	ASCII
File format	Fixed formatted for text variables, free formatted for integer values.
Filename	<name.bar>
Generated	manually offline

Record description:

Record	Record description
each record	Name of the barrier (20 characters), barrier orientation (U or V), grid indices of the begin and end point (4 integers), bottom level of gate (1 real, positive up), barrier loss type (a or b), energy loss coefficients (1 or 2 reals).

Restrictions:

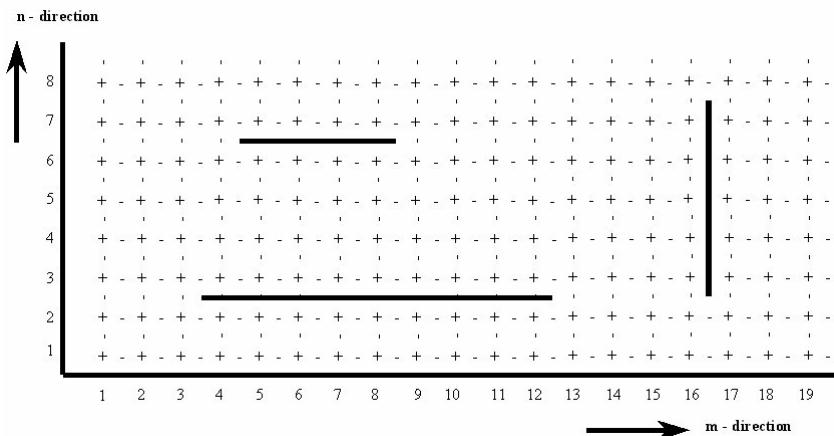
- ◊ One record per barrier.
- ◊ The barrier's name must start in position one.
- ◊ Maximum record length is 132.
- ◊ Name, barrier orientation and other parameters must be separated by one or more blanks.



Remarks:

- ◊ The bottom level of the gate specified in this file may be overruled during the computation by means of the real-time control module. See [B.28](#)
- ◊ For barrier loss type 'a', you have to specify one energy loss coefficient, namely c_{loss} .
- ◊ For barrier loss type 'b', you have to specify two energy loss coefficients, namely α and β .
- ◊ Old barrier records imply the use of barrier loss type 'a'. Such old records contain the coefficients in the following order: name of the barrier (20 characters), barrier orientation (U or V), grid indices of the begin and end point (4 integers), energy loss coefficient c_{loss} (1 real), bottom level of gate (1 real, positive up).



**Figure B.2:** Barriers in model area**Example:**

File containing 3 barriers for the configuration shown in [Figure B.2](#). The bottom levels of the gates vary between 2.0 and 4.0. All three barriers use barrier loss type 'a'. The first barrier has an energy loss coefficient c_{loss} of 0.9, while the other two have been assigned a loss coefficient of 0.8.

Barrier North	V 5	6	8	6	4.0	a	0.9
Barrier East	U 16	3	16	7	2.0	a	0.8
Barrier South	V 4	2	12	2	3.0	a	0.8

B.3.2.2 Bridge

The effect of a bridge can be schematized using porous plates. See [subsection B.3.3](#).

B.3.2.3 Current deflection wall (CDW)

Keyword	Value	Description	Default
Filcdw	#name.cdw#	Filename for CDW data	none

A CDW is represented by a vertical plate in the flow along one of the grid directions with fixed dimensions in the vertical. The flow across the plate is blocked. In σ grid the number of layers that are blocked will vary with water elevation. Partial blocking is accounted for. The plate is supported by pylons, [Figure B.3](#).

There is no momentum exchange across the plate and friction exerted on the flow by the supportive pylons is taken into account. Current Deflecting Wall is used to deflect some of the flow that enters into a harbour to enhance flow exchange and thereby to reduce siltation in the harbour. CDW must be defined inside the grid enclosure.

File contents	Definition of impermeable sheet blocking the flow in N- or M-direction and its co-ordinates in the vertical. This information is followed by friction coefficient for the supportive pylons.
File format	Free formatted.
Generated	Manually offline.

**Remarks:**

- ◊ The keyword `Filcdw` is case sensitive!
- ◊ Value of `Filcdw` *must* be enclosed between two number signs (#).

Record description:

Record	Record description				
each record	<p>Direction in which the flow is blocked (1 character):</p> <table style="margin-left: 40px; margin-top: 10px;"> <tr><td>U</td><td>U-gate</td></tr> <tr><td>V</td><td>V-gate</td></tr> </table> <p>Begin and end indices of the gate, i.e. m1, n1, m2, n2 (4 integers) Friction coefficient to account for the effect of the supportive pylons (1 real) z-co-ordinates over which the gate extends, i.e. zbot and ztop (2 reals).</p>	U	U-gate	V	V-gate
U	U-gate				
V	V-gate				

Restrictions:

- ◊ The CDW structure blocking the velocity must be given in position one.
- ◊ A single CDW structure is allowed along a line segment. The angle of the line segment and the M-direction must be an integer multiple of 45 degrees.
- ◊ Gates may not be defined along the model boundaries (which by default lie along the lines $m = 1, n = 1, m = M_{\text{max}}$ or $n = N_{\text{max}}$). Therefore, the indices of these points must lie between $m = 2$ and $M_{\text{max}} - 1$ and $n = 2$ and $N_{\text{max}} - 1$, respectively.
- ◊ Input items in a record must be separated by one or more blanks.
- ◊ You *can not* combine CDW with other structures, such as rigid sheets, porous plates or local weirs at the same grid location (m, n).
- ◊ The friction coefficient must be larger than 0.
- ◊ Vertical position of the CDW plates (bottom and top) are specified in the Cartesian coordinate in the z-direction (positive up relative to reference plane).
- ◊ z co-ordinate at the top must be $>$ z co-ordinate at the bottom.
- ◊ z co-ordinate at the top may be arbitrarily high (e.g. 999.99 to ensure that the plate is always above water).

Example:

This example defines a CDW structure blocking the V-velocity (i.e. structure aligned with flow in U-direction):

- ◊ (5, 3) to (7, 3) at the vertical position between $z = -1.5$ m and $z = 999.99$ m.

CDW is supported by pylons introducing friction with friction coefficient of 0.116

V 5 3 7 3 0.116 -1.5 999.99

B.3.2.4 Weirs (2D model)

Keyword	Value	Description	Default
<code>Fil2dw</code>	<code>#name.2dw#</code>	Filename for data spillways, weirs and summer dikes	none
<code>ThetaW</code>	[0,1]	Relaxation parameter	0

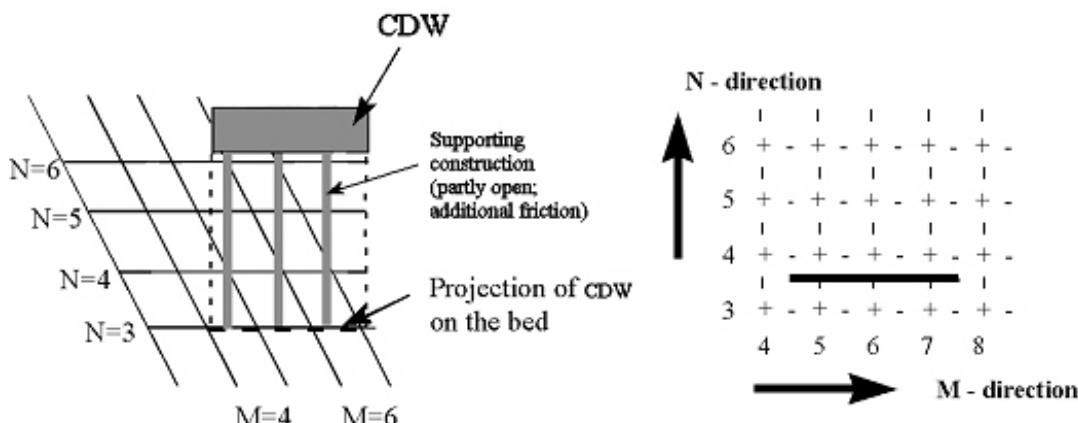


Figure B.3: Example of CDW in perspective view (left) and top view (right)

A 2D weir can only be used in a 2D model. (not yet implemented for the Z-model).

A 2D weir is modelled as a *sub-grid* phenomenon, i.e. its dimensions are (assumed to be) much smaller than the grid size and only its overall influence on the flow is taken into account. A 2D weir is characterised by its location and its crest height. Contrary to local weirs, the 2D weirs are based on an empirical estimation of the energy loss over the weir in both the case of a perfect weir and an imperfect weir. This energy loss is converted into an effective friction coefficient, which may be adapted using a calibration coefficient; see [subsection 10.9.2](#) for full details. For a perfect weir the energy term controls the discharge to the theoretical discharge for supercritical flow given the the energy head upstream of the weir.



Remark:

- ◊ The option `Waqopt=#YY#` to change the discretisation has been replaced by `Momsol = #WAQUA#`.

File contents	Location, dimensions and energy losses of spillways, weirs and summer dikes.
File format	Free formatted.
Filename	User-defined.
Generated	Manually offline.



Remarks:

- ◊ The keyword is case sensitive!
- ◊ Values of character-type *must* be enclosed between two number signs (#).



Restrictions:

- ◊ One 2D weir per record.
- ◊ The 2D weir type must be given in position one of each record.
- ◊ 2D weirs may not be defined along the model boundaries (which by default lie along the lines $M = 1$, $N = 1$, $M = M_{max}$ or $N = N_{max}$). Therefore, the indices of these points must lie between $M = 2$ and $M_{max}-1$ and $N = 2$ and $N_{max}-1$, respectively.
- ◊ The friction calibration coefficient must have a positive value.
- ◊ Input items in a record must be separated by one or more blanks.

Record description:

Record	Record description				
each record	<p>Direction of the 2D weir normal to the flow (1 character):</p> <table style="margin-left: 40px;"> <tr><td>U</td><td>U-weir</td></tr> <tr><td>V</td><td>V-weir</td></tr> </table> <p>Begin and end indices of the 2D weir, i.e. m1, n1, m2, n2 (4 integers), Friction calibration coefficient normally equal to 1.0 [-] (1 real), Crest height [m] measured positive downward (1 real), Real number; not yet used (1 real).</p>	U	U-weir	V	V-weir
U	U-weir				
V	V-weir				

Remark:

- ◊ The definition of the weir crest is the same as for the bottom depth: the depth below the horizontal plane of reference (*positive downward*).

**Example:**

The file contains three 2D weirs:

- ◊ The first weir is a ‘U-weir’, i.e. influencing the flow in U-direction at m = 6, from n = 4 to n = 6. The calibration coefficient is 1.0 and the crest height is 12.0 metres below the reference plane.
- ◊ The second weir is a ‘V-weir’, at m = 14 to m = 16 at n = 3. The calibration coefficient is 1.0 and the crest height is 11.0 metres below the reference plane.
- ◊ The third weir is defined along a 45 degrees line and consists of four small ‘U-weirs’ each. The calibration coefficient is 1.0 and the crest height is 9.0 metres below the reference plane.

See [Figure B.4](#) for a top view of the location in the grid.

```

U   6   4   6   6   1.   12.   0.
V   14  3   16  3   1.   11.   0.
U   7   4   10  7   1.   9.    0.

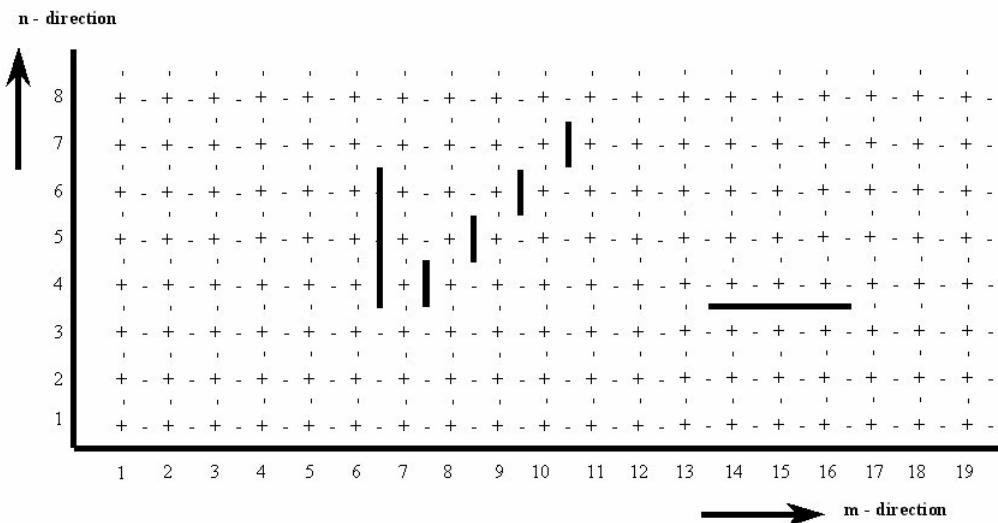
```

B.3.2.5 Local weir

Keyword	Value	Description	Default
Fillwl	#name.lwl#	Filename for local weir data	none
Upwlwl	#Y# or #N#	Upwind advection scheme near structure	#Y#

A local weir may be used in a 3D model. It is assumed that the friction coefficient is the same for all the computational layers.

A local weir is modelled as a *sub-grid* phenomenon, i.e. its dimensions are (assumed to be) much smaller than the grid size and only its overall influence on the flow is taken into account. This overall influence is modelled by the energy losses due to the weir. These energy losses are described as an additional quadratic friction term in the momentum equations; see Sections [9.10.2](#) and [10.9.2](#) for full details. A local weir can be defined at multiples of 45 degree angles with the grid directions and they must be defined inside the grid enclosure, [Figure B.5](#).

**Figure B.4:** Top view of 2D weirs

With the keyword Upwlwl you can specify to use an upwind approximation in the advective terms of the momentum equations at the location of the structure. You can use this option to prevent (damp) oscillations that may occur due to large gradients.

File contents	Location, dimensions and energy losses of local weirs.
File format	Free formatted.
Generated	Manually offline.

**Remarks:**

- ◊ The keyword is case sensitive!
- ◊ Values of character-type *must* be enclosed between two number signs (#).

Record description:

Record	Record description
each record	Direction of the local weir normal to the flow (1 character): U U-weir V V-weir Begin and end indices of the local weir, i.e. m1, n1, m2, n2 (4 integers), Friction coefficient of the quadratic friction [-] (1 real), Crest height [m] measured positive downward (1 real), Real number; not yet used (1 real).

**Remark:**

- ◊ The definition (reference system) of the weir crest is the same as for the bottom depth, so below the horizontal plane of reference (*positive downward*).

**Restrictions:**

- ◊ The crest height is ignored when the local water depth at the local weir (averaged water depth on both sides of the local weir) is smaller than the crest height specified.
- ◊ One local weir per record.
- ◊ The local weir type must be given in position one of each record.
- ◊ Local weirs can be defined along a line segment at an angle with the M-direction of an

integer multiple of 45 degrees.

- ◊ Local weirs may not be defined along the model boundaries (which by default lie along the lines $m = 1, n = 1, m = M_{\max}$ or $n = N_{\max}$). Therefore, the indices of these points must lie between $m = 2$ and $M_{\max}-1$ and $n = 2$ and $N_{\max}-1$, respectively.
- ◊ The friction coefficient must have a positive value.
- ◊ Input items in a record must be separated by one or more blanks.
- ◊ You *can not* combine weirs with other structures, such as 3D gates, porous plates or rigid sheets at the same grid location (m, n).

Example:

A local weir is defined at $m = 5, n = 2$. The quadratic friction coefficient is 0.5 and the crest height is 12.0 metres below the reference plane, see [Figure B.5](#).

U 5 2 5 2 .5 12. 1.

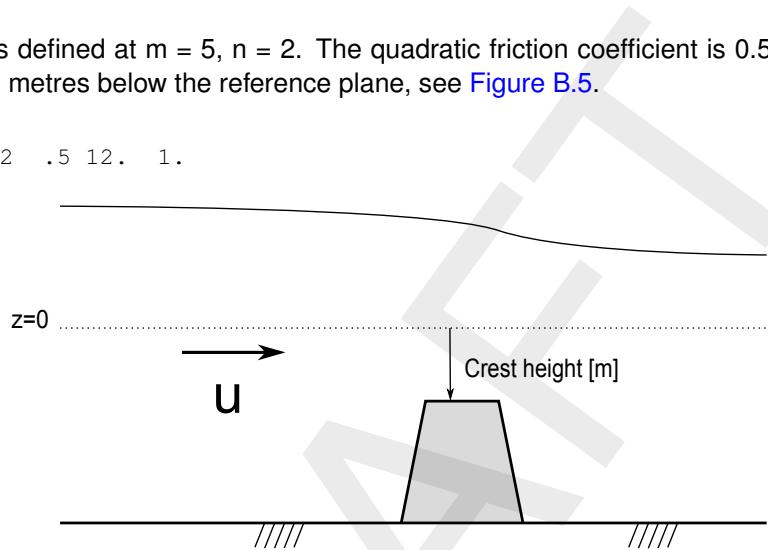


Figure B.5: Local weir

B.3.3 Porous plate

Keyword	Value	Description	Default
Filpppl	#name.ppl#	Filename for porous plate data	none
Upwppl	#Y# or #N#	Upwind advection scheme near structure	#Y#

A porous plate is a partially transparent structure that extends into the flow along one of the grid directions, that covers some or all layers in the vertical, but whose thickness is much smaller than the grid size in the direction normal to the porous plate. Due to its partial transparency mass and momentum is exchanged across the porous plate. A porous plate can be used to represent objects that are not totally oblique to the flow and are thin relative to the grid size such as partly transparent breakwaters. The porosity of the plate is controlled by a quadratic friction term; the friction is an input parameter, see Sections [9.10.2](#) and [10.9.2](#) for full details. Porous plates can only be defined at multiples of 45 degree angles with the grid directions and they must be defined inside the grid enclosure.

With the keyword Upwppl you can specify to use an upwind approximation in the advective terms of the momentum equations at the location of the structure. You can use this option to prevent (damp) oscillations that may occur due to large gradients.

File contents

Definition of the location of a porous plate extending over one or more layers of the water column with a quadratic friction to simulate

File format	the energy losses.
Generated	Free formatted.
	Manually offline.

**Remarks:**

- ◊ The keyword is case sensitive!
- ◊ Values of character-type *must* be enclosed between two number signs (#).

Record description:

Record	Record description				
each record	<p>Direction of the porous plate normal to the flow (1 character):</p> <table style="margin-left: 40px; margin-top: 10px;"> <tr> <td>U</td> <td>U-porous plate</td> </tr> <tr> <td>V</td> <td>V-porous plate</td> </tr> </table> <p>Begin and end indices, i.e. m1, n1, m, n2 (4 integers), Layers over which the porous plate extend, i.e. k1 and k2 (2 integers), Friction coefficient of the quadratic friction [-] (1 real).</p>	U	U-porous plate	V	V-porous plate
U	U-porous plate				
V	V-porous plate				

**Restrictions:**

- ◊ One record per porous plate.
- ◊ A porous plate can be defined along a line segment at an angle of an integer multiple of 45 degrees with the M-direction.
- ◊ A porous plate may not be defined along the model boundaries (which by default lie along the lines m = 1, n = 1, m = Mmax or n = Nmax). Therefore, the indices of these points must lie between m = 2 and m = Mmax - 1 and n = 2 and n = Nmax - 1, respectively.
- ◊ The friction coefficient must be positive.
- ◊ Input items in a record must be separated by one or more blanks.
- ◊ You *can not* combine porous plates with other structures, such as 3D gates, rigid sheets or local weirs at the same grid location (m, n).

Example:

Porous plates are defined at the following points:

- ◊ A U-direction porous plate from (16, 2) to (16, 4) extending over seven layers,
- ◊ A V-direction porous plate from (4, 1) to (4, 6) extending over five layers,
- ◊ A U-direction porous plate from (7, 14) to (7, 20) extending over three layers.

```
U 16   2   16   4   1   7   1000.
V  4    1    4    6   1   5   1000.
U  7   14   7   20   3   5    800.
```

B.3.4 Culvert

Presently there are 4 types of discharges:

- ◊ Normal
- ◊ Momentum
- ◊ Walking
- ◊ In-out

A culvert is a special intake/outlet coupling in which the discharge rate depends on the flow regime. The location of the culvert are defined in the `<*.src>` file. The characteristics of the culverts are defined in the new file `<*.cul>`. The keyword `Filcul = #name.cul#` has to be added to the MDF-file. A dummy discharge (series) has to be specified in the discharge data `<*.dis>` file for culverts, but such value(s) will *not* be used in the computation. For a culvert the discharge is computed by Delft3D-FLOW itself, this means that the discharge values in the `<*.dis>` file have no meaning. You are advised to prescribe dummy values in this file (e.g. -999.99).

The other two files are described now.

B.3.4.1 Definition of culvert in the discharge input file (`<*.src>`)

The attribute file for discharges, the so-called `<*.src>` file, has been extended with an identification mark and the locations of the intake/outlet of the culvert. Culverts are identified by 'c', 'd', 'e' , 'f' or 'u' (case insensitive) and the M, N, K indices of the coupled locations. For a one-way culvert (types 'c' and 'e') the flow is always from the intake location to the outlet location. For a two-way culvert (type 'd' and 'f') flow in both directions is possible; for type 'd', the direction is from high to low water level, for type 'f', the direction is from high to low pressure, taking density differences into account. For the user-defined culvert/structure (type 'u') the flow may be either one or both directions depending on the requirements of the user.

In case of a one-way culvert, the first (M, N, K) pair in the `<*.src>` file defines the intake location and the second (M, N, K) pair defines the outlet location. Flow through the culvert is only possible from the intake to the outlet, which is the case if the water elevation at the intake is higher than at the outlet location. If the water elevation at the outlet location is higher than at the intake, then there is no flow. Furthermore, if the intake location is set dry, then there is no flow through the culvert.

Below an example is given of a `<*.src>` file. The first discharge represents a normal discharge, the second one a walking discharge and the third one a so-called power station (or intake/outlet combination). The latter four correspond to four types of culverts, respectively. A line for culverts of type 'u' is the same except for the type character.

Example of `<*.src>` file

Discharge 1	Y	2	2	1				
Discharge 2	Y	2	11	5	w			
Power station 1	Y	8	5	3	p	2	4	9
Culvert 1	Y	2	4	2	c	2	7	2
Culvert 2	Y	5	4	0	D	6	7	0
Culvert 3	Y	7	7	0	e	6	6	0
Culvert 4	Y	3	3	0	f	4	4	0

The first 20 positions of each record are reserved for the name of the culvert.



Remarks:

- ◊ In case of a domain decomposition simulation, the intake and outlet locations of a power station and of culverts should be in the same domain.
- ◊ For a culvert of type 'd', 'e', 'f', the layer numbers (fourth and seventh entry in the <*.src> file) have no meaning. For type 'c' the layer number must be specified. 0 is interpreted as equally devided over the full water column. For a culvert of type 'd', 'e', 'f' also the vertical position has to be specified (see description of file <*name.cul*>) and the layer numbers of the intake and outlet are determined by Delft3D-FLOW. Note that in case of a power station the layer numbers should be specified on input.

**Restriction:**

- ◊ The locations of the intake and outlet must be different, otherwise an error will be generated by the program.

B.3.4.2 Properties for culverts defined in INI file (<*name.cul*>)

The following line has to be added to the MDF-file

```
Filcul= #name.cul#
```

With <*name.cul*> the file with the culverts characteristics.

Depending on the type 'c', 'd', 'e', 'f' or 'u' additional properties of the culvert need to be specified. These properties are specified by a file with name <*name.cul*> in the directory of the corresponding MDF-file.

This file is of the INI file format. It starts with one so-called chapter [CulvertFileInformation] with three so-called key-value pairs (i.e. a key with a specification of its value). From these key-value pairs only the specification of the key FileVersion and its value are necessary.

Example

```
[CulvertFileInformation]
  FileCreatedBy      = Delft3D user
  FileCreationDate  = 12:32:06, 26-04-2004
  FileVersion        = 01.02
```

After the chapter [CulvertFileInformation] and its key-values pairs, for each culvert there should be one separate chapter [Culvert]. For each culvert this chapter needs a key Name for which the corresponding value specifies the name of the culvert. This name must match exactly the name in the <*.src> file.

Example

```
[Culvert]
  Name = Culvert 1
```

Depending on the type 'c', 'd', 'e', 'f' or 'u', each chapter [Culvert] contains additional key-value pairs. These key-values are described separately for culverts of type 'c', for culverts of type 'd' or 'e', for culverts of type 'f' and for culverts of type 'u'.

B.3.4.3 Additional key-value pairs for culvert of type ‘c’

For a culvert of type ‘c’ the area of the culvert opening A (in m^2) and the culvert loss coefficient (μ , dimensionless) should be specified, see Eq. 10.89.

These properties A and μ of the culvert are specified by the keys `WetArea` and `LossCoefficient` respectively.

Example

```
[CulvertFileInfo]
  FileCreatedBy      = Delft3D user
  FileCreationDate  = 12:32:06, 26-04-2004
  FileVersion       = 01.02
[Culvert]
  Name              = Culvert 1
  LossCoefficient   = 0.9
  WetArea           = 4.0
```

B.3.4.4 Additional key-value pairs for culvert of type ‘d’ or ‘e’

Since for culverts of type ‘d’ or ‘e’ the culvert loss coefficient is now dependent on the wet section of the culvert, a table has to be specified that relates the culvert loss coefficient and the wet section of the culvert. Furthermore, for each culvert of type ‘d’ or ‘e’, three coefficients have to be specified, for flow types (1/2/3), (4/6) and 5, respectively.

Because of this, for culvert of type ‘d’ or ‘e’ the following key-value pairs need to be specified:

- ◊ Height with the value of H , the height of the culvert [m],
- ◊ Width with the value of W , the width of the culvert [m],
- ◊ VerticalPosition with the value of $z_{culvert}$, the vertical position of the bottom of the culvert relative to the reference level (positive upwards) [m],
- ◊ Length with the value of L , the length of the culvert [m],
- ◊ FrictionCoefficient with the value of n , the energy loss coefficient of the culvert (Manning’s coefficient [$\text{s}/\text{m}^{1/3}$]),
- ◊ CorrectionCoefficient with the value of a , the energy loss correction coefficient of the culvert [-],
- ◊ NumberofRelations1, NumberofRelations2, and NumberofRelations3 with the number of relations as an integer value (i.e. the number of entries of the table) between the culvert loss coefficient and the wet section of the culvert for flow types (1/2/3), (4/6) and 5, respectively, and
- ◊ Corresponding to the keys NumberofRelations1, NumberofRelations2, and NumberofRelations3 the keys LossCoefficient1, LossCoefficient2, and LossCoefficient3, respectively, for the culvert loss coefficient and the keys WetArea1, WetArea2, and WetArea3, respectively, for the wet section of the culvert. The values of these keys define the table that relates the culvert loss coefficient and the wet section of the culvert. Note that the number of relations as specified by the value of the key, say NumberofRelations1, should correspond with the number of values (all values on one line, each value separated by a space) of the key LossCoefficient1 and the key WetArea1.

Example

```
[CulvertFileInfo]
  FileCreatedBy      = Delft3D user
  FileCreationDate  = 12:32:06, 26-04-2004
```

```

FileVersion      = 01.02
[Culvert]
  Name          = Culvert 2
  Height        = 1.0
  Width         = 1.0
  VerticalPosition = -4.0
  Length        = 20.0
  FrictionCoefficient = 0.015
  CorrectionCoefficient = 0.03
  NumberOfRelations1 = 2
  LossCoefficient1 = 0.6 0.7
  WetAreal       = 10.0 13.0
  NumberOfRelations2 = 2
  LossCoefficient2 = 0.6 0.7
  WetArea2       = 10.0 13.0
  NumberOfRelations3 = 2
  LossCoefficient3 = 0.6 0.7
  WetArea3       = 10.0 13.0

```



Remarks:

- ◊ For each culvert c_D^1 , c_D^2 and c_D^3 have to be specified, respectively.
- ◊ The energy loss correction coefficient α may be different for each culvert, but is independent of the flow condition.
- ◊ It is assumed that the wet areas are specified in ascending order.
- ◊ If the actual wet area is smaller than the first value specified in the table, then the first loss coefficient in the table is used.
- ◊ Similarly, if the actual wet area is larger than the last value specified in the table, then the last loss coefficient in the table is used.
- ◊ In between linear interpolation is applied.
- ◊ Maximally ten pairs of culvert loss coefficients and wet areas can be specified.

B.3.4.5 Additional key-value pairs for culvert of type ‘f’

For a culvert of type ‘f’ the following key-value pairs need to be specified:

- ◊ Height with the value of H , the height of the culvert [m],
- ◊ Width with the value of W , the width of the culvert [m],
- ◊ VerticalPosition with the value of $z_{culvert}$, the vertical position of the bottom of the culvert relative to the reference level (positive upwards) [m],
- ◊ LossCoefficient with the value of μ [-].

Example

```

[CulvertFileInformation]
  FileCreatedBy    = Delft3D user
  FileCreationDate = 12:32:06, 05-04-2013
  FileVersion      = 01.02
[Culvert]
  Name          = Culvert 4
  LossCoefficient = 0.9
  Height        = 0.25
  Width         = 0.25
  VerticalPosition = -4.0

```

B.3.4.6 Additional key-value pairs for culvert of type ‘u’

The discharge through a culvert of type ‘u’ is not computed using a predefined formula, but it is determined by a routine that you provide yourself in a shared library. Hence the keywords required for culverts of type ‘u’ differ significantly from those required for the culverts

of type 'c', 'd' and 'e' mentioned above. The interface for the routine is described in detail in subsection B.3.8.

The following key-value pairs need to be specified:

- ◊ CulvertLib Name of the shared object (.dll on Windows, or .so on Linux) without file extension that contains the routine CulvertFunction,
- ◊ CulvertFunction Name of the routine that should be used to compute the discharge through the culvert/structure,
- ◊ CulvertFile This key can optionally be used to point to an input file with parameter settings for the user routine CulvertFunction. See subsection B.3.8 for a description of the subroutine interface.

Example

```
[CulvertFileInfo]
  FileCreatedBy      = Delft3D user
  FileCreationDate  = 12:32:06, 26-04-2004
  FileVersion        = 01.02
[Culvert]
  Name              = Culvert 4
  CulvertLib        = plugin_culvert
  CulvertFunction   = TABLE
  CulvertFile       = culvert_params.txt
```

B.3.4.7 More culverts

In case of more culverts, which can be of different type, the additional properties are specified in one INI file (<name.cul>).

Example

```
[CulvertFileInfo]
  FileCreatedBy      = Delft3D user
  FileCreationDate  = 12:32:06, 26-04-2004
  FileVersion        = 01.02
[Culvert]
  Name              = Culvert 2
  Height            = 1.0
  Width             = 1.0
  VerticalPosition = -4.0
  Length            = 20.0
  FrictionCoefficient = 0.015
  CorrectionCoefficient = 0.03
  NumberOfRelations1 = 3
  LossCoefficient1 = 0.6 0.65 0.7
  WetAreal          = 0.0 0.4 1.0
  NumberOfRelations2 = 3
  LossCoefficient2 = 0.7 0.75 0.8
  WetArea2          = 0.0 0.6 1.0
  NumberOfRelations3 = 3
  LossCoefficient3 = 0.8 0.85 0.9
  WetArea3          = 0.0 0.5 1.0
[Culvert]
  Name              = Culvert 1
  LossCoefficient   = 0.9
  WetArea           = 4.0
```

B.3.5 Linear friction

B.3.5.1 Rigid sheet

Keyword	Value	Description	Default
Filrgs	#name.rgs#	Filename for rigid sheet data	none
Upwrgs	#Y# or #N#	Upwind advection scheme near structure	#Y#

A rigid sheet is a plate that extends into the flow along one of the grid lines and that covers one or more layers in the vertical. The (linear) friction, see Sections 9.10.3 and 10.9.3 for details, exerted by the plate on the flow can be taken into account. You can specify a line of rigid sheets along grid lines or along lines at multiples of 45 degrees.

With the keyword Upwrgs you can specify to use an upwind approximation in the advective terms of the momentum equations at the location of the rigid sheet. You can use this option to prevent (damp) oscillations that may occur due to large gradients.

File contents	The location and extent of rigid sheets and the additional (linear) friction coefficient.
File format	Free formatted, but starting in position one of each record.
Generated	Manually offline.



Remarks:

- ◊ The keyword is case sensitive!
- ◊ Character values *must* be enclosed between two number signs (#).
- ◊ There is no other limitation to Value.
- ◊ A rigid sheet can also be used in a 2DH model.

File description:

Record	Record description				
each record	<p>Direction of the flow blocked by the rigid sheets (1 character):</p> <table style="margin-left: 40px; margin-top: 10px;"> <tr> <td>U</td> <td>U-weir</td> </tr> <tr> <td>V</td> <td>V-weir</td> </tr> </table> <p>Begin and end indices of the rigid sheet in m- and n-direction, i.e. m1, n1, m2, n2 (4 integers),</p> <p>Extension in the vertical, i.e. k1 and k2 (2 integers).</p> <p>Friction coefficient of the rigid sheet [m/s] (1 real).</p>	U	U-weir	V	V-weir
U	U-weir				
V	V-weir				



Restrictions:

- ◊ One (set of) rigid sheet(s) per record.
- ◊ The angle of the line of rigid sheets and the m-direction must be an integer multiple of 45 degrees (in the computational space).
- ◊ Rigid sheets can not be defined along the model boundaries and should not cross the grid enclosure.
- ◊ The friction coefficient must be positive.
- ◊ Input items on a record must be separated by one or more blanks.
- ◊ You *can not* combine rigid sheets with other structures, such as 3D gates, porous plates or local weirs at the same grid (m, n) location.

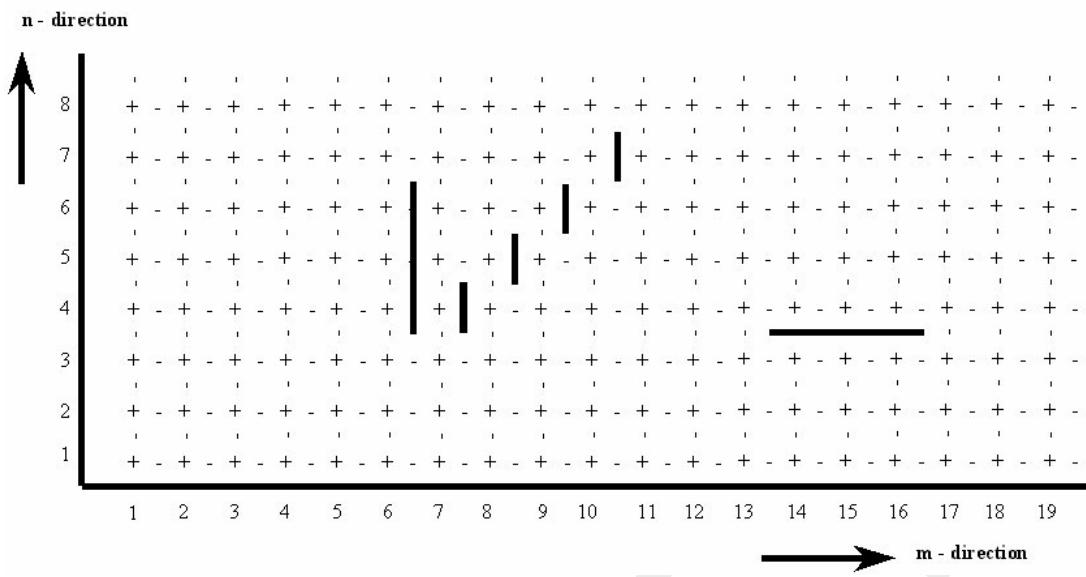


Figure B.6: Top view of rigid sheets

Example:

The file contains three rigid sheets:

- ◊ The first sheet is a ‘U-sheet’, i.e. blocking the flow in U-direction at $m = 6$, from $n = 4$ to $n = 6$, over the first 7 layers.
- ◊ The second rigid sheet is a ‘V-sheet’, at $m = 14$ to $m = 16$ at $n = 3$, extending over 3 grid cells, and extending over the first 5 layers in the vertical.
- ◊ The third rigid sheet is defined along a 45 degrees line and consists of four small sheets each covering one grid cell and layers 2 to 6.

See [Figure B.6](#) for a top view of the location in the grid.

```

U   6   4   6   6   1   7   1000.
V   14  3   16  3   1   5   1000.
U   7   4   10  7   2   6   800.

```

B.3.6 Floating structure

Keyword	Value	Description	Default
Filfls	#name.flst#	Filename for a floating structure data	none
Riglid	1 real number	Compressibility of the flow beneath the structure	1.0
Upwfls	#Y# or #N#	Upwind advection scheme near structure	#Y#

Floating structures are characterised by their horizontal extent, the depth they penetrate the surface and their ability to react on the local upward pressure, [Figure B.7](#). The floating structure is modelled as a locally applied surface pressure. The friction exerted by the structure on the flow is not taken into account. With the keyword Riglid you can specify the flow beneath the flow structure to be treated as incompressible, see [subsection 9.3.1](#) for details.

With the keyword `Upwf1s` you can specify to use an upwind approximation in the advective terms of the momentum equations at the location of the floating structure. You can use this option to prevent (damp) oscillations that may occur due to large gradients.

In Delft3D-FLOW version 3.07.00 and higher the following processes are switched off at the locations of a floating structure:

- ◊ Evaporation.
- ◊ Wind (both the wind stress and the wind speed itself, used in the Internal Wave Energy model, are set to zero).
- ◊ Wave influences (wave stresses, wave period, wave angle, mass fluxes due to waves, dissipation due to waves and the orbital velocity in the bottom layer).

Unchanged are the following processes:

- ◊ Precipitation (is drained through the floating structure).
- ◊ Turbulence formulation in the vertical.
- ◊ No friction on the flow by the floating structure.

File contents	Thickness of the floating structure (in metres) defined in water level points for all grid points in the top layers.
File format	Free formatted.
Generated	QUICKIN or manually offline.



Remarks:

- ◊ The keyword is case sensitive!
- ◊ Value *must* be enclosed between two number signs (#).
- ◊ Floating structures are located at water level points (centre of a grid cell).

Record description:

Record	Record description
1 to p	Thickness of the floating structure in metres for all grid points in the top layer starting for n = 1 and m = 1 to Mmax, with as many values per record as allowed by the maximal record length (132) and as many continuation records (p) as necessary to specify all Mmax points.
p+1 to p · Nmax	Repeating the set of p-records for N = 2 to Nmax.



Restrictions:

- ◊ Maximum record length is 132.
- ◊ Values from the file will not be checked against their domain.
- ◊ Where no floating structure is present the thickness is specified as zero.
- ◊ Input items in a record must be separated by one or more blanks.
- ◊ You *can not* combine floating structures with other structures, such as 3D gates, porous plates or local weirs at the same grid (m, n) location.
- ◊ Floating structures cannot be defined at M = 1, M = Mmax or N = 1, N = Nmax.

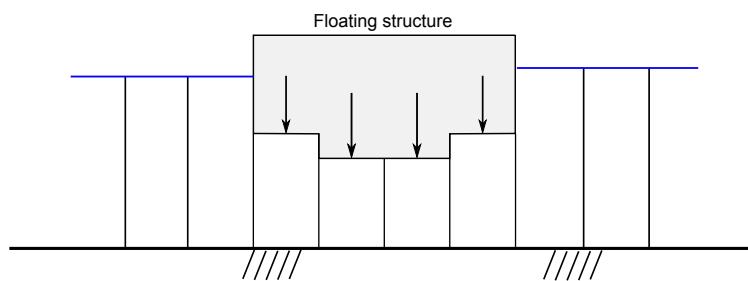


Figure B.7: Cross-sectional view floating structure
The vertical lines are drawn through the velocity points

Example:

For a model area of $14 * 7$ grid points. The structure has a rectangular shape and extends over three grid cells in N-direction ($n = 3$ to 5) and 5 grid cells in M-direction ($n = 7$ to 11) with a varying depth ranging from 1 to 3 m. An example of a cross-sectional view is given in Figure B.7.

```
-999 -999 -999 -999 -999 -999 -999 -999 -999 -999 -999 -999 -999 -999 {N=1}
-999 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -999 {N=2}
-999 0.0 0.0 0.0 0.0 0.0 2.0 3.0 3.0 2.5 1.0 0.0 0.0 -999 {N=3}
-999 0.0 0.0 0.0 0.0 0.0 2.0 3.0 3.0 2.5 1.0 0.0 0.0 -999 {N=4}
-999 0.0 0.0 0.0 0.0 0.0 2.0 3.0 3.0 2.5 1.0 0.0 0.0 -999 {N=5}
-999 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -999 {N=6}
-999 -999 -999 -999 -999 -999 -999 -999 -999 -999 -999 -999 -999 -999 {N=7}
```

B.3.7 Upwind at Discharges

Keyword	Value	Description	Default
Upwsrc	#Y#, #N# or #Momentum only#	Upwind advection scheme near discharge locations	#Momentum only#

With the keyword `Upwsrc` you can specify to use an upwind approximation in the advective terms of the momentum equations at the location of discharges. The upwind approximation can be used either for all discharges, no discharges at all, or only momentum discharges. You can use this option to prevent (damp) oscillations that may occur due to large gradients.

Remarks:

- ◊ The keyword is case sensitive!
- ◊ Values of character-type *must* be enclosed between two number signs (#).



B.3.8 User defined discharge through a structure

This section gives an example code for a user-defined routine that computes the discharge through a structure (the feature can be enabled by using a culvert of type 'u', see [subsection B.3.4](#)). This particular example implements two routines: the formula for a culvert of type 'd' and a simple table specifying discharge as a function of the water level difference.

The order and way in which arguments are passed to external functions differs among FORTRAN compilers. Most compilers allow you to change these settings, however, if these settings do not match the settings of the calling program the program will crash when trying to call the subroutine in the supplied library. It is, therefore, important to make sure that the

settings that you use to compile the library match the settings used when compiling delftflow. It therefore recommended to generate the library using the same compiler as delftflow; at this moment this is the Intel Fortran 11 compiler¹ using default settings for a multithreaded program.

Most of the code is straightforward and standard Fortran, so it should be fairly easy to understand. Some statements may, however, require some explanation. The `!DEC$` statement is read by the compiler and makes sure the subroutine name is visible to the outside world. The real type “hp” is defined as a double precision quantity: even if the simulation runs in single precision, the communication with the library occurs in double precision hence all input and output arguments of the subroutine should be defined to be of the type “real(hp)”.

The input quantities are provided using three linear arrays, namely a real array, an integer array and a character array. Each element of these arrays represents a different quantity; see the example for a description of the various input items. Input parameters for the routine are the water levels and bed levels at both ends of the culvert/structure, the discharge at the previous time step and the gravitational acceleration used in the computation. You may either use the various elements directly in your formulae or define local variables as done in the example; the latter approach keeps your own code more readable and separate from the Delft3D interface. The array approach for the input items allows us to add extra input parameters to the end of the arrays without breaking compatibility with older routines.

The following module implements two formulations for culverts. The routine CULVERT implements the equations for culvert of type ‘d’ and the routine externtable (externally known at TABLE) implements a simple tabular approach to determine the discharge as function of the water level difference. The output variables of the routine are

- `discharge` If positive, the discharge from intake to outlet, and if negative, the discharge from outlet to intake. The discharge should be specified in [m³/s].
- `zpos1` If set, it specifies the height in [m] above the reference level of the culvert at the intake side. If not set, it defaults to a discharge uniform over the water depth.
- `zpos2` If set, it specifies the height in [m] above the reference level of the culvert at the outlet side. If not set, it defaults to a discharge uniform over the water depth.

```
module plugin_culvert
!
! Local constants
! Interface is in high precision
!
implicit none
integer      , parameter :: hp    = kind(1.0d0)

contains

subroutine culvert(dll_integers, max_integers, &
                   dll_reals   , max_reals   , &
                   dll_strings , max_strings , &
                   discharge, zpos1, zpos2, &
                   error_message )
!DEC$ ATTRIBUTES DLLEXPORT, ALIAS: 'CULVERT' :: CULVERT
!!--description-----
!
! Computes culvert discharge
!
!!--pseudo code and references-----
! NONE
```

¹If you can't use this compiler then make sure that the following minimum conditions are met: multi-threaded library with calling convention “C, by reference” and “string length arguments after all arguments”.

```

!!--declarations-----
implicit none
!
! Subroutine arguments: input
!
integer , intent(in) :: max_integers
integer , intent(in) :: max_reals
integer , intent(in) :: max_strings
integer , dimension(max_integers), intent(in) :: dll_integers
real(hp) , dimension(max_reals) , intent(in) :: dll_reals
character(len=256) , dimension(max_strings) , intent(in) :: dll_strings
!
! Subroutine arguments: output
!
real(hp) , intent(out) :: discharge ! discharge from 1 to 2 [m3/s]
character(len=256), intent(out) :: error_message ! not empty: echo and stop run
!
! Subroutine arguments: optional output arguments
!
real(hp) :: zpos1 ! vertical position at 1 [m]
real(hp) :: zpos2 ! vertical position at 2 [m]
!
! Local variables for input parameters
!
integer :: kfs1, kfs2
integer :: m1, m2
integer :: n1, n2, nm1, nm2
real(hp) :: ag
real(hp) :: olldis
real(hp) :: timsec
real(hp) :: zbl, zb2, zw1, zw2
character(len=256) :: filenm
character(len=256) :: runid
!
! Local variables
!
integer :: numrel1, numrel2, numrel3
real(hp) :: htcul, wtcul, poscul, calfa, cleng, cmann
real(hp), dimension(2) :: wetar1, wetar2, wetar3, closs1, closs2, closs3
!
integer :: i, iflow
real(hp) :: area
real(hp) :: cc1, cd1, cd2, cd3, coefl
real(hp) :: hin, hout, height
real(hp) :: zz1, zz2
!
!! extract array variables -----
!
if (max_integers < 8) then
    error_message = 'Insufficient integer values provided by delftflow'
    return
endif
nm1 = dll_integers( 1 ) ! nm index of point 1
m1 = dll_integers( 2 ) ! m index of point 1
n1 = dll_integers( 3 ) ! n index of point 1
kfs1 = dll_integers( 4 ) ! dry flag of point 1 (0 = dry, 1 = wet)
nm2 = dll_integers( 5 ) ! nm index of point 2
m2 = dll_integers( 6 ) ! m index of point 2
n2 = dll_integers( 7 ) ! n index of point 2
kfs2 = dll_integers( 8 ) ! dry flag of point 2 (0 = dry, 1 = wet)
!
if (max_reals < 7) then
    error_message = 'Insufficient real values provided by delftflow'
    return
endif
timsec = dll_reals( 1 ) ! current time since reference time [s]

```

```

zw1      = dll_reals( 2)      ! water level at point 1 [m]
zw2      = dll_reals( 3)      ! water level at point 2 [m]
zb1      = dll_reals( 4)      ! bed level at point 1 [m]
zb2      = dll_reals( 5)      ! bed level at point 2 [m]
olddis  = dll_reals( 6)      ! discharge at previous time step [m3/s]
ag       = dll_reals( 7)      ! gravitational acceleration [m/s2]
!
if (max_strings < 2) then
  error_message = 'Insufficient strings provided by delftflow'
  return
endif
runid   = dll_strings( 1)    ! user-specified run-identification
filenm  = dll_strings( 2)    ! user-specified file name (keyword: CulvertFile)
!
!! executable statements -----
write(*,*) 'plugin_culvert.dll : culvert : called'
!
! The output argument error_message MUST have value '' to continue the calculation.
!
error_message = ''
!
! If you want to indicate that this subroutine has encountered some invalid input or
! encountered some unexpected situation, you can set the error_message to a non-empty
! string. This error_message will then be shown in the log file of the calling program
! and the simulation will abort. This is shown by the next line, remove it to enable
! this subroutine.
!
! error_message = 'Use culvert formula ''D'' inside \DFLOW'
!
! Set some parameters and compute derivative quantities.
!
!-----
!
poscul = -4.0_hp
htcul = 1.0_hp
wtcul = 1.0_hp
numrel1 = 2
wetar1(1) = 10.0_hp
closs1(1) = 0.6_hp
wetar1(2) = 13.0_hp
closs1(2) = 0.7_hp
numrel2 = 2
wetar2(1) = 10.0_hp
closs2(1) = 0.6_hp
wetar2(2) = 13.0_hp
closs2(2) = 0.7_hp
numrel3 = 2
wetar3(1) = 10.0_hp
closs3(1) = 0.6_hp
wetar3(2) = 13.0_hp
closs3(2) = 0.7_hp
calfa = 0.03_hp
cleng = 20.0_hp
cmann = 0.015_hp
!
! compute loss coefficient:
!
area = 0.0
coefl = 0.0
if (kfs1 == 1 .or. kfs2 == 1) then
  hin = max (0.0_hp,zw1-poscul)
  hout = max (0.0_hp,zw2-poscul)
  height = 0.5 * ( hin + hout )
  height = min ( height , htcul )
  area = height * wtcul
!
```

```

! compute cd1:
!
if ( area < wetar1(1) ) cd1 = closs1(1)
if ( area >= wetar1(numrel1) ) cd1 = closs1(numrel1)
do i = 2,numrel1
  if ( area > wetar1(i-1) .and. area <= wetar1(i) ) then
    zz1 = area - wetar1(i-1)
    zz2 = wetar1(i) - wetar1(i-1)
    ccl = closs1(i) - closs1(i-1)
    cd1 = closs1(i-1) + zz1 / zz2 * ccl
  endif
enddo
!
! compute cd2:
!
if ( area <= wetar2(1) ) cd2 = closs2(1)
if ( area >= wetar2(numrel2) ) cd2 = closs2(numrel2)
do i = 2,numrel2
  if ( area > wetar2(i-1) .and. area <= wetar2(i) ) then
    zz1 = area - wetar2(i-1)
    zz2 = wetar2(i) - wetar2(i-1)
    ccl = closs2(i) - closs2(i-1)
    cd2 = closs2(i-1) + zz1 / zz2 * ccl
  endif
enddo
!
! compute cd3:
!
if ( area <= wetar3(1) ) cd3 = closs3(1)
if ( area >= wetar3(numrel3) ) cd3 = closs3(numrel3)
do i = 2,numrel3
  if ( area > wetar3(i-1) .and. area <= wetar3(i) ) then
    zz1 = area - wetar3(i-1)
    zz2 = wetar3(i) - wetar3(i-1)
    ccl = closs3(i) - closs3(i-1)
    cd3 = closs3(i-1) + zz1 / zz2 * ccl
  endif
enddo
endif
!
!-----
!
! Compute discharge
!
discharge = 0.0_hp           ! discharge (positive = from 1 to 2, negative = from 2 to 1)
if (zw1 >= zw2) then
  if (kfs1 .eq. 1) then
    call cptdis(ag      ,area      ,calfa      ,cd1      , &
                 & cd2      ,cd3      ,cleng      ,cmann      , &
                 & coefl     ,olddis    ,htcul      ,iflow      , &
                 & poscul   ,discharge ,zw1       ,zw2       , &
                 & wtcul    )
  endif
else
  !
  ! intake and outfall exchange; recompute area and energy loss
  !
  if (kfs2 .eq. 1) then
    call cptdis(ag      ,area      ,calfa      ,cd1      , &
                 & cd2      ,cd3      ,cleng      ,cmann      , &
                 & coefl     ,olddis    ,htcul      ,iflow      , &
                 & poscul   ,discharge ,zw2       ,zw1       , &
                 & wtcul    )
    discharge = -discharge
  endif
endif

```

```

!
! Optionally set vertical position of culvert
!
zpos1 = poscul
zpos2 = poscul
end subroutine culvert

subroutine cptdis(ag      ,area      ,alfa      ,cd1      , &
                  & cd2      ,cd3      ,leng      ,mann      , &
                  & coefl     ,olddis    ,htculv    ,itype      , &
                  & positc   ,rdis     ,s0in     ,s0out     , &
                  & width    )
!!--description-----
!
! Computes the discharge relation through a culverts for five flow regimes
!
!!--pseudo code and references-----
! NONE
!!--declarations-----
!
implicit none
!
!
! Local parameters
!
! minimum depth voor H* or R* (see documentation) of 0.001 m
real(hp), parameter :: hmin = 0.001
!
! Global variables
!
integer          :: itype ! flow regime (2 to 6)
real(hp), intent(in) :: ag   ! acceleration due to gravity
real(hp), intent(in) :: alfa ! energy loss correction coefficient of culvert
real(hp)           :: area  ! effective area (=htculv * width)
real(hp), intent(in) :: cd1   ! energy loss coefficients for the three types
real(hp), intent(in) :: cd2   ! energy loss coefficients for the three types
real(hp), intent(in) :: cd3   ! energy loss coefficients for the three types
real(hp)           :: coefl ! culvert loss coefficient (cd1, cd2 or cd3)
real(hp), intent(in) :: htcuv ! effective height of culvert
real(hp), intent(in) :: leng  ! length of culvert
real(hp), intent(in) :: mann  ! Manning's friction coefficient of culvert
real(hp), intent(in) :: olddis ! discharge at previous time step
real(hp), intent(in) :: positc ! vertical position of culvert
real(hp)           :: rdis   ! discharge through culvert
real(hp), intent(in) :: s0in  ! water elevation at intake
real(hp), intent(in) :: s0out ! water elevation at outfall
real(hp), intent(in) :: width  ! width of culvert
!
! Local variables:
!
integer          :: iexit
integer          :: iter
real(hp)          :: eps
real(hp)          :: hc    ! critical depth
real(hp)          :: hhh
real(hp)          :: hster ! value for wet cross section area
real(hp)          :: locdis
real(hp)          :: muster ! auxiliary value
real(hp)          :: rster
real(hp)          :: value
real(hp)          :: ztin   ! water level at intake
real(hp)          :: ztout  ! water level at outfall
!
! executable statements -----
!
```

```

eps    = 0.001
ztin  = max(0.0_hp, s0in - positc)
ztout = max(0.0_hp, s0out- positc)
!
! compute h_critical (based on averaged waterlevel):
!
locdis = olldis
do iter=1,10
    hhh    = locdis**2.0 / (ag * width**2.0)
    hc     = hhh** (1.0 / 3.0)
    hster  = min(htculv, 0.5 * ztin + 0.5 * ztout )
    hster  = max(hster, hmin)
    rster  = (hster * width ) / max(hmin, 2.0 * hster + width)
    value  = 1.0 + (2.0 * ag * leng * (mann**2.0) / rster** (4.0 / 3.0) +
    & + alfa) * (cd2**2.0)
    muster = cd2 / max(hmin, sqrt(value) )
    locdis = muster * hster * width * &
    & sqrt(2.0 * ag * max(0.0_hp, ztin - ztout))
enddo
itype = -999
if (ztin/htculv > 1.0 .and. ztout/htculv > 1.0) then
!
! type 4 (submerged flow):
!
itype = 4
rster = (htculv * width ) / max(hmin, 2.0 * htculv + 2.0 * width)
value = 1.0 + (2.0 * ag * leng * (mann**2.0) / rster** (4.0 / 3.0) + &
& alfa) * (cd2**2.0)
muster = cd2 / max(hmin, sqrt(value) )
rdis = muster * htculv * width * sqrt(2*ag*(ztin - ztout))
!coefl = muster
!area = htculv * width
elseif (ztin/htculv < 1.5 .and. ztout/htculv <= 1.0) then
if (ztout <= hc) then
!
! type 2 (supercritical flow):
!
itype = 2
do iter=1,10
    hhh    = locdis**2.0 / (ag * width**2.0)
    hc     = hhh** (1.0 / 3.0)
    hster  = min(htculv, 0.5 * hc + 0.5 * ztin )
    hster  = max(hster, hmin)
    rster  = (hster * width ) / max(hmin, 2.0 * hster + width)
    value  = 1 + (2.0 * ag * leng * (mann**2.0) / rster** (4.0 / 3.0) + &
    & alfa) * (cd1**2.0) * (hc / max(hmin, hster))**2
    muster = cd1 / max(hmin, sqrt(value) )
    rdis = muster * hc * width * sqrt(2.0 * ag * (max(hmin, ztin - hc)))
    locdis = rdis
enddo
!coefl = muster
!area = hc * width
elseif (ztout > hc) then
!
! type 3 (tranquil flow):
!
itype = 3
hster = min(htculv, 0.5 * ztin + 0.5 * ztout )
hster = max(hster, hmin)
rster = (hster * width ) / max(hmin, 2.0 * hster + width)
value = 1.0 + (2.0 * ag * leng * (mann**2.0) / rster** (4.0/3.0) + &
& alfa) * (cd1**2.0) * (ztout / max(hmin,hster))**2.0
muster = cd1 / max(hmin, sqrt(value) )
rdis = muster * ztout * width * &
& sqrt(2.0 * ag * (max(hmin, ztin - ztout)))
!coefl = muster

```

```

        !area    = ztout * width
        endif
        elseif ((ztin / htcuv >= 1.5) .and. (ztout / htcuv <= 1.0)) then
            if (ztout <= hc) then
                !
                ! type 5 (rapid flow at inlet):
                !
                itype = 5
                rdis = cd3 * htcuv * width * sqrt(2.0 * ag * ztin)
                !coefl = cd3
                !area = htcuv * width
            elseif (ztout > hc) then
                !
                ! type 6 (full flow free outlet):
                !
                itype = 6
                rster = (htcuv * width) / max(hmin, 2.0 * htcuv + 2.0 * width)
                value = 1 + (2.0 * ag * leng * (mann**2.0) / rster**(4.0 / 3.0) + &
                             alfa) * (cd2**2.0)
                muster = cd2 / max(hmin, sqrt(value))
                rdis = muster * htcuv * width * sqrt(2.0 * ag * (ztin - htcuv))
                !coefl = cd2
                !area = htcuv * width
            endif
        else
            !write (*,*) 'none of the conditions is satisfied'
            !write (*,*) 'ERROR'
        endif
    end subroutine cptdis

    subroutine externtable(dll_integers, max_integers, &
                           dll_reals   , max_reals   , &
                           dll_strings , max_strings , &
                           discharge, zpos1, zpos2, &
                           error_message )
!DEC$ ATTRIBUTES DLLEXPORT, ALIAS: 'TABLE' :: EXTERNTABLE
!!--description-----
!
! Computes discharge based on table
!
!!--pseudo code and references-----
! NONE
!!--declarations-----
implicit none
!
! Subroutine arguments: input
!
integer                               , intent(in) :: max_integers
integer                               , intent(in) :: max_reals
integer                               , intent(in) :: max_strings
integer      , dimension(max_integers), intent(in) :: dll_integers
real(hp)      , dimension(max_reals)  , intent(in) :: dll_reals
character(len=256), dimension(max_strings), intent(in) :: dll_strings
!
! Subroutine arguments: output
!
real(hp)      , intent(out) :: discharge      ! discharge from 1 to 2 [m3/s]
character(len=256), intent(out) :: error_message ! not empty: echo and stop run
!
! Subroutine arguments: optional output arguments
!
real(hp)          :: zpos1           ! vertical position at 1 [m]
real(hp)          :: zpos2           ! vertical position at 2 [m]
!
! Local variables for input parameters

```

```

!
integer          :: kfs1, kfs2
integer          :: m1, m2
integer          :: n1, n2, nm1, nm2
real(hp)         :: ag
real(hp)         :: olldis
real(hp)         :: timsec
real(hp)         :: zbl, zb2, zw1, zw2
character(len=256) :: filenm
character(len=256) :: runid
!
! Local variables
!
integer          :: istat
integer          :: n
real(hp)         :: a
real(hp)         :: dz
!
logical, save    :: first = .true.
integer, save    :: npairs
real(hp), dimension(:), allocatable, save :: dzw
real(hp), dimension(:), allocatable, save :: disch
!
!! extract array variables -----
!
if (max_integers < 8) then
    error_message = 'Insufficient integer values provided'
    return
endif
nm1    = dll_integers( 1 ) ! nm index of point 1
m1     = dll_integers( 2 ) ! m index of point 1
n1     = dll_integers( 3 ) ! n index of point 1
kfs1   = dll_integers( 4 ) ! dry flag of point 1 (0 = dry, 1 = wet)
nm2    = dll_integers( 5 ) ! nm index of point 2
m2     = dll_integers( 6 ) ! m index of point 2
n2     = dll_integers( 7 ) ! n index of point 2
kfs2   = dll_integers( 8 ) ! dry flag of point 2 (0 = dry, 1 = wet)
!
if (max_reals < 7) then
    error_message = 'Insufficient real values provided'
    return
endif
timsec  = dll_reals( 1 )      ! current time since reference time [s]
zw1     = dll_reals( 2 )      ! water level at point 1 [m]
zw2     = dll_reals( 3 )      ! water level at point 2 [m]
zbl    = dll_reals( 4 )       ! bed level at point 1 [m]
zb2    = dll_reals( 5 )       ! bed level at point 2 [m]
olldis  = dll_reals( 6 )      ! discharge at previous time step [m3/s]
ag     = dll_reals( 7 )       ! gravitational acceleration [m/s2]
!
if (max_strings < 2) then
    error_message = 'Insufficient strings provided'
    return
endif
runid   = dll_strings( 1 )    ! user-specified run-identification
filenm  = dll_strings( 2 )    ! user-specified file name
!
!! executable statements -----
!
write(*,*) 'plugin_culvert.dll : externtable : called'
! The output argument error_message MUST have value '' to continue the calculation.
!
error_message = ''
!
! Set some parameters and compute derivative quantities.
!

```

```

!-----
!
if (first) then
!
! Read parameters from filenm
!
open(717, file=filenm, status='OLD', action='READ', iostat=istat)
if (istat/=0) then
    write(error_message,*) 'Error ',istat,' while reading input file'
    return
endif
!
read(717,* , iostat=istat) npairs
if (istat/=0) then
    write(error_message,*) 'Error reading dimension of discharge table'
    return
elseif (npairs<=0) then
    write(error_message,*) 'Number of values should be at least 1'
    return
endif
!
allocate(dzw(npairs) , stat=istat)
if (istat==0) allocate(disch(npairs), stat=istat)
if (istat/=0) then
    write(error_message,*) 'Memory allocation error'
    return
endif
!
do n = 1,npairs
    read(717,* , iostat=istat) dzw(n) , disch(n)
    if (istat/=0) then
        write(error_message,*) 'Error reading value pair ',n,' of discharge table'
        return
    elseif (n==1) then
        if (dzw(n)<=0.0_hp) then
            write(error_message,*) 'First water level difference should be positive'
            return
        endif
    elseif (n>1) then
        if (dzw(n)<=dzw(n-1)) then
            write(error_message,*) 'Water level differences should be increasing'
            return
        endif
    endif
enddo
close(717)
!
first = .false.
endif
!
!-----
!
! Compute discharge (positive = from 1 to 2, negative = from 2 to 1)
!
dz = abs(zwl-zw2)
if (dz<=dzw(1)) discharge = (dz/dzw(1))*disch(1)
if (dz>dzw(npairs)) discharge = disch(npairs)
do n = 2,npairs
    if (dz>dzw(n-1) .and. dz<=dzw(n)) then
        a = (dz-dzw(n-1))/(dzw(n)-dzw(n-1))
        discharge = (1.0_hp-a)*disch(n-1) + a*disch(n)
    endif
enddo
if (zwl<zw2) discharge = -discharge
!
! Optionally set vertical position

```

```

!
!zpos1 = poscul
!zpos2 = poscul
end subroutine externtable

end module plugin_culvert

```

The last routine would require an input file containing

```

1
1.0      5.0

```

in which the first line indicates the number of lines in the table and the second and following lines indicate pivotal values for the water level difference and the associated discharge. In this particular example the discharge would vary linearly from $-5 \text{ m}^3/\text{s}$ to $5 \text{ m}^3/\text{s}$ from a water level difference of -1 to 1 m and would be constant beyond these limits. Note that the variables `zpos1` and `zpos2` have been set by the CULVERT routine but not by the TABLE routine, hence the latter discharge will be imposed uniformly over the water depth whereas the culvert discharges will be imposed as point source/sink at the specified elevation.

B.4 Space varying Coriolis coefficients

Keyword	Value	Description	Default
<code>Filcjo</code>	<code>#name.ext#</code>	Filename for spacy varying Coriolis coefficients	none

A space varying Coriolis force is useful in large area models that are not using a spherical grid (in a spherical grid formulation the Coriolis force is by definition space varying). You must specify the Coriolis force in each grid cell. The domain is $-1.4544 \cdot 10^{-4}$ to $1.4544 \cdot 10^{-4}$.

File contents A space varying Coriolis coefficient.
 File format Free formatted.
 Generated QUICKIN or manually offline.

Remarks:



- ◊ The keyword is case sensitive!
- ◊ Value *must* be enclosed between two number signs (#).
- ◊ There is no other limitation to Value.

Record description:

Record	Record description
1 to p	Coriolis coefficient for all grid points in a layer starting for n = 1 and m = 1 to Mmax, with as many values per record as allowed by the maximal record length (132) and as many continuation records (p) as necessary to specify all Mmax points.
p+1 to p · Nmax	Repeating the set of p-records for N = 2 to Nmax.

Restrictions:



- ◊ Maximum record length is 132.

- ◊ Values from the file will not be checked against their domain.
- ◊ The input items on a record must be separated by one or more blanks.

Example:

For a model area of $8 * 8$ points with its x -axis in west-east direction.

```
.1177E-3 .1177E-3 .1177E-3 .1177E-3 .1177E-3 .1177E-3 .1177E-3
.1178E-3 .1178E-3 .1178E-3 .1178E-3 .1178E-3 .1178E-3 .1178E-3
.1179E-3 .1179E-3 .1179E-3 .1179E-3 .1179E-3 .1179E-3 .1179E-3
.1180E-3 .1180E-3 .1180E-3 .1180E-3 .1180E-3 .1180E-3 .1180E-3
.1182E-3 .1182E-3 .1182E-3 .1182E-3 .1182E-3 .1182E-3 .1182E-3
.1183E-3 .1183E-3 .1183E-3 .1183E-3 .1183E-3 .1183E-3 .1183E-3
.1184E-3 .1184E-3 .1184E-3 .1184E-3 .1184E-3 .1184E-3 .1184E-3
.1185E-3 .1185E-3 .1185E-3 .1185E-3 .1185E-3 .1185E-3 .1185E-3
```

B.5 Temperature modelling

This feature concerns the elaboration of some of the available heat models that are available in Delft3D-FLOW. The extra features are:

- ◊ Direct specification of net solar radiation. This option is only available for the Ocean heat model ($ktemp=5$). An extra time series can be added to the temperature file `#name.tem#`. Used keyword: `SolRad`.
- ◊ Specification of the coefficient of free convection. This option is only available for the Ocean heat model ($ktemp=5$). Used keyword: `cFrCon`.
- ◊ Output of computed heat fluxes. For all computational points the heat fluxes (solar radiation, back radiation, evaporation, convection and the total heat flux) are written to the Map file. Used keyword: `HeaOut`.
- ◊ Output of computed evaporation rate. For all computational points the evaporation rate can be written to the Map file, as well as for all stations to the His file. The His file includes the precipitation rates; this quantity will be written to the Map file only if it varies spatially. Used keyword: `AirOut`.

B.5.1 Direct specification of net solar radiation

Keyword	Value	Description	Default
<code>SolRad</code>	<code>#Y#</code> or <code>#N#</code>	Specify net solar radiation directly in the temperature file <code>#name.tem#</code>	<code>#N#</code>

For the Ocean heat model ($ktemp=5$), the option is available to directly specify a time series of net (short wave) solar radiation in the temperature file `#name.tem#`, e.g. a time series of measured solar radiation. The specified solar radiation is then used directly in the heat model, instead of determining it from the specified cloud coverage. The cloud coverage must still be provided by the user, as it is used in other parts of the heat model, e.g. in the determination of the back radiation. For details on the specification of the input in the temperature file, see subsection A.2.18.



Remark:

- ◊ Values of character-type *must* be enclosed between two number signs (#).

B.5.2 Specification of the coefficient of free convection

Keyword	Value	Description	Default
cFrCon	1 real number	Coefficient of free convection	0.14

In the Ocean heat model ($ktemp=5$), the contribution of free convection is incorporated in the computation of the heat fluxes. The coefficient for free convection has a default value of 0.14 [-]. The contribution of free convection can be switched off by giving the coefficient a value 0.0:

```
cFrCon= 0.0
```

For more details on the contribution of free convection and its conceptual description, see Sections [9.8.6](#) and [9.8.7](#).

B.5.3 Output of computed heat fluxes

Keyword	Value	Description	Default
HeaOut	#Y# or #N#	Write the computed heat fluxes to the Map file for all computational points	#N#

Using keyword HeaOut the computed heat fluxes are written to the Map file. Which heat fluxes are written to the Map file, depends on the heat model that is used.

For heat models 1, 2, 4 and 5:

- ◊ Evaporation heat flux
- ◊ Heat flux of forced convection
- ◊ Nett back radiation
- ◊ Nett solar radiation
- ◊ Total nett heat flux
- ◊ Computed minus derived heat flux (when relevant, see [section B.6](#))

Additionaly for heat model 5 - Ocean model (if free convection is included):

- ◊ Free convection of sensible heat
- ◊ Free convection of latent heat

For heat model 3 - Excess temperature model:

- ◊ Exchange coefficient in Excess temperature model
- ◊ Total nett heat flux

Remarks:

- ◊ Values of character-type *must* be enclosed between two number signs (#).
- ◊ The heat fluxes are not (yet) written to the History file. Therefore one can not yet inspect time series of heat fluxes at monitoring stations.
- ◊ More details on the incorporation of free convection can be found in Sections [9.8.6](#) and [9.8.7](#). Free convection is only included in the Ocean heat model ($ktemp=5$).

B.6 Evaporation and precipitation

In certain simulations, e.g. in tropical areas, the evaporation and precipitation can not be neglected. Upon using the `Fileva` keyword, you can specify the time-series for evaporation and precipitation (and its temperature) in a separate ASCII file. The interpolation method (*Block* or *Linear*) can be set via the keyword `Evaint`.

When `Fileva` **IS NOT** specified, the evaporation will be calculated internally when needed for the heat balance (heat model 1, 2, 4 or 5, see [subsection 9.8.8](#)). In the heat balance, the calculated evaporation heat flux **is** used. In the mass balance, the calculated evaporation mass flux **is not** used, by default. When adding `Maseva = #Y#` to the MDF-file, the mass flux of the calculated evaporation *will* be used in the mass balance.

When `Fileva` **IS** specified with the value -999.0 for the evaporation in the complete input file, the evaporation will be calculated internally when needed for the heat balance (heat model 1, 2, 4 or 5, see [subsection 9.8.8](#)). In the heat balance, the calculated evaporation heat flux and the specified precipitative heat flux **are** used. In the mass balance, the calculated evaporation mass flux and the specified precipitative mass flux **are** used, by default. When adding `Maseva = #N#` to the MDF-file, the mass flux of the calculated evaporation and the specified precipitation *will not* be used in the mass balance.

When `Fileva` **IS** specified with realistic values for the evaporation in the complete input file, the evaporation will be calculated internally when needed for the heat balance (heat model 1, 2, 4 or 5, see [subsection 9.8.8](#)). In the heat balance, the *calculated* evaporation heat flux and the specified precipitative heat flux **are** used, by default. In the mass balance, the *specified* evaporation mass flux and the specified precipitative mass flux **are** used, by default. When adding `Maseva = #N#` to the MDF-file, the mass flux of the specified precipitation and the calculated evaporation *will not* be used in the mass balance. When adding `QEvp = #derived#` to the MDF-file, the heat flux of the specified precipitation and the *specified* evaporation *will* be used in the heat balance.



Remark:

- ◊ By using the keyword `AirOut`, the evaporation rate for all computational points can be written to the Map file, as well as for all stations to the His file. The His file includes the precipitation rates; this quantity will be written to the Map file only if it varies spatially.

Keyword	Value	Description	Default
<code>Fileva</code>	<code>#name.eva#</code>	Filename	none
<code>Evaint</code>	<code>#Y#</code> or <code>#N#</code>	Linear (Y) or block (N) interpolation	<code>#Y#</code>
<code>Maseva</code>	<code>#Y#</code> or <code>#N#</code>	Add evaporation and precipitation to the continuity equation	<code>#Y#</code> (with <code>Fileva</code>) <code>#N#</code> (without <code>Fileva</code>)
<code>QEvp</code>	<code># derived #</code> or <code># computed #</code>	Evaporation heat flux: <ul style="list-style-type: none"> ◊ is derived from the prescribed evaporation mass flux (<i>derived</i>) ◊ is calculated internally, independent of the prescribed mass flux (<i>computed</i>) 	<code># computed #</code>

File contents	Time dependent data for evaporation and precipitation.
File format	Free formatted.
Filename	<name.eva>
Generated	Manually offline.

Remarks:

- ◊ Evaporation and precipitation can not be combined with the Excess temperature model (model 3 in [subsection 9.8.8](#)).
- ◊ When using an evaporation value of -999.0 in the complete input file, Delft3D-FLOW will calculate internally both the evaporation mass flux and heat flux.
- ◊ When adding `HeaOut = #Y#` to the MDF-file, additional output will be written to the map-file, including the mismatch between the calculated and the derived evaporation heat flux (when relevant), see [subsection B.5.3](#).
- ◊ The keyword is case sensitive!
- ◊ Non-numerical values *must* be enclosed between two number signs (#).

**Record description:**

Record	Record description
each record	Time in [minutes] after 00:00 on the Reference Date (1 real), Precipitation in [mm/hour] (1 real), Evaporation in [mm/hour] (1 real), Rain temperature in [°C] (1 real).

Restrictions:

- ◊ Time must be integer multiples of the integration time step.
- ◊ The value of the parameters is not checked against its domain.
- ◊ Input items in a record must be separated by one or more blanks.

**Example:**

```

0.      1.      1.      10.
10.     1.      1.      10.
20.     2.      1.      11.
30.     2.      10.     12.
150.    1.      15.     15.
600.    1.      23.     21.

```

B.7 Space varying wind and pressure

In many cases the space varying wind data is provided by a meteorological station. This data is often defined on a different grid than the computational grid used in Delft3D-FLOW. Translating these files into files defined on the (curvilinear) grid of the computational engine is often a lengthy process and can result in huge files. This feature facilitates the reading of the meteorological data on its own grid and interpolates the data internally to the grid of Delft3D-FLOW. Delft3D-FLOW can handle three types of meteorological input:

- 1 Time- and space-varying wind on an equistant grid
- 2 Time- and space-varying wind on a curvilinear grid
- 3 Time- and space-varying wind on a Spiderweb grid

The descriptions in this section are conform the current file format for meteorological in-

put. A large update of the meteo input file formats was introduced in Delft3D-FLOW version 3.55.05.823. Since then, several smaller modifications have been made. Details on these modifications in the file formats can be found in the sections below describing the different meteo input types.

Convert file format

To convert a meteo file from before Delft3D-FLOW version 3.55.05.823 (not keyword based) into the new format, a script is available, written in Matlab, named <meteo_old2new.m>. The Delft3D distribution contains this script. Instructions for conversion of files with versions later than Delft3D-FLOW version 3.55.05.823 to newer versions can be found in the sections below.

B.7.1 Space varying wind and pressure on an equidistant grid

Keywords	Value	Description	Default
Filwp	#name.amp#	file for atmospheric pressure	none
Filwu	#name.amu#	file for wind speed in East-direction	none
Filwv	#name.amv#	file for wind speed in North-direction	none

Additional keywords for relative air humidity, air temperature and air cloudiness are also (optionally) available:

Keywords	Value	Description	Default
Filwr	#name.amr#	file for relative air humidity	none
Filwt	#name.amt#	file for air temperature	none
Filwc	#name.amc#	file for air cloudiness	none

File contents Time-series of a space varying wind and atmospheric pressure defined on an equidistant (Cartesian or spherical) grid.

File format Free formatted or unformatted, keyword based.
Generated Some offline program.



Remark:

- ◊ The keywords are case insensitive.

Header description for the wind velocity files:

Keywords	Value	Description
FileVersion	1.03	version of file format
Filetype	meteo_on_equidistant_grid	meteo input on equidistant grid
NODATA_value	free	value used for input that is to be neglected

Keywords	Value	Description
n_cols	<i>free</i>	number of columns used for wind datafield
n_rows	<i>free</i>	number of rows used for wind datafield
grid_unit	<i>m or degree</i>	unit of distances on the grid in both <i>x</i> - and <i>y</i> -direction
x_llcorner	<i>free</i>	<i>x</i> -coordinate of lower left corner of lower left grid cell (in units specified in grid_unit)
y_llcorner	<i>free</i>	<i>y</i> -coordinate of lower left corner of lower left grid cell (in units specified in grid_unit)
x_llcenter	<i>free</i>	<i>x</i> -coordinate of centre of lower left grid cell (in units specified in grid_unit)
y_llcenter	<i>free</i>	<i>y</i> -coordinate of centre of lower left grid cell (in units specified in grid_unit)
dx	<i>free</i>	gridsize in <i>x</i> -direction in units specified in grid_unit
dy	<i>free</i>	gridsize in <i>y</i> -direction in units specified in grid_unit
n_quantity	1	number of quantities specified in the file
quantity1	<i>x_wind or y_wind</i>	the velocity component given in unit unit1
unit1	<i>m s-1</i>	unit of quantity1: metre/second

The user must specify the location of the equidistant grid on which the meteorological data is specified. If one has the location of the lower left corner of the lower left grid cell, one can specify the starting point of the grid using keywords x_llcorner and y_llcorner. If one has the location of the cell centre of the lower left grid cell, one should use the keywords x_llcenter and y_llcenter. Using the first option, the first data value is placed at $(x_{llcorner} + \frac{1}{2}dx, y_{llcorner} + \frac{1}{2}dy)$, which is the cell centre of cell (1,1). Using the latter option, the first data value is placed at (x_llcenter, y_llcenter), which is again the cell centre of cell (1,1), i.e. the data values are always placed at the cell centres of the meteorological grid. Note that the lower left grid cell is defined to be the grid cell with index (1,1). When using the option of meteorological data on a separate curvilinear grid, the origin and orientation of the data set can be chosen freely with respect to the grid on which it is specified, see subsection B.7.2 for details.

Time definition and data block description for the wind velocity files

Keywords	Value	Description
Time	<i>fixed format described below</i>	time definition string

The time definition string has a fixed format, used to completely determine the time at which a dataset is valid. The time definition string has the following format:

TIME *minutes/hours since YYYY-MM-DD HH:MM:SS TIME ZONE*, e.g.

360 minutes since 2008-07-28 10:55:00 +01:00

The format of the string is completely fixed. No extra spaces or tabs can be added between the different parts of the definition. The time definition is followed by the datablock of input values corresponding to the specified time. The data block contains values for the wind velocity in *x*- or *y*-direction for *n_cols* by *n_rows* points, starting at the top left point. The time definition and the data block are repeated for each time instance of the time-series.

The atmospheric pressure file

The header for the atmospheric pressure is similar to that of the wind velocity files, except for the following differences.

Keywords	Value	Description
quantity1	air_pressure	air pressure
unit1	Pa or mbar	unit of quantity1: Pascal or millibar

The specification of the time definition and the data block is fully conform the wind velocity files.

File version and conversion

The current description holds for FileVersion 1.03. The table below shows the latest modifications in the file format (and version number).

FileVersion	Modifications
1.03	Use of keyword <code>Value_pos</code> to indicate the position of the lower left corner of the grid replaced by use of the combination of keywords: <code>x_llcorner</code> and <code>y_llcorner</code> or <code>x_llcenter</code> and <code>y_llcenter</code>
1.02	No changes for this meteo input type, but for the meteo type <code>meteo_on_spiderweb_grid</code>
1.01	Changed keyword <code>MeteoType</code> to <code>FileType</code> Changed fixed value of input type (Keyword <code>Filetype</code>) from <code>ArclInfo</code> to <code>meteo_on_equidistant_grid</code>

Restrictions:

- ◊ The contents of the file will not be checked on its domain.
- ◊ Keywords are followed by an equal sign '=' and the value of the keyword.
- ◊ When a keyword has value *free*, the value of this keyword is free to choose by the user.
 When only one value is given for a keyword, this keyword has a fixed value and when 2 or more options are shown, the user can choose between those values.
- ◊ Times must be specified exactly according to the time definition. See the examples shown in this section.
- ◊ The atmospheric pressure file must use the same grid definition and time frame as the files for the wind velocity components.
- ◊ The unit of the meteo grid must be the same as the computational grid, i.e. both with `grid_unit = [m]` or both with `grid_unit = [degree]`.
- ◊ Input items in a data block are separated by one or more blanks.
- ◊ The wind components are specified at the cell centres (water level points) of the numerical grid.
- ◊ The wind components are specified in the west-east (`x_wind`) and south-north directions (`y_wind`).

**Remarks:**

- ◊ The time definition in the meteo files contains the number of minutes or hours since a reference date and time in a certain time zone. The reference time and time zone may differ from those of the simulation. During a simulation the computational engine will search in the meteo file for the current simulation time and interpolate between neighbouring times if necessary. Possible differences in time zone will be accounted for by shifting the meteo input data. The reference times within the time definition string may vary in a meteo file, i.e. it is possible to attach new input with a different reference time, behind the last data block. Consecutive times must always be increasing in the input file.
- ◊ Comments can be added after pound signs (#). These are not read.

**Example of a file containing wind in x-direction (west-east)**

The data blocks in this example are the result of the following FORTRAN statements:

```
do j = nrows,1,-1
  write(out,*) (xwind(i,j),i=1,ncols)
enddo
```

The *x*-wind velocity file for a 3 (`n_cols`) by 4 (`n_rows`) grid has the following layout:

```

FileVersion      = 1.03
filetype        = meteo_on_equidistant_grid
NODATA_value   = -999.000
n_cols          = 3
n_rows          = 4
grid_unit       = degree
x_llcenter     = -12.000
y_llcenter     = 48.000
dx              = 0.12500
dy              = 0.083333333
n_quantity      = 1
quantity1       = x_wind
unit1           = m s-1
TIME =         0.0 hours since 2008-01-15 04:35:00 +00:00
2   3.0  3.6
3   4.5  2
2.2 1   2.3
1.2 0.7 -0.4
TIME =         6.0 hours since 2008-01-15 04:35:00 +00:00
-1.1 -2.3 -3.6
-3.2  0.8  1.1
2.2 -1   -1.6
1.2 -0.7 -0.4

```

This results in an *x*-component of wind velocity given - in [m/s] - on a spherical, 3 by 4, equidistant grid, with grid sizes given by *dx* and *dy* (in degrees) and where the centre point of the lower left cell of the grid lies in (longitude, latitude) (-12.0, 48.0) on the globe. Data is given at two times: 0 and 6 hours since January 15th, 2008, 4:35 AM, in UTC+0.



Remarks:

- ◊ The wind file specified in the Data Group *Physical parameters - Wind* is overruled when the keywords for wind on an equidistant grid (see [subsection B.7.1](#)) are specified in the Data Group *Additional parameters*.
- ◊ The layout of the data blocks is from north to south (whereas most of the other files in Delft3D-FLOW, such as the curvilinear grid file, are ordered from south to north).
- ◊ On open boundaries, an input signal is prescribed that is consistent with some average pressure. Usually the signal corresponds to Mean Sea Level. One actually wants to prescribe an input signal corresponding to the local pressure prescribed by the space varying meteo input. To this end, it is possible to specify an average pressure - which should correspond to your input signal on the open boundaries - which is then used to determine local pressure gradients that need to be applied along the open boundaries to obtain an input signal that is consistent with the local atmospheric pressure. This functionality used to be specified in the wind file but it should now be specified in the Master Definition File in the Data Group *Additional parameters*, using *PavBnd: Average Pressure on Boundaries*. Using this keyword one can specify an average pressure that is used on all open boundaries, independent of the type of wind input. The pressure must be specified in N/m². An example:

PavBnd= 101300.0

B.7.2 Space varying wind and pressure on a separate curvilinear grid

Keywords	Value	Description	Default
Fwndgp	#name.amp#	file for atmospheric pressure	none
Fwndgu	#name.amu#	file for wind speed in East-direction	none
Fwndgv	#name.amv#	file for wind speed in North-direction	none

Optionally additional keywords for relative air humidity, air temperature and air cloudiness:

Keywords	Value	Description	Default
Fwndgr	#name.amr#	file for relative air humidity	none
Fndgt	#name.amt#	file for air temperature	none
Fndgc	#name.amc#	file for air cloudiness	none

File contents Time-series of a space varying wind and atmospheric pressure defined on a curvilinear (Cartesian or spherical) grid.
 File format Free formatted or unformatted, keyword based.
 Generated Some offline program.

Remark:

- ◊ The keywords are case insensitive.



Header description for the wind velocity files:

Keywords	Value	Description
FileVersion	1.03	version of file format
Filetype	meteo_on_curvilinear_grid	meteo input on curvilinear grid
NODATA_value	free	value used for input that is to be neglected
grid_file	free.grd	name of the curvilinear grid file on which the data is specified
first_data_value	grid_llcorner or grid_ulcorner or grid_lrcorner or grid_urcorner	see example below
data_row	grid_row or grid_column	see example below
n_quantity	1	number of quantities specified in the file
quantity1	x_wind or	the velocity component given in

Keywords	Value	Description
	y_wind	unit unit1
unit1	m s-1	unit of quantity1: metres/second

Time definition and data block description for the wind velocity files

For a description of the time definition and data block see [subsection B.7.1](#).

The atmospheric pressure file

For a description of the atmospheric file see [subsection B.7.1](#).

File version and conversion

The current description holds for `FileVersion` 1.03. The table below shows the latest modifications in the file format (and version number).

FileVersion	Modifications
1.03	<p>Fixed bug in interpolation of data from meteo grid to computational grid: Conversion script mirrored data set erroneously. This was treated correctly by meteo module. Fixed both the conversion script and the meteo module together: Required modification in meteo input file:</p> <p>Change <code>first_data_value = grid_llcorner</code> into <code>grid_ulcorner</code> or vice versa</p> <p>or</p> <p>Change <code>first_data_value = grid_lrcorner</code> into <code>grid_urcorner</code> or vice versa</p>
1.02	No changes for this meteo input type, but for the meteo type <code>meteo_on_spiderweb_grid</code>
1.01	<p>Changed keyword <code>MeteoType</code> to <code>FileType</code></p> <p>Changed keyword <code>Curvi_grid_file</code> to <code>Grid_file</code></p> <p>Changed fixed value of input type (<code>Keyword filetype</code>) from <code>Curvi</code> to <code>meteo_on_curvilinear_grid</code></p>



Restrictions:

- ◊ The restrictions for space varying wind and pressure on a separate curvilinear grid are the same as for space varying wind and pressure on an equidistant grid, described in [subsection B.7.1](#). A difference is that the data values on the curvilinear grid are not specified in the cell centres, but in the grid points (cell corners).
- ◊ The unit of the meteo grid must be the same as the computational grid, i.e. both with `grid_unit = [m]` or both with `grid_unit = [degree]`.



Remark:

- ◊ The remarks for space varying wind and pressure on a separate curvilinear grid are the same as for space varying wind and pressure on an equidistant grid, described in

subsection B.7.1.

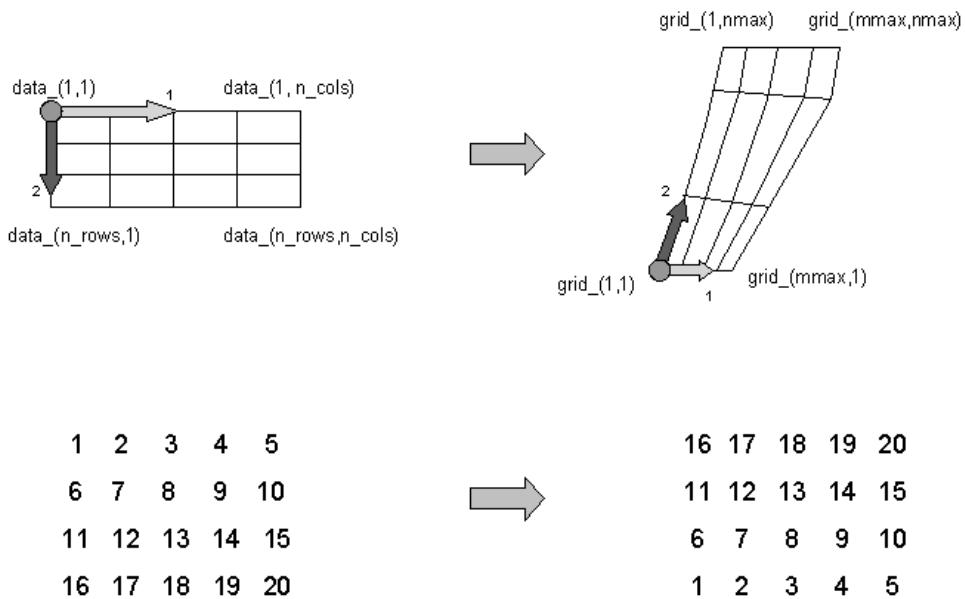


Figure B.8: Illustration of the data to grid conversion for meteo input on a separate curvilinear grid

Example:

A file for input of x -velocity (in west-east direction) on a 4 (n_rows) by 5 (n_cols) curvilinear grid, where the meteorological data is mirrored vertically with respect to the grid, has the following layout:

```

FileVersion      = 1.03
filetype        = meteo_on_curvilinear_grid
NODATA_value   = -999.000
grid_file       = curviwind.grd
first_data_value = grid_llcorner
data_row        = grid_row
n_quantity     = 1
quantity1      = x_wind
unit1          = m s-1
TIME = 0.0 minutes since 1993-06-28 14:50:00 -02:00
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
TIME = 600.0 minutes since 1993-06-28 14:50:00 -02:00
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20

```

This results in an x -component of velocity given - in [m/s] - on the curvilinear grid specified in file <curviwind.grd>. The data set will be mirrored such that the first value of the data (upper left corner, in the example '1') corresponds to the lower left corner of the grid (point (1,1)) and a row of data corresponds to a row on the grid, see [Figure B.8](#). Data is given at two times: 0 and 600 minutes since June 28th, 1993, 14:50 PM, in UTC-2.

Remark:



- ◊ The wind file specified in the Data Group *Physical parameters - Wind* is overruled when the keywords for wind on a separate curvilinear grid (see [subsection B.7.2](#)) are specified in the Data Group *Additional parameters*.

B.7.3 Space varying wind and pressure on a Spiderweb grid

Keywords	Value	Description	Default
Filweb	#name.web#	file containing wind speed, wind direction and atmospheric pressure drop	none

Cyclone winds are governed by a circular motion, combined with a cyclone track. This type of wind is generally very difficult to implement on a curvilinear grid. This feature facilitates the reading of the so-called Spiderweb files and interpolates the wind and pressure data internally to the computational grid. A special feature of the space varying wind and pressure on the Spiderweb grid is that it can be combined with one of the other meteorological input options described in this manual, i.e. to either uniform wind and pressure, or to one of the space varying wind and pressure options, see [subsection A.2.8](#).

File contents	Time-series of a space varying wind and atmospheric pressure defined on a Spiderweb grid. This grid may be specified in Cartesian or spherical coordinates.
File format	Free formatted or unformatted, keyword based.
Generated	Some offline program.



Remarks:

- ◊ The keywords are case insensitive.
- ◊ Space varying wind and pressure on a Spiderweb grid is added to other wind input and the wind fields are interpolated and combined in and around the cyclone.

Header description of the Spiderweb wind and pressure file:

Keywords	Value	Description
FileVersion	1.03	version of file format
Filetype	meteo_on_spiderweb_grid	meteo input on Spiderweb grid
NODATA_value	free	value used for input that is to be neglected
n_cols	free	number of gridpoints in angular direction
n_rows	free	number of gridpoints in radial direction
grid_unit	m or degree	unit of the Spiderweb grid
spw_radius	free	radius of the spiderweb given in units given by spw_rad_unit

Keywords	Value	Description
spw_rad_unit	m	unit of the Spiderweb radius
spw_merge_frac	[0.0,1.0]	fraction of the Spiderweb radius where merging starts of the background wind with the Spiderweb wind. Default is 0.5
air_pressure_reference	air_pressure_default_from_computational_engine <i>or free</i>	Both keyword and value are too long to fit on one line. Reference value related to p_drop is the default air pressure of the computational engine <i>or</i> the value specified. If missing, p_drop is extracted from the actual atmospheric pressure.
n_quantity	3	number of quantities specified in the file
quantity1	wind_speed	wind speed given in unit unit1
quantity2	wind_from_direction	direction where the wind is coming from given in unit unit2
quantity3	p_drop	drop in atmospheric pressure given in unit unit3
unit1	m s-1	unit of quantity1: metres/second
unit2	degree	unit of quantity2: degrees
unit3	Pa or mbar	unit of quantity3: Pascal or millibar

Time definition and data block description

For a description of the time definition see [subsection B.7.1](#).

Cyclone track information:

For each time in the time series of space varying wind and pressure on a Spiderweb grid, the position of the cyclone eye (and thus also the spiderweb grid) must be given, as well as the drop of atmospheric pressure in the cyclone eye.

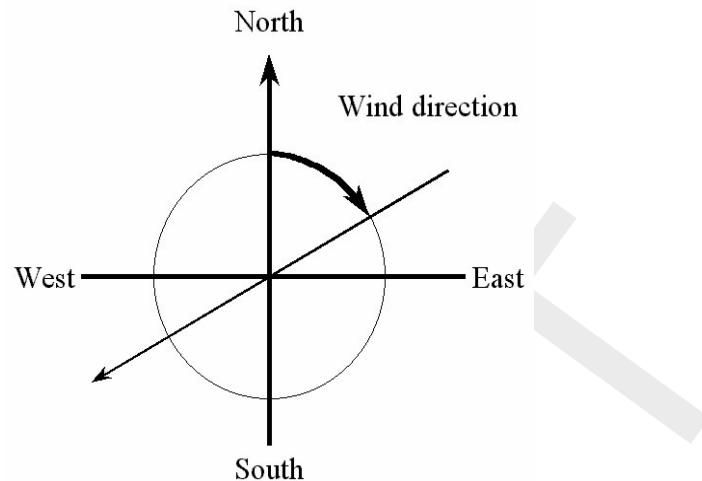


Figure B.9: Wind definition according to Nautical convention

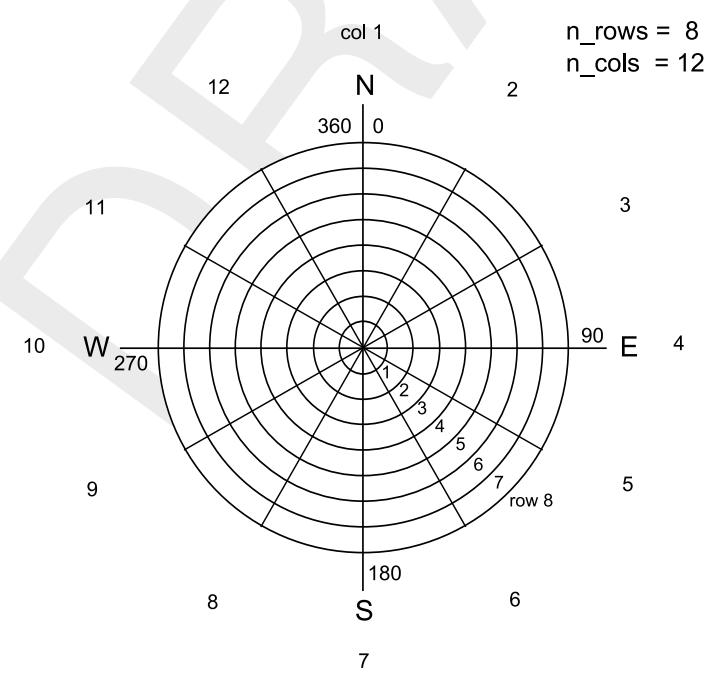


Figure B.10: Spiderweb grid definition

File version and conversion

The current description holds for FileVersion 1.03. The table below shows the latest modifications in the file format (and version number).

FileVersion	Modifications
1.03	No changes for this meteo input type
1.02	Changed the use of keyword <code>n_rows</code> and <code>n_cols</code> . The radius of the cyclone is divided in <code>n_rows</code> rings of width: spw_radius/n_rows [m] and the circle is divided in <code>n_cols</code> parts of $2\pi/n_cols$ [rad].
1.01	Changed keyword <code>MeteoType</code> to <code>FileType</code> Changed fixed value of input type (Keyword <code>Filetype</code>) from <i>Spiderweb</i> to <i>meteo_on_spiderweb_grid</i>

Restriction:

- ◊ The restrictions for space varying wind and pressure on a Spiderweb grid are the same as for space varying wind and pressure on an equidistant grid, described in [subsection B.7.1](#).



Remarks:

- ◊ The remarks for space varying wind and pressure on a separate curvilinear grid are the same as for space varying wind and pressure on an equidistant grid, described in [subsection B.7.1](#).
- ◊ The Spiderweb grid is circular and the definitions of the number of rows `n_rows` and the number of columns `n_cols` is therefore different than for the other meteo input formats. For the Spiderweb grid, the number of rows determines the grid size in radial direction. The number of columns defines the grid size in angular direction. See [Figure B.10](#).
- ◊ The wind is specified according to the nautical convention, i.e. wind from the true North has direction zero and the wind turns clockwise with an increasing angle. See [Figure B.9](#).



Example:

A file for input of space varying wind and pressure on a 5x3 Spiderweb grid, has the following layout:

```

FileVersion          = 1.03
filetype            = meteo_on_spiderweb_grid
NODATA_value       = -999.000
n_cols              = 3
n_rows              = 5
grid_unit           = degree
spw_radius          = 600000.0
spw_rad_unit        = m
air_pressure_reference = air_pressure_default_from_computational_engine
n_quantity          = 3
quantity1            = wind_speed
quantity2            = wind_from_direction
quantity3            = p_drop
unit1                = m s-1
unit2                = degree
unit3                = Pa

```

```

TIME      = 0.0 hours since 1997-07-14 03:00:00 -06:00
x_spw_eye = 115.1
y_spw_eye = 18.9
p_drop_spw_eye = 5300.0
1.38999   1.38261   1.38315
1.28251   1.34931   1.22571
1.27215   1.31214   1.32451
1.38999   1.86592   2.87732
1.43899   1.24912   2.21519
60.0000   180.0000  270.0000
28.7500   20.0000   31.2500
42.5000   53.7500   65.0000
49.3400   60.2400   81.5200
51.4100   62.0000   43.1200
5301.280  5294.490  5156.240
5043.460  5112.040  5264.020
5140.020  5202.520  5411.210
5294.730  5285.760  5235.250
5242.530  5156.190  5124.240
TIME      = 1.0 hours since 1997-07-14 03:00:00 -06:00
x_spw_eye = 114.8
y_spw_eye = 18.8
p_drop_spw_eye = 5250.0
1.35763   1.35763   1.35763
1.35763   1.87273   2.24784
1.92214   2.47836   2.17266
1.87662   2.72116   2.82375
1.26585   2.24146   2.38722
159.0000  346.5200  290.6400
342.3200  282.1400  20.2400
10.7500   25.5300   36.4500
61.8400   81.6200   45.5100
49.5250   56.7500   75.1300
5314.520  5104.490  5287.240
5124.240  5285.760  5252.420
5152.460  5247.040  5222.020
5242.020  5223.520  5475.210
5244.270  5211.210  4998.110

```

This results in the following set of meteo data. Velocities given in [m/s] and pressure drops in [Pa] on a Spiderweb grid which is given in spherical coordinates (grid_unit = degree). The cyclone and spiderweb grid have a radius of 600 km. The grid is 5x3, which means the radius is divided in five parts of 120 km and the 360 degrees are divided in 3 parts of 120 degrees each. Wind speeds, wind directions and pressure drops are given at two times: 0 and 1.0 hour since July 14th, 1997, 03:00 AM, in UTC-6. Between these two times the cyclone eye moves from (longitude, latitude) (115.1, 18.9) to (114.8, 18.8) on the globe and the pressure drop in the cyclone eye decreases from 5300.0 [Pa] to 5250.0 [Pa].

B.8 Horizontal large eddy simulation

HLES is supported by the FLOW-GUI, see Data Group *Physical parameters - Viscosity*.

Keywords	Value and unit	Description	Default	Range
Htur2d	#YES#	Enable the use of HLES	#N#	
Htural	1.666666 [-]	Slope in log-log spectrum	1.666666	[1,3]
Hturnd	2 [-]	Dimensional number	2	2 or 3
Hurst	0.7 [-]	Prandtl-Schmidt number	0.7	[0.5,1]
Hturlp	0.3333333 [-]	Spatial low-pass filter coefficient	0.3333333	[0.2,1]
Hturrtau	1.0 [min]	Relaxation time	-1.0	> 0.0 (or -1.0)
Hturdm	0.0 [m ² /s]	Molecular diffusivity	0.0	= 0.0
Hturel	#YES#	Switch to add Elder's term	#YES#	

In addition the *Slip condition* for **Wall roughness** can be changed to “no”.

Remark:



- ◊ The possible values for the *Slip condition* are:
 - free slip condition at the side walls (*Irov* = 0),
 - partial slip with user-defined wall roughness length (*Irov* = 1),
 - no-slip condition at the side walls (*Irov* = 2).
- ◊ The choice of the slip condition depends on the grid size versus the boundary layer thickness. It is of no use to apply a partial or no-slip condition when the grid size cannot resolve the boundary layer.
- ◊ No-slip and partial slip conditions along smoothly curved boundaries can support the separation of the boundary layer and the wall, whereas a free slip condition cannot.
- ◊ Elder's term is only applicable for 2DH models.
- ◊ The relaxation time τ should be assigned a positive value in agreement with the time scale to be used in Eq. B.7. If τ is set to -1.0 then the $\bar{\psi}_{n+1}^t$ term in Eq. B.6 is ignored such that in Eq. B.5 the gradients of the full instantaneous velocities are used instead of those of the high-pass filtered velocities. This means that if τ is set to -1.0 also a (near-)steady velocity gradient will contribute to the sub-grid eddy viscosity. For practical applications a positive value of τ should be used; its value may be determined as 2 to 6 times the (largest) time scale of the eddies, however, one should take care not to filter out the higher harmonics (e.g. M4 or M6) of the flow forcing. It is advised to verify model sensitivity for the chosen value of τ .

Htural, Hturnd, Hurst, Hturlp, Hturrtau and Hturdm correspond with α , n_D , σ_T , f_{lp} , τ , and D_{mol} . Hturel corresponds with Elder: a switch to enable the addition of the Elder's term to the depth-averaged parabolic diffusivity profile to take into account the effect of a 3D velocity profile (so-called Taylor-shear dispersion) in a depth-averaged model.

Detailed description:

Below we summarise the essential aspects of the sub-grid scale model for Horizontal Large Eddy Simulation (HLES) of shallow water flows subjected to bed friction. The result of this HLES model is a formulation of the horizontal component of the sub-grid eddy viscosity and the sub-grid eddy diffusivity.

The model is based on theoretical considerations presented by [Uittenbogaard \(1998\)](#); [Uittenbogaard and Van Vossen \(2003\)](#) and summarised by [Van Vossen \(2000\)](#). This model for the sub-grid eddy viscosity ν_{SGS} includes the damping of sub-grid eddies by bed friction reads:

$$\nu_{SGS} = \frac{1}{k_s^2} \left(\sqrt{(\gamma \sigma_T S^*)^2 + B^2} - B \right), \quad (\text{B.2})$$

with:

$$B = \frac{3g |\vec{U}|}{4HC^2}, \quad (\text{B.3})$$

where C is the Chézy coefficient and H is the water depth.

Correspondingly, the sub-grid eddy diffusivity D_{SGS} for horizontal mixing of heat, salt, mud and dissolved constituents reads:

$$D_{SGS} = \frac{\nu_{SGS}}{\sigma_T}. \quad (\text{B.4})$$

The definition of all symbols in Eqs. [B.2](#) to [B.4](#) are presented hereafter.

In Eq. [B.2](#) the sum of strain rates squared (i.e. double contraction of horizontal component of the strain rate tensor) reads:

$$(S^*)^2 = 2 \left(\frac{\partial u^*}{\partial x} \right)^2 + 2 \left(\frac{\partial v^*}{\partial y} \right)^2 + \left(\frac{\partial u^*}{\partial y} \right)^2 + \left(\frac{\partial v^*}{\partial x} \right)^2 + 2 \frac{\partial u^*}{\partial y} \frac{\partial v^*}{\partial x}. \quad (\text{B.5})$$

In Eq. [B.5](#), the superscript $(\dots)^*$ denotes fluctuating flow variables and these are defined through the following recursive high-pass filter operator ([Bendat and Piersol, 1971](#), p. 297–298):

$$\psi^* = \psi_{n+1} - \bar{\psi}_{n+1}^t, \quad (\text{B.6})$$

with:

$$\bar{\psi}_{n+1}^t = (1 - a) \psi_{n+1} + a \bar{\psi}_n^t, \quad \bar{\psi}_0^t = 0 \text{ and } a = \exp(-\Delta t/\tau), \quad (\text{B.7})$$

with Δt is the time step, n is the time step number, $(n+1)$ denotes the present time step and τ is the relaxation time. In Eq. [B.6](#), ψ represents any of the velocity components (u, v).

If the relaxation time τ is specified as -1.0 then $\bar{\psi}_{n+1}^t$ is set to zero which means that, effectively, the high-pass filtering operation is thereby cancelled. A positive value τ is strongly recommended; see also remarks above.

When you apply the HLES-model, the following considerations are of importance:

- ◊ According to the theory just the spatially and temporarily varying fluctuations should enter Eq. B.2.
- ◊ The application of a high-pass filter just for temporal fluctuations and not (also) in space is a pragmatic choice. Spatial filtering on non-equidistant curvilinear grids with thin dams or structures is computationally tedious whereas temporal variations on sufficiently short time scales are due to turbulent eddies only and hence just temporal filtering is the optimal choice.
- ◊ Not only the total velocity vector (u, v) is on output but also the high-pass filtered horizontal velocity components $(\bar{u}_n^t, \bar{v}_n^t)$ are internally stored as well as written to standard output thereby enabling plotting or animating the spatial vector field (u_n^*, v_n^*) that visually presents vortex structures superimposed on the (mean) current.
- ◊ For steady flows $(u, v)^* = 0$ holds so that $S^* = 0$ and consequently the sub-grid scale (SGS) eddy viscosity $\nu_{SGS} = 0$.
- ◊ For $\nu_{SGS} = 0$ the simulated (mean) flow subjected to some horizontal shearing can be become unstable, in agreement with the Orr-Sommerfeld hydrodynamic stability equation, provided (physical and numerical) internal friction and bed friction are sufficiently small. The SGS model's property $\nu_{SGS} = 0$ for time independent flow is similar to the essential property of the well-known dynamic SGS model that applies spatial filtering on equidistant Cartesian grids, see [Germano et al. \(1991\)](#).
- ◊ When the unstable flow creates temporal velocity variations then the SGS eddy viscosity Eq. B.2 develops.
- ◊ Towards drying conditions, the local water depth H decreases to nearly zero which then reduces ν_{SGS} to zero, see Eq. B.2.
- ◊ At drying, the time filtered velocity components $(\bar{\psi}_{n+1}^t$ in Eq. B.6) are set to zero.
- ◊ After flooding, the water depth H increases from nearly zero and then ν_{SGS} increases gradually for increasing water depth but constant high-pass filtered strain rates, see Eq. B.2.
- ◊ In principle, turbulence fluctuations with appropriate time and length scales should be superimposed on the mean open boundaries conditions on inflow. Presently, this is not the case in Delft3D-FLOW, although its open boundary conditions allow for time series or harmonics. In most cases, the internal topography or geometry (breakwaters etc.) is such that hydrodynamic instabilities develop inside the flow domain but relevant exceptions exist such as the flow in a straight channel with constant depth. The latter flow case is marginally unstable i.e. wavy velocity perturbations superimposed on the mean flow create secondary instabilities. This type of flow needs the upstream turbulence perturbations as open boundary conditions on inflow. To that purpose, a generator of synthetic time series of spatially correlated (coherent eddy structures) turbulence fluctuations is recommended.

The high-pass filter Eq. B.6 is designed for stationary flows only but it is also applicable to weakly non-stationary flows with eddy turnover times significantly smaller than the time scale of the base flow's boundary conditions. For tidal flows, with turbulent horizontal eddies superimposed, the high-pass filter can be improved e.g. through linear regression based on the flow's history. The fluctuations $(\dots)^*$ then fall into the category of so-called *turbulence with stationary first increments* for which much of the same theory as for stationary turbulence applies. Experience on typical time scales and sensitivity of the eddies for τ must determine whether improvements on the high-pass filter are advisable.

In [Equation B.2](#), k is the wave number magnitude of the truncating wave length below which [Equation B.2](#) is designed to represent the net effect of sub-grid turbulence and its relation to a horizontally isotropic grid size reads:

$$k_s = \frac{\pi f_{lp}}{\Delta}; \quad f_{lp} \leq 1. \quad (\text{B.8})$$

where f_{lp} is the spatial low-pass filter coefficient that depends on the spatial filtering properties

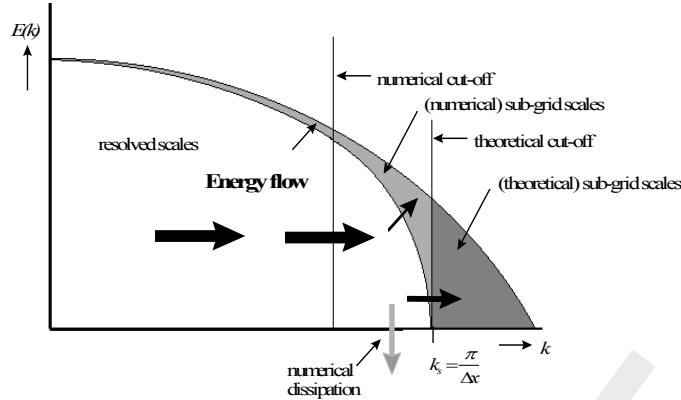


Figure B.11: Definition of truncation wave numbers due to resolution and numerical damping

Table B.2: Default parameter settings for the SGS model

Parameter	Value	Input parameter	Description
α	5/3	Htural	Slope in log-log spectrum
n_D	2	Hturnd	Dimensional number
σ_T	0.7	Hurst	Prandtl-Schmidt number
f_{lp}	0.3	Hurlp	Spatial low-pass filter coefficient
τ	($)^*$	Hurrt	Relaxation time

of the advection scheme. For the ideal scheme, $f_{lp} = 1$ holds without any damping up to the shortest resolvable wave length (2Δ), see [Figure B.11](#). For anisotropic grids we propose, in agreement with others, to consider the cell area i.e.

$$\frac{1}{k_s^2} = \frac{\Delta x \Delta y}{(\pi f_{lp})^2}, \quad (\text{B.9})$$

Moreover, the derivation of Eq. [B.2](#) is generally applicable for 2D as well as 3D turbulence ($n_D = 2$ or 3) with slope α in the log-log spectrum of energy density $E(k)$ (usually $\alpha = 3$ or $5/3$ holds) as a function of wave number magnitude k and these properties are introduced through γ in [Equation B.2](#) that yields:

$$\gamma = I_\infty \sqrt{\frac{1 - \alpha^{-2}}{2n_D}}, \quad I_\infty = 0.844. \quad (\text{B.10})$$

The parameters to the SGS model that are user-defined input are shown in [Table B.2](#) together with their recommended values.

Where $($)^*$$ is typically twice the largest time scale of the eddies.

The value for f_{lp} is determined semi-empirically i.e. it is obtained by previous HLES experiments ([Jagers and Van Schijndel, 2000](#)) but it is partially founded on theoretical considerations of the 2nd order ADI central-upwind advection scheme of Delft3D-FLOW ([Stelling, 1984](#)).

For a proper use of HLES with Delft3D-FLOW, the following three time step limitations hold:

- 1 Courant number CFL_{BT} for surface waves (barotropic mode) less than $4\sqrt{2}$:

$$CFL_{BT} = 2\sqrt{2} \frac{\Delta t \sqrt{gH}}{\Delta x} \leq 4\sqrt{2}. \quad (\text{B.11})$$

- 2 Courant number CFL_{conv} for convection less than unity:

$$CFL_{conv} = \frac{U \Delta t}{\Delta x} \leq 1. \quad (\text{B.12})$$

- 3 Explicit integration of SGS stresses:

$$\Delta t \leq \frac{(\Delta x)^2}{2\nu_{SGS}}. \quad (\text{B.13})$$

Usually, the most restrictive time step limitation is due to the barotropic Courant number CFL_{BT} . This restriction is based on accuracy and near-conservation of energy rather than stability because the non-viscous part of the entire solution procedure is unconditionally stable. Note that the advection scheme of Delft3D-FLOW does not strictly conserve kinetic energy. However, it approximates this property quite well for $CFL_{BT} < 4\sqrt{2}$, as demonstrated by [Van Vossen \(2000, Chapter 6\)](#).

In view of a usually small Froude number, the second convection/advection condition is usually satisfied when $CFL_{BT} < 4\sqrt{2}$. In view of HLES, we call the second limitation a *space-time consistence condition* stating that the spatial increments (time step \times velocity) in transporting eddies should be in agreement or consistent with their minimal resolvable size. If *space-time consistence condition* is violated then the large time step destroys the proper simulation of the deformation of the transported eddy by shearing of other eddies or the mean flow.

Finally, the third limitation is rarely violated and it is the single one referring to pure numerical stability rather than to accuracy.

B.9 Sediment transport and morphology

Sediment transport and morphology is supported by the FLOW-GUI, see Data Groups *Processes* and *Physical parameters - Sediment, Morphology*.

Keywords	Value	Description	Default
Filsed	#name.sed#	Sediment input file; define sediment fractions and other input quantities	none
Filmor	#name.mor#	Morphological input file; define parameters for the morphodynamic computation	none
TraFrm	#name.tra#	Sediment transport input file; define sediment transport formula and its coefficients	none

This feature concerns the use of sediment transport in Delft3D-FLOW. With the feedback of bed level changes to the hydrodynamic computation you can execute a full morphodynamic computation. You can include the influence of waves by running this version of Delft3D-FLOW in combination with the Delft3D-WAVE module. This sediment transport feature offers the following functionalities:

- ◊ Cohesive sediment transport, including the effect of salinity on flocculation.
- ◊ Non-cohesive suspended sediment (sand) transport.
- ◊ Bedload transport, including effect of wave asymmetry.
- ◊ Influence of waves and hindered settling.
- ◊ Updating the bed-level and feedback to the hydrodynamics.
- ◊ Up to 99 suspended fractions (combinations of sand or mud) and arbitrary number of bedload fractions.
- ◊ Numerous sediment transport formulations

File contents	See the following sections.
Filetype	ASCII
File format	Free format.
Generated	FLOW-GUI.



Remarks:

- ◊ Sediment names are case sensitive!
- ◊ The files should not contain tabs.
- ◊ All suspended sediment fractions must have been defined as a constituent with special names that begin with "Sediment" (e.g. "Sediment 1").
- ◊ For these constituents, boundary conditions should be prescribed which can be overwritten by equilibrium concentrations using one of the options described in the morphodynamic input file.
- ◊ For these constituents initial conditions and concentrations at sources should be specified.

B.9.1 Sediment input file

The sediment input file contains the characteristics of all sediment fractions. In the record description the name of the quantities are given to simplify their reference in the formulas given in [chapter 11](#).



Remarks:

- ◊ There are several versions of the <*.sed> file. The keyword based file is described in [Table B.3](#). FLOW-GUI v3.39.14 and higher reads and saves the keyword based file. The file with no keywords can also be read by this GUI. FLOW can handle both types of files.
- ◊ FLOW-GUI v3.39.25 and higher writes keyword `SdbUni` not anymore. Instead a new keyword, `IniSedThick` is written.

Table B.3: Sediment input file with keywords

Keyword	Record description
SedimentFileInfo	
FileCreatedBy	contains version number of FLOW-GUI
FileCreationDate	creation date and time of the <*.sed> file
FileVersion	version number
SedimentOverall	
continued on next page	

Table B.3 – continued from previous page

Keyword	Record description
TopSus	option for determining suspended sediment diameter (see subsection 11.3.1) (integer)
Cref	reference density (CSOIL) for hindered settling calculations (not active when $\geq 1 \cdot 10^6$) [kg/m ³] (1 real)
MudCnt	mud fraction (only if none of the sediment fractions is of type mud) uniform value (1 real) or file <*.mdc> with spatially varying values at cell centres (1 string)
Sediment	
Name	name between # as specified in NamC in mdf-file
SedTyp	type of sediment; must be “sand”, “mud” or “bedload”: (1 string) “sand” and “mud” fractions need to be specified before “bedload” fractions.
RhoSol	specific density of sediment fraction [kg/m ³] (1 real)
SedDxx	xx percentile sediment diameter (for sand or bedload) where xx can take on values from 01 to 99 [m] (1 real)
SedMinDia	minimum sediment diameter (for sand or bedload) [m] (1 real)
SedDia	median sediment diameter (for sand or bedload) equivalent to SedD50 [m] uniform value (1 real) or file <*.d50> with spatially varying values at cell centres (1 string)
SedSg	geometric standard deviation of sediment diameter (for sand or bedload) [m] (1 real)
SedMaxDia	maximum sediment diameter (for sand or bedload) [m] (1 real)
SalMax	salinity for saline settling velocity (for mud) [ppt] (1 real)
WS0	settling velocity fresh water (for mud) [m/s] (1 real)
WSM	settling velocity saline water (for mud) [m/s] (1 real)
continued on next page	

Table B.3 – continued from previous page

Keyword	Record description
TcrSed	critical stress for sedimentation [N/m ²] uniform value (1 real) or file <*.tcd> with spatially varying values at cell centres (1 string)
TcrEro	critical stress for erosion [N/m ²] uniform value (1 real) or file <*.tce> with spatially varying values at cell centres (1 string)
EroPar	erosion parameter [kg/m ² s] uniform value (1 real) or file <*.ero> with spatially varying values at cell centres (1 string)
CDryB	dry bed density [kg/m ³] (1 real)
SdBUni or IniSedThick	initial sediment mass at bed per unit area [kg/m ²] initial sediment layer thickness at bed [m] uniform value (1 real) or filename <*.sdb> with non-uniform values at cell centres (1 string)
FacDSS	factor for determining suspended sediment diameter $D_{sus} = FACDSS * SEDDIA$ (for sand) [-] (1 real)
TraFrm	Name of fraction specific sediment transport formula (for sand or bedload) (1 string) This formula overrules the default formula given by the TraFrm keyword in the MDF file.
SettleLib	Name of the shared object (.dll on Windows, or .so on Linux) without file extension in which the user-defined settling velocity routine is located (1 string). Only to be specified in combination with SettleFunction.
SettleFunction	Name of the user-defined subroutine which provides the settling velocity (1 string). Only to be specified in combination with SettleLib. See subsection B.9.6 for a description of the subroutine interface.
SettleInput	Optional file name to be passed to SettleFunction (1 string). Only to be specified in combination with SettleFunction.
The Sediment block can be repeated for other fractions	

**Remarks:**

- ◊ The SedDia parameter may only be specified as a file if the simulation includes only one sediment fraction.

- ◇ There are a couple of options to specify the characteristic diameter(s) for sand or bed-load fractions. **Table B.4** gives an overview of the various options, the associated sediment distribution types and the implicit assumptions.

Table B.4: Options for sediment diameter characteristics

Specified quantities	Assumptions
SedDia (uniform value) or SedD50 (uniform value)	Piecewise log-uniform distribution SedD10 = 0.75 SedD50 SedD90 = 1.5 SedD50
SedDia (filename) or SedD50 (filename)	Lognormal distribution (spatially varying grain size) SedSg = 1.34
SedDia (filename) or SedD50 (filename), SedSg	Lognormal distribution (spatially varying grain size)
SedDxx (any xx), SedSg	Lognormal distribution
Two SedDxx values	Lognormal distribution SedSg computed from xx and SedDxx
SedMinDia, SedMaxDia	Loguniform distribution
SedMinDia or SedMaxDia, One SedDxx value	Loguniform distribution
More than two SedDxx, SedMinDia, SedMaxDia values	Piecewise loguniform distribution
Other combinations	not allowed

Example of a version 2 file, with keywords:

```
[SedimentFileInfo]
  FileCreatedBy      = Delft3D-FLOW-GUI, Version: 3.39.14.03
  FileCreationDate = Thu Dec 08 2005, 14:47:46
  FileVersion       = 02.00
[SedimentOverall]
  IopSus           = 0                               Suspended sediment size is Y/N
                                                       calculated dependent on d50
  Cref             = 1.60e+03 [kg/m3]            CSoil Reference density for hindered
                                                       settling
[Sediment]
  Name              = #Sediment sand#          Name as specified in NamC in mdf-file
  SedTyp            = sand                   Must be "sand", "mud" or "bedload"
  RhoSol            = 2.6500000e+003 [kg/m3] Specific density
  SedDia            = 2.0000000e-004 [m]        Median sediment diameter (D50)
  CDryB             = 1.6000000e+003 [kg/m3] Dry bed density
  IniSedThick       = 0.50e+000 [m]         Initial sediment layer thickness at bed
                                                       (uniform value or file name)
  FacDSS            = 1.0e+0               [-]   FacDss*SedDia = Initial suspended
                                                       sediment diameter.
[Sediment]
  Name              = #Sediment mud#          Name as specified in NamC in mdf-file
```

SedTyp	= mud	Must be "sand", "mud" or "bedload"
RhoSol	= 2.6500000e+003 [kg/m ³]	Specific density
SalMax	= 0.0000000e+000 [ppt]	Salinity for saline settling velocity
WS0	= 2.5000001e-004 [m/s]	Settling velocity fresh water
WSM	= 2.5000001e-004 [m/s]	Settling velocity saline water
TcrSed	= 1.0000000e+003 [N/m ²]	Critical stress for sedimentation
TcrEro	= 5.0000000e-001 [N/m ²]	Critical stress for erosion
EroPar	= 9.999997e-005 [kg/m ² s]	Erosion parameter
CDryB	= 5.0000000e+002 [kg/m ³]	Dry bed density
IniSedThick	= 0.05e+000 [m]	Initial sediment layer thickness at bed (uniform value or file name)
FacDSS	= 1.00e+00	[$-$] FacDss*SedDia = Initial suspended sediment diameter

Table B.5: Sediment input file without keywords

Record	Record description
1	VERSION: version number (in the present Delft3D Release, this line should read “* Version 01”)
2	LSED: number of sediment fractions CSOIL: reference density for hindered settling calculations (not active when $\geq 1 \cdot 10^6$) [kg/m ³] IOPSUS: option for determining suspended sediment diameter (see subsection 11.3.1) (1 integer, 1 real, 1 integer)
3	SEDTYP: type of sediment: “sand” or “mud” (1 string)
4	RHOSOL: specific density of sediment fraction [kg/m ³] (1 real)
5	SEDDIA: median sediment diameter (for sand) [m] (1 real)
6	SALMAX: salinity for saline settling velocity (for mud) [ppt] (1 real)
7	WS0: settling velocity fresh water (for mud) [m/s] (1 real)
8	WSM: settling velocity saline water (for mud) [m/s] (1 real)
9	TCDUNI: critical stress for sedimentation (uniform value) [N/m ²] (1 real) or FILTCD: file with spatially varying values of TCDUNI <*.tcd> at cell centres (1 string)
10	TCEUNI: critical stress for erosion (uniform value) [N/m ²] (1 real) or FILTCE: file with spatially varying values of TCEUNI <*.tce> at cell centres (1 string)
11	EROUNI: erosion parameter (uniform value) [kg/m ² s] (1 real) or FILERO: file with spatially varying values of EROUNI <*.ero> at cell centres (1 string)
12	CDRYB: dry bed density [kg/m ³] (1 real)
13	SDBUNI: initial sediment mass at bed per unit area [kg/m ²] (1 real) or FILSDB: filename with non-uniform values of SDBUNI <*.sdb> at cell centres (1 string)

Table B.5: Sediment input file without keywords

Record	Record description
14	FACDSS: factor for determining suspended sediment diameter Dsus = FACDSS*SEDDIA (for sand) [-] (1 real)
15 to 27	repeat records 3 to 14 for a second sediment fraction; repeat this block as many times as there are sediment fractions (max. LSED)

Remarks:

- ◊ If no version number is specified, the program will read records 2 to 13 from the input file, and set FACDSS = 1. This has been done to ensure backward compatibility.
- ◊ The filenames for non-uniform parameters have a mandatory extension as indicated in [Table B.5](#).
- ◊ Records 6–11 only have meaning for mud fractions (but have to be included for sand as well).
- ◊ Record 5 only has meaning for sand fractions (but has to be included in case of mud).

**Example of a version 1 file, with no keywords:**

This example concerns the transport of two sediment fractions one sand (with a $D_{50} = 140$ micron) and one mud (with a spatially-varying critical stress for erosion, defined in the file <critstr.tce>). Comments are printed between brackets.

```
* Version 01
2 1600. 2 {LSED : no. of sediment, CSOIL, IOPSUS }
sand {SEDTYP : type of sediment}
2650.0 {RHOSOL : density sediment [kg/m3]}
0.00014 {SEDDIA : median sediment diameter [m]}
0.0000 {SALMAX : salinity saline water [ppt]}
0.0000 {WS0 : settling velocity fresh water [m/s]}
0.0000 {WSM : settling velocity saline water [m/s]}
0.0000 {TCDUNI : critical stress for sedimentation [N/m2]}
0.0000 {TCEUNI : critical stress for erosion [N/m2]}
0.0000 {EROUNI : Erosion parameter [kg/m2/s]}
1600.0 {CDRYB : dry bed density [kg/m3]}
800.0 {SDBUNI : Uniform value of init. mass on bed [kg/m2]}
0.8 {FACDSS : factor for suspended sediment diameter [-]}
mud {SEDTYP : type of sediment}
2650.0 {RHOSOL : density sediment [kg/m3]}
0.0000 {SEDDIA : median sediment diameter [m]}
10.000 {SALMAX : salinity saline water [ppt]}
0.0005 {WS0 : settling velocity fresh water [m/s]}
0.0020 {WSM : settling velocity saline water [m/s]}
0.1000 {TCDUNI : critical stress for sedimentation [N/m2]}

critstr.tce
0.0001 {EROUNI : Erosion parameter [kg/m2/s]}
300.0 {CDRYB : dry bed density [kg/m3]}
800.0 {SDBUNI : Uniform value of init. mass on bed [kg/m2]}
1.0 {FACDSS : factor for suspended sediment diameter [-]}
```

Remark:

- ◊ Additional comments in the input file are not mandatory.

**Restrictions:**

- ◊ Records specifying an attribute filename should not contain additional comment text.
- ◊ The values of the parameters are not checked against their domains.

Attribute files in the sediment input file

In the sediment input file 4 optional attribute files can be defined for spatially varying data:

FILTCD	critical stress for sedimentation [N/m ²].
FILTCE	critical stress for erosion [N/m ²].
FILER0	erosion parameter [kg/m ² s].
FILSDB	initial sediment mass at the bed per unit area [kg/m ²].

These (ASCII) files have the same format as the depth file. See [subsection A.2.4](#):

Record description:

File type	Record description
Free formatted	Parameter values per row, starting at N = 1 to N = Nmax, separated by one or more blanks. The number of continuation lines is determined by the number of grid points per row (Mmax) and the maximum record size of 132.



Restrictions:

- ◊ The maximum record length in the free formatted file is 132.
- ◊ Parameter values from the file will not be checked against their domain.
- ◊ The input items are separated by one or more blanks.

B.9.2 Morphology input file

The morphological input file contains additional information necessary for a morphodynamic run. Like the <*.sed> file there is the (old) version without keywords and a newer version with keywords.

Table B.6: Morphological input file with keywords

Keyword	Record description
MorphologyFileInfo	
FileCreatedBy	contains version number of FLOW-GUI
FileCreationDate	creation date and time of the <*.mor> file
FileVersion	version number
Morphology	
continued on next page	

Table B.6 – continued from previous page

Keyword	Record description
MorFac	morphological scale factor constant (1 real) or file with time-dependent values (string) in case of a file: no text may be used after the filename
MorStt	time interval in Tunit after the start of the simulation after which morphological changes will be calculated (1 real)
Thresh	threshold sediment thickness for reducing sediment exchange [m] (1 real)
MorUpd	update bed level during flow run (1 logical: false or true) Keyword to be replaced by BedUpd.
BedUpd	update bed level during flow run (1 logical: false or true)
CmpUpd	update bed composition during flow run (1 logical: false or true)
NeglectEntrainment	compute bed level changes from suspended sediment transport rates instead of from the entrainment and deposition terms; neglect the entrainment of suspended sediment in the mass balance (1 logical: false or true)
EqmBc	Neumann boundary condition for non-cohesive suspended sediment concentrations at inflow boundaries (1 logical: false or true) Keyword to be replaced by NeuBcSand
NeuBcSand	Neumann boundary condition for non-cohesive suspended sediment concentrations at inflow boundaries (1 logical: false or true)
NeuBcMud	Neumann boundary condition for cohesive suspended sediment concentrations at inflow boundaries (1 logical: false or true)
DensIn	include effect of sediment on density gradient (1 logical: false or true)
AksFac	Van Rijn's reference height = AKSFAC * KS (1 real)
continued on next page	

Table B.6 – continued from previous page

Keyword	Record description
RWave	multiplication factor estimated ripple height; wave related roughness = RWAVE * estimated ripple height (1 real) NOTE: estimated ripple height now fixed at 0.025 m.
Sus	current-related suspended sediment transport factor (1 real)
Bed	current-related bedload transport factor (1 real)
SusW	wave-related suspended sediment transport factor (1 real)
BedW	wave-related bedload sediment transport factor (1 real)
SedThr	minimum depth for sediment calculations (1 real)
ThetSD	global / maximum dry cell erosion factor (1 real)
HMaxTH	maximum depth for variable THETSD (1 real)
FWFac	tuning parameter for wave streaming (1 real)
EpsPar	Only for waves in combination with k-epsilon turbulence model; Van Rijn's parabolic mixing coefficient (1 logical)
TopKCW	option for ks and kw (1 real)
RDC	current related roughness ks (1 real)
RDW	wave related roughness kw (1 real)
UpdInf	update bed levels at inflow boundaries (1 logical: false or true)
Multi	"Mormerge": run multiple morphological simulations in parallel and continuously merge the morphological developments (1 logical: false or true). See https://svn.oss.deltares.nl/repos/delft3d/trunk/examples/05_mormerge
DzMax	maximum bed level change per time step expressed as percentage of water depth (1 real)
CaMax	maximum equilibrium sediment concentration in Van Rijn formula (1 real)
continued on next page	

Table B.6 – continued from previous page

Keyword	Record description
OldMudFrac	if true, do not include the fractional reduction factor in the mud entrainment term (1 logical: false or true)
IHidExp	hiding and exposure formulation number (1 integer in the range 1 to 5) If the simulation includes a mud fraction, only 1 is allowed. 1 no hiding and exposure 2 Egiazaroff formulation 3 Ashida & Michiue formulation 4 Parker et al. formulation 5 Wu, Wang & Jia formulation
ASKLHE	hiding and exposure parameter for formulation 4 (1 real)
MWWJHE	hiding and exposure parameter for formulation 5 (1 real)
Espir	effect of spiral flow on bedload transport (1 real)
ISlope	bed slope formulation (1 integer in the range 1 to 4) 1 no bed slope 2 Bagnold formulation 3 Koch & Flokstra formulation 4 Parker & Andrews formulation
AlfaBs	longitudinal bed gradient factor for bedload transport (1 real) used by all formulations except formulation 1
AlfaBn	transverse bed gradient factor for bedload transport for formulation 2 (1 real)
Ashld	multiplicative factor Ash for bed slope formulation 3 (1 real)
Bshld	power Bsh for bed slope formulation 3 (1 real)
Cshld	power Csh for bed slope formulation 3 (1 real)
Dshld	power Dsh for bed slope formulation 3 (1 real)
CoulFri	Coulomb friction for bed slope formulation 4 (1 real)
F1FdRat	Lift/drag force ratio for bed slope formulation 4 (1 real)
ThetaCr	Critical shear stress for bed slope formulation 4 (1 real)
continued on next page	

Table B.6 – continued from previous page

Keyword	Record description
BcFil	file containing morphological boundary conditions (1 string)
Boundary	
Name	name of boundary as specified in BND file (1 string)
IBedCond	bedload or bed level boundary condition (1 integer in the range 0 to 5) 0 no bed level constraint 1 bed level fixed 2 depth specified as function of time 3 depth change specified as function of time 4 bedload transport rate prescribed (volume rate of bed material) 5 bedload transport rate prescribed (volume rate of stone)
ICmpCond	bed composition boundary condition (1 integer in the range 0 to 3) 0 no bed composition constraint 1 composition constant 2 mass fractions specified as function of time 3 volume fractions specified as function of time
the Boundary block can be repeated for other boundaries	
Numerics	
UpwindBedload	use upwind bedload (true, default) or central bedload (false)
LaterallyAveragedBedload	smoothed bedload transport rates (1 logical: false or true). Default false.
MaximumWaterdepth	use locally maximum water depth to compute characteristic velocity for sediment transport at cell centre (1 logical: false or true). Default false.
Output	
	All keywords in this block determine whether the named quantity is to be written to the map-file. Unless indicated otherwise, the value should be 1 logical and the default is false.
continued on next page	

Table B.6 – continued from previous page

Keyword	Record description
VelocAtZeta	characteristic velocity vector at cell centres
VelocMagAtZeta	velocity magnitude at cell centres
VelocZAtZeta	height of characteristic velocity at cell centres
ShearVeloc	shear velocity u_*
BedTranspAtFlux	bedload transport (due to currents and waves) plus suspended transport due to waves at staggered cell faces. Default true.
BedTranspDueToCurrentsAtZeta	bedload transport due to currents at cell centres
BedTranspDueToCurrentsAtFlux	bedload transport due to currents at staggered cell faces
BedTranspDueToWavesAtZeta	bedload transport due to waves at cell centres
BedTranspDueToWavesAtFlux	bedload transport due to waves at staggered cell faces
SuspTranspAtFlux	depth integrated suspended load transport minus near bed corrective transport at staggered cell faces. Default true.
SuspTranspDueToWavesAtZeta	suspended load transport due to waves at cell centres
SuspTranspDueToWavesAtFlux	suspended load transport due to waves at staggered cell faces
NearBedTranspCorrAtFlux	near bed corrective transport at staggered cell faces
EquilibriumConcentration	equilibrium concentration at which entrainment equals deposition. Only for 2D simulations. Default true.
NearBedRefConcentration	near bed reference concentration. Default true.
SettlingVelocity	settling velocity. Default true.
SourceSinkTerms	source and sink terms
Bedslope	bed slope vector
Taurat	shear stress ratio
Bedforms	bed form roughnesses
continued on next page	

Table B.6 – continued from previous page

Keyword	Record description
Dm	arithmetic mean sediment diameter
Dg	geometric mean sediment diameter
Frac	sediment fraction in top layer
MudFrac	mud fraction in top layer
FixFac	non-erodible layer reduction factor
HidExp	hiding and exposure factor
Percentiles	percentiles of characteristic sediment diameter to be written to the map-file (default: 10, 50 and 90)
CumNetSedimentationFlux	accumulated net sedimentation over the period between two map-file output times
BedLayerSedimentMass	mass of each sediment fraction per unit area per bed layer. Default: true.
BedLayerDepth	depth of bed layer interfaces. Default: true.
BedLayerVolumeFractions	volume fraction of each sediment fraction per bed layer. Default: true.
BedLayerPorosity	porosity per bed layer. Only if IPorosity>0. Default: true.
StatWaterDepth	The value lists the statistics to be computed and stored for the water depth over the period between two output times (1 string). The program will check whether the following substrings occur in the value string: min, max, mean, and std for minimum, maximum, mean, and standard deviation respectively. Default: none.
StatVelocity	The value lists the statistics to be computed and stored for the depth averaged velocity over the period between two output times (1 string). The program will check whether the following substrings occur in the value string: min, max, mean, and std for minimum, maximum, mean, and standard deviation respectively. Default: none.
continued on next page	

Table B.6 – continued from previous page

Keyword	Record description
StatBedload	The value lists the statistics to be computed and stored for the total bed load (accumulated over all sediment fractions) over the period between two output times (1 string). The program will check whether the following substrings occur in the value string: min, max, mean, and std for minimum, maximum, mean, and standard deviation respectively. Default: none.
StatSusupload	The value lists the statistics to be computed and stored for the total suspended load (accumulated over all sediment fractions) over the period between two output times (1 string). The program will check whether the following substrings occur in the value string: min, max, mean, and std for minimum, maximum, mean, and standard deviation respectively. Default: none.
TranspType	sediment transport unit (1 integer in the range 0 to 2): 0 kg/(s m) 1 m ³ bed material/(s m) 2 m ³ stone/(s m) (default option)
AverageAtEachOutputTime	cumulative average of sediment transport rates at each map output time step instead of only at end of simulation
Underlayer	
IUnderLyr	stratigraphy schematisation model (1 integer, either 1 or 2) 1 one well mixed layer 2 multiple layers
MxNULyr	maximum number of underlayers (excluding transport and base layers) in case IUnderLyr=2 (1 integer)
TTLForm	transport layer thickness formulation in case IUnderLyr=2 (1 integer in the range 1 to 3) 1 constant user-defined 2 proportional to water depth 3 proportional to bedform height
ThTrLyr	thickness of transport layer in case of TTLForm=1 uniform or (1 real) or file <*.ttl> with spatially varying values at cell centres (1 string)
TTLAlpha	proportionality constant in case of TTLForm=2 or 3
continued on next page	

Table B.6 – continued from previous page

Keyword	Record description
TTLMin	minimum thickness in case of TTLForm=2 or 3
ThUnLyr	characteristic maximum thickness of stratigraphy layers in case IUnderLyr=2 (1 real)
UpdBaseLyr	update option for composition and thickness of base layer (1 integer in the range 1 to 4) <ul style="list-style-type: none"> 1 composition and thickness computed 2 composition constant, thickness computed 3 composition copied from layer above, thickness computed 4 composition and thickness constant
IniComp	file containing initial bed composition (1 string) usage of this keyword overrules the IniSedThick (or older SdBUni) values given in the sed file (see subsection B.9.9)

For these boundary conditions you need to specify the imposed time-series in the file referred to using the `BcFil` keyword. File format described in [subsection B.9.7](#).

Example of a version 2 file, with keywords:

```
[MorphologyFileInformation]
  FileCreatedBy      = Delft3D-FLOW-GUI, Version: 3.39.14.03
  FileCreationDate  = Thu Dec 08 2005, 14:47:50
  FileVersion        = 02.00
[Morphology]
  EpsPar    = false           Only for waves in combination with k-eps
  IopKCW    = 1                Flag for determining Rc and Rw
  RDC       = 0.01             [m] Current related roughness (IopKCW = 0)
  RDW       = 0.02             [m] Wave related roughness (IopKCW = 0)
  MorFac    = 1.0000000e+000   [-] Morphological scale factor
  MorStt    = 7.20e+02         [min] Spin-up interval from TStart till start of morph changes
  Thresh    = 5.0e-002         [m] Threshold sed thickness for reducing sed
                                 exchange
  MorUpd   = true             Update bathymetry during flow run
  EqmBc    = true             Equilibrium concentration at inflow
                                 boundaries
  DensIn   = false            Include effect of sediment on density
                                 gradient
  AksFac   = 1.000000e+000   [-] Van Rijn's reference height = AKSFAC * KS
  RWave    = 2.0e+0           [-] Wave related roughness = RWAVE * estimated
                                 ripple height. Van Rijn Recommends range 1-3
  AlfaBs   = 1.00e+00         [-] Longitudinal bed gradient factor for bedload transport
  AlfaBn   = 1.50e+00         [-] Transverse bed gradient factor for bedload transport
  Sus      = 1.0e+0            [-] Multipl fac for suspended sed reference concentration
  Bed      = 1.0e+00           [-] Multipl factor for bedload transport vector magnitude
  SusW    = 1.0e+000           [-] Wave-related suspended sed. transport factor
  BedW    = 1.00e+000          [-] Wave-related bedload sed. transport factor
  SedThr  = 1.00e-001         [m] Minimum threshold depth for sediment computations
  ThetSD   = 0.00e+000         [-] Fraction of erosion to assign to adjacent dry cells
  HMaxTH  = 1.50e+00          [m] Max depth for var THETSD. Set < SEDTHR to use global
```

FWFac = 1.0e+00 [-] Only for waves in combination with k-eps model

Table B.7: Morphological input file without keywords

Record	Record description
1	VERSION: version number (in the present Delft3D Release, this line should read “* Version 01”)
2	MORFAC: morphological scale factor (1 real)
3	MORSTT: time interval in minutes after the start of the simulation after which morphological changes will be calculated (1 real)
4	THRESH: threshold sediment thickness for reducing sediment exchange [m] (1 real)
5	MORUPD: update bathymetry during flow run (1 logical: .false. or .true.)
6	EQMBC: equilibrium sediment concentration profile at inflow boundaries (1 logical: .false. or .true.)
7	DENSIN: include effect of sediment on density gradient (1 logical: .false. or .true.)
8	AKSFAC: Van Rijn's reference height = AKSFAC * KS (1 real)
9	RWAVE: multiplication factor estimated ripple height; wave related roughness = RWAVE * estimated ripple height (1 real) NOTE: estimated ripple height now fixed at 0.025 m.
10	ALFABS: longitudinal bed gradient factor for bedload transport (1 real)
11	ALFABN: transverse bed gradient factor for bedload transport (1 real)
12	SUS: current-related suspended sediment transport factor (1 real)
13	BED: current-related bedload transport factor (1 real)
14	SUSW: wave-related suspended sediment transport factor (1 real)
15	BEDW: wave-related bedload sediment transport factor (1 real)
16	SEDTHR: minimum depth for sediment calculations (1 real)
17	THETSD: global / maximum dry cell erosion factor (1 real)
18	HMAXTH: maximum depth for variable THETSD (1 real)
19	FWFAC: tuning parameter for wave streaming (1 real)
20	EPSPAR: Van Rijn's parabolic mixing coefficient (1 logical)

Table B.7: Morphological input file without keywords

Record	Record description
21	IOPKCW: option for ks and kw, RC: current related roughness ks, RW: wave related roughness kw

Example version 1 file with no keywords:

This example concerns a morphological simulation in which the bed-level updating commences 6250 minutes after the start of the simulation. Comments are printed between brackets.

```
* Version 01
180.          {MORFAC : morphological scale factor}
6250.         {MORSTT : time interval in min after start, for bed updating}
0.2           {THRESH : threshold sed. thickness for reducing sed. exchange}
.true.         {MORUPD : update bathymetry during flow run}
.true.         {EQMBC : equilibrium concentration at inflow boundaries}
.true.         {DENSSIN : include effect of sediment on density gradient}
1.0           {AKSFAC : Van Rijn's reference height = AKSFAC * KS}
2.0           {RWAVE : multiplication factor estimated ripple height}
1.0           {ALFABS : long. bed gradient factor for bedload transport}
1.5           {ALFABN : tran. bed gradient factor for bedload transport}
1.0           {SUS : multiplication factor sus. sed. ref. concentration}
1.0           {BED : multiplication factor bedload transp. vector magn.}
1.0           {SUSW : wave-related suspended sediment transport factor}
1.0           {BEDW : wave-related bedload sediment transport factor}
0.25          {SEDETHR : minimum depth for sediment calculations}
1.0           {THETSD : global / maximum dry cell erosion factor}
0.01          {HMAXTH : maximum depth for variable THETSD}
0.1            {FWFAC : tuning parameter for wave streaming}
.false.        {EPSPAR : Van Rijn's parabolic-linear mixing distribution}
1 0.01 0.02  {IOPKCW, RDC, RDW : parameters for determining morphological roughness}
```

**Remarks:**

- ◊ If no version number is specified, the program will read records 2 to 19 from the input file, and set EPSPAR = FALSE and IOPKCW = 1. This has been done to ensure backward compatibility.
- ◊ If EPSPAR is set to TRUE, Van Rijn's parametric mixing distribution is also used if the $k-\varepsilon$ model is selected in FLOW. If it is set to FALSE, the mixing is set equal to the ε of the $k-\varepsilon$ model.
- ◊ IOPKCW is a switch to determine ks and kw:

IOPKCW = 0 : ks = RDC and kw = RDW

IOPKCW = 1 : ks is taken from current-related effective roughness height as determined in the FLOW (spatially varying) and $k_w = RWAVE \cdot \Delta_r$ (see subsection 11.3.3).

- ◊ The file for specifying bedload, bed level and/or bed composition boundary conditions is described in subsection B.9.7.

**Restriction:**

- ◊ The values of the parameters are not checked against their domains.

Output enhancements:

The following parameters or (time dependent) data have been added to the result files:

Result file	Data
History file	Sediment concentrations, bedload transport, suspended transport, cumulative bedload and suspended transports in monitoring points and integrated over cross-sections, near-bed reference concentrations
Map file	Sediment concentrations per layer, bedload and suspended transport, instantaneous and averaged, time-varying bathymetry, available sediment per fraction, cumulative erosion/sedimentation, near-bed reference concentrations, as well as any additional quantity selected in the output block of the morphology input file (as indicated in Table B.6)

Remarks:

- ◊ Use the MATLAB interface or Delft3D-QUICKPLOT to plot all these parameters.
- ◊ Use the parameter DENSIN to switch on or off the effect of the sediment concentration on the density and thereby its influence on the turbulence (hindered settling).
- ◊ You can use the utility DATSEL to read these values from file. This requires the following modifications to the <files.def>-file in directory <\$D3D_HOME\ARCH\flow\default>. For PC this directory is usually located at: <c:\delft3d\w32\flow\default>. The section about the 'trim'-file should read:

```
'trim-'
'XCOR','YCOR','ALFAS','map-const'
10
'initial bottom depth',      'DP0' , ' ',   'map-const' , 'CODB', 'TEMPOUT','d'
'depth water level points', 'DP0' , ' ',   'map-const' , 'KCS','map-const','s'
'water level',               'S1'  , ' ',   'map-series','KCS','map-const','c'
'velocity',                  'U1'  , 'V1' ,   'map-series','KCS','map-const','u'
'bottom stress',             'TAUKSI','TAUETA','map-series','KCS','map-const','u'
'thickness of bed layer',    'DPSED' , ' ',   'map-sed-series','KCS','map-const','c'
'time-varying depth',        'DPS' , ' ',   'map-sed-series','KCS','map-const','c'
'bottom sediment kg/m2',     'BODSED' , ' ',   'map-sed-series','KCS','map-const','c'
'bedload transport',         'SBUU' , 'SBVV' , 'map-sed-series','KCS','map-const','u'
'susp. load transport',       'SSUU' , 'SSVV' , 'map-sed-series','KCS','map-const','u'
```

B.9.3 Sediment transport input file

By default, the formulations of [Van Rijn et al. \(2000\)](#) are applied for the suspended and bed-load transport of non-cohesive sediment. In addition this feature offers a number of extra sediment transport relations for non-cohesive sediment; [Table B.8](#) gives an overview of those additional formulae. You can also supply a library with your own favourite sediment transport formula and use one of those, but more about that in [Section B.9.4](#). Let's focus now on the set of predefined alternative formulations.

If you want to use one of these formulae, you must create a sediment transport input file <*.tra>. The filename of the sediment input file must be specified in the MDF-file (formula used for all non-cohesive sediment fractions) or in the sediment input file (formula used for only the selected non-cohesive sediment fraction) using the keyword TraFrm. In the former case, use the Data Group *Addition parameters* in the FLOW-GUI with keyword and value: TraFrm=#name.tra#. For these pre-defined alternative transport relations the sediment transport input file should comply with the following specifications:

Table B.8: Additional transport relations

Formula	Bedload	Affected by Waves	IFORM
11.5.1, Van Rijn (1993)	Bedload + suspended	Yes	-1
11.5.2, Engelund-Hansen (1967)	Total transport	No	1
11.5.3, Meyer-Peter-Muller (1948)	Total transport	No	2
11.5.4, General formula	Total transport	No	4
11.5.5, Bijker (1971)	Bedload + suspended	Yes	5
11.5.6, Van Rijn (1984)	Bedload + suspended	No	7
11.5.7, Soulsby/Van Rijn	Bedload + suspended	Yes	11
11.5.8, Soulsby	Bedload + suspended	Yes	12
11.5.9, Ashida–Michiue (1974)	Total transport	Yes	14
11.5.10, Wilcock–Crowe (2003)	Bedload	No	16
11.5.11, Gaeuman et al. (2009) laboratory calibration	Bedload	No	17
11.5.12, Gaeuman et al. (2009) Trinity River calibration	Bedload	No	18
User-defined	Bedload + suspended	Yes	—

- ◊ The file may start with an arbitrary number of lines not containing the text IFORM.
- ◊ Then a line starting with sediment transport formula number IFORM and containing text IFORM.
- ◊ Then an arbitrary number of lines starting with an asterisk (*) may follow.
- ◊ Then a line starting with the number sign (#) followed by a transport formula number optionally followed by text identifying the transport formula for the user. The next lines should contain the parameter values of the transport formula coefficients: one parameter value per line optionally followed by text identifying the parameter. There may be an arbitrary number of blocks starting with # in the file, but exactly one should correspond to the transport formula number IFORM specified above.

An example file for transport formula 5 referred to as “Bijker (1971)” is provided below. The following table lists the parameters to be specified in the sediment transport input file for each separate transport formula.

Table B.9: Transport formula parameters

Formula	Parameter	Unit
11.5.1, Van Rijn (1993)	none	
11.5.2, Engelund-Hansen (1967)	calibration coefficient α bed roughness height r_k (dummy)	- m
11.5.3, Meyer-Peter-Muller (1948)	calibration coefficient α dummy argument	- NA
11.5.4, General formula	calibration coefficient α power b power c ripple factor or efficiency factor μ critical Shields' mobility parameter θ_c	- - - -
11.5.5, Bijker (1971)	calibration coefficient b for shallow water BS	-

Formula	Parameter	Unit
	calibration coefficient b for deep water BD	-
	shallow water (h_w/h) criterion C_s	-
	deep water (h_w/h) criterion C_d	-
	dummy argument	NA
	bed roughness height r_c	m
	settling velocity w	m/s
	porosity ε	-
	wave period T_{user} (used if computed wave period $< 10^{-6}$)	s
11.5.6, Van Rijn (1984)	calibration coefficient α_1	-
	dummy argument	NA
	reference level ξ_c	m
	settling velocity w_s	m/s
11.5.7, Soulsby/Van Rijn	calibration coefficient A_{cal}	-
	D_{90}/D_{50} ratio	-
	z_0 roughness height	m
11.5.8, Soulsby	calibration coefficient A_{cal}	-
	model index modind	-
	D_{50}/z_0 ratio χ	-
11.5.9, Ashida–Michiue (1974)	calibration coefficient α	-
	critical Shields' mobility parameter θ_c	-
	power m	-
	power p	-
	power q	-
11.5.10, Wilcock–Crowe (2003)	none	-
11.5.11, Gaeuman et al. (2009) laboratory calibration	calibration coefficient θ_{c0}	-
11.5.12, Gaeuman et al. (2009) Trinity River calibration	calibration coefficient α_0	-
User-defined	calibration coefficient θ_{c0}	-
User-defined	calibration coefficient α_0	-
User-defined	none	-

Remarks:

- ◊ Van Rijn (1993) does not require any additional parameters. Only the transport formula number (-1) followed by the string `IFORM` is required.
- ◊ The user-defined transport formula requires a keyword based transport input file as described below.
- ◊ The keyword `IFORM` must be present in the same line as the formula number.
- ◊ The file should not contain tabs.

If you want to use your own sediment transport formula, the transport input file should use a new keyword based file format described by Table B.10. This file points to a shared library (on Windows commonly referred to as a DLL) and the subroutine in that library that should be called to get the transport rates. The library may contain one or more sediment transport formulae. The interface of the subroutine depends on the type of sediment ("sand", "bedload"

Table B.10: Sediment transport formula input file with keywords

Keyword	Record description
TransportFormulaFileInformation	
FileCreatedBy	contains version number of FLOW-GUI
FileCreationDate	creation date and time of the <*.tra> file
FileVersion	version number, currently 01.00
TransportFormula	
Name	name of transport formula (string)
DLL	library name without <.dll> or <.so> extension (string)
Function	function name within library to call
InputFile	optional file name to be passed to library

or “mud”) for which the formula is to be used. For sand fractions the routine should return `sbc_total`, `sbc`, `sbcu`, `sbcv`, `equi_conc`, `cesus`, `ssus`, `sbwu`, `sbwv`, `sswu`, `sswv` and `t_relax`. Even though bedload fractions don’t have a suspended load component, the subroutine for such sediment fractions should return the same list of quantities as for the case of sand fractions. For mud fractions the routine should return `source` and `sink`. The meaning of these output arguments is described in the Sections [B.9.4](#) and [B.9.5](#) for sand and mud fractions, respectively.

Example for Bijker formula:

This is an example of a sediment transport input file to be used with the Bijker formulation.

```

5           Number of transport formula   IFORM
*-----
#5          BIJKER with wave effect
 5          -Par 1- Coefficient shallow water    BS
 2          -Par 2- Coefficient deep water      BD
 .4         -Par 3- Shallow water limit        CRITS
 .05        -Par 4- Deep water limit          CRITD
 .000300    -Par 5- D90 grain size          D90
 .05        -Par 6- Bottom roughness height   RK
 .016       -Par 7- Particle fall velocity   W
 .4         -Par 8- Porosity                 POR
 7.0        -Par 9- Wave Period             T
# End of specification of transport relation

```

Note that only the number 5, the marker #5 and the values on the next lines are required. All the other text is optional.

Example for Van Rijn 1993 formula:

This is an example of a sediment transport input file to be used with the [Van Rijn \(1993\)](#) formulations in which case there is no need for additional parameters. The file could simply read:

-1 IFORM

Example for user-defined transport formula:

This is an example of the Engelund Hansen transport formula implemented as a subroutine EngHan in a library <MyTransportLibrary.dll> or <MyTransportLibrary.so> depending on the operating system. The DLL points to the appropriate library file (which should be located in the work directory or somewhere on the Windows search path). The Function points to the appropriate subroutine in the library file.

```
[TransportFormulaFileInformation]
  FileCreationDate = Thu Dec 08 2005, 14:47:50
  FileVersion      = 01.00
[TransportFormula]
  Name      = My Engelund Hansen implementation
  DLL       = MyTransportLibrary
  Function  = EngHan
```

B.9.4 User defined transport routine for sand or bedload fractions

This section gives an example code for a user-defined transport routine for sand or bedload fractions. The interface of the routine for mud fractions is slightly different; it is described in subsection B.9.5. This particular example implements the [Engelund and Hansen \(1967\)](#) formula as a total load transport formula; the code can be found in the source directory of the Delft3D installation on your computer.

The order and way in which arguments are passed to external functions differs among FORTRAN compilers. Most compilers allow you to change these settings, however, if these settings do not match the settings of the calling program the program will crash when trying to call the subroutine in the supplied library. It is, therefore, important to make sure that the settings that you use to compile the library match the settings used when compiling delftflow. It is therefore recommended to generate the library using the same compiler as delftflow; at this moment this is the Intel Fortran 11 compiler² using default settings for a multithreaded program.

Most of the code is straightforward and standard Fortran, so it should be fairly easy to understand. Some statements may, however, require some explanation. The !DEC\$ statement is read by the compiler and makes sure the subroutine name is visible to the outside world. The real type “hp” is defined as a double precision quantity: even if the simulation runs in single precision, the communication with the library occurs in double precision hence all input and output arguments of the subroutine should be defined to be of the type “real(hp)”.

The input quantities are provided using three linear arrays, namely a real array, an integer array and a character array. Each element of these arrays represents a different quantity; see the example for a description of the various input items. You may either use the various elements directly in your formulae or define local variables as done in the example; the latter approach keeps your own code more readable and separate from the Delft3D interface. The array approach for the input items allows us to add extra input parameters to the end of the arrays without breaking compatibility with older routines.

```
subroutine enghan(dll_integers, max_integers, &
                  dll_reals   , max_reals   , &
                  dll_strings , max_strings , &
```

²If you can't use this compiler then make sure that the following minimum conditions are met: multi-threaded library with calling convention “C, by reference” and “string length arguments after all arguments”.

```

        sbc_total, sbc , sbcu, sbcv, sbwu, sbwv      , &
        equi_conc, cesus, ssus, sswu, sswv, t_relax , &
        error_message   )
!DEC$ ATTRIBUTES DLLEXPORT, ALIAS: 'ENGHAN' :: ENGHAN
!!--description-----
!
! Computes sediment transport according to
! Engelund and Hansen (1967)
!
!!--pseudo code and references-----
! NONE
!!--declarations-----
implicit none
!
! Local constants
! Interface is in high precision
!
integer      , parameter :: hp    = kind(1.0d0)
!
! Subroutine arguments: input
!
integer          , intent(in)  :: max_integers
integer          , intent(in)  :: max_reals
integer          , intent(in)  :: max_strings
integer          , dimension(max_integers), intent(in) :: dll_integers
real(hp)         , dimension(max_reals)   , intent(in) :: dll_reals
character(len=256), dimension(max_strings), intent(in) :: dll_strings
!
! Subroutine arguments: output
!
real(hp)         , intent(out) :: sbc           ! bedload due to currents, magnitude [m3/m/s]
real(hp)         , intent(out) :: sbcu          ! bedload due to currents, m component [m3/m/s]
real(hp)         , intent(out) :: sbcv          ! bedload due to currents, n component [m3/m/s]
real(hp)         , intent(out) :: sbwu          ! bedload due to waves, m component [m3/m/s]
real(hp)         , intent(out) :: sbwv          ! bedload due to waves, n component [m3/m/s]
real(hp)         , intent(out) :: cesus          ! susp load concentration [m3/m3]
real(hp)         , intent(out) :: ssus           ! susp load due to currents, magnitude [m3/m/s]
real(hp)         , intent(out) :: sswu           ! susp load due to waves, m component [m3/m/s]
real(hp)         , intent(out) :: sswv           ! susp load due to waves, n component [m3/m/s]
real(hp)         , intent(out) :: t_relax         ! relaxation time in 2D mode [s]
character(len=256), intent(out) :: error_message ! not empty: echo and stop run
logical          , intent(out) :: equi_conc       ! true: contration cesus returned by formula
logical          , intent(out) :: mudfrac         ! false: susp load ssus returned by formula
logical          , intent(out) :: sbc_total        ! true: bedload magnitude returned by formula
                                                ! false: bedload components returned
!
! Local variables for input parameters
!
integer          :: l
integer          :: m
integer          :: n, nm
real(hp)         :: ag
real(hp)         :: chezy
real(hp)         :: d10, d50, d90, dss, dstar
real(hp)         :: h, hidexp, hrms
real(hp)         :: mudfrac
real(hp)         :: rhosol, rhowat, rlabda
real(hp)         :: sal
real(hp)         :: taub, tem, teta, timsec, tp
real(hp)         :: u, umod, uorb, utot, uuu
real(hp)         :: v, vicmol, vvv
real(hp)         :: ws
real(hp)         :: zumod
character(len=256) :: runid
!
! Local variables

```

```

!
real(hp)          :: acal
real(hp)          :: cf
real(hp)          :: delta
real(hp)          :: ustar
!
!! extract array variables -----
!
if (max_integers < 4) then
    error_message = 'Insufficient integer values provided by delftflow'
    return
endif
nm      = dll_integers( 1 ) ! nm index of the grid cell
m       = dll_integers( 2 ) ! m index of the grid cell
n       = dll_integers( 3 ) ! n index of the grid cell
l       = dll_integers( 4 ) ! number of the sediment fraction in the computation
!
if (max_reals < 30) then
    error_message = 'Insufficient real values provided by delftflow'
    return
endif
timsec  = dll_reals( 1 )   ! current time since reference time [s]
u       = dll_reals( 2 )   ! m component of effective depth-averaged velocity [m/s]
v       = dll_reals( 3 )   ! n component of effective depth-averaged velocity [m/s]
utot    = dll_reals( 4 )   ! magnitude of effective depth-averaged velocity [m/s]
uuu    = dll_reals( 5 )   ! m component of characteristic velocity [m/s]
vvv    = dll_reals( 6 )   ! n component of characteristic velocity [m/s]
umod   = dll_reals( 7 )   ! magnitude of characteristic velocity [m/s]
zumod  = dll_reals( 8 )   ! height above bed of characteristic velocity [m]
h       = dll_reals( 9 )   ! water depth [m]
chezy  = dll_reals(10)   ! local Chezy value [m1/2/s]
hrms   = dll_reals(11)   ! wave height [m]
tp     = dll_reals(12)   ! wave period [s]
teta   = dll_reals(13)   ! angle between wave dir and local grid orientation [deg]
rlabda = dll_reals(14)   ! wave length [m]
uorb   = dll_reals(15)   ! orbital velocity at the bed [m/s]
d50    = dll_reals(16)   ! sediment diameter of fraction [m]
dss    = dll_reals(17)   ! sediment diameter of fraction when in suspension [m]
dstar  = dll_reals(18)   ! critical dimensionless grain size parameter [-]
d10    = dll_reals(19)   ! 10-percentile diameter of local sediment mixture [m]
d90    = dll_reals(20)   ! 90-percentile diameter of local sediment mixture [m]
mudfrac= dll_reals(21)  ! mud fraction [-]
hidexp  = dll_reals(22)  ! hiding & exposure factor [-]
ws     = dll_reals(23)   ! settling velocity [m/s]
rhosol = dll_reals(24)   ! solid sediment density [kg/m3]
rhowat = dll_reals(25)   ! local water density [kg/m3]
sal    = dll_reals(26)   ! local salinity [ppt]
tem    = dll_reals(27)   ! local water temperature [degC]
ag     = dll_reals(28)   ! gravitational acceleration [m/s2]
vicmol = dll_reals(29)   ! molecular viscosity of water [m2/s]
taub   = dll_reals(30)   ! bed shear stress [N/m2]
!
if (max_strings < 2) then
    error_message = 'Insufficient strings provided by delftflow'
    return
endif
runid  = dll_strings( 1 ) ! user-specified run-identification
filenn = dll_strings( 2 ) ! user-specified file name (keyword: InputFile)
!
!! executable statements -----
!
write(*,*) 'plugin_delftflow_traform.dll : enghan : called'
! The output argument error_message MUST have value '' to continue the calculation.
!
error_message = ''
!
```

```

! If you want to indicate that this subroutine has encountered some invalid input or
! encountered some unexpected situation, you can set the error_message to a non-empty
! string. This error_message will then be shown in the log file of the calling program
! and the simulation will abort. This is shown by the next line, remove it to enable
! this subroutine.
!
! error_message = 'Use the Engelund Hansen formula inside \DFLOW'
!
! Set some parameters and compute derivative quantities.
!
delta = (rhosol - rhowat) / rhowat ! relative density of sediment particle
acal = 0.6_hp
cf = ag/chezy/chezy
ustar = sqrt(cf)*utot
!
! Set the transport rates and concentrations.
! Note: all transport quantities should be given in m3 solid material per s.
!       i.e. rate (or concentration) in kg/s =
!             rhosol * specified rate (or concentration)
!
sbc_total = .true. ! set flag to indicate that bedload magnitude is given
sbc = acal*0.05_hp*utot*ustar**4/ag**2/sqrt(cf)/delta**2/d50
                    ! bedload magnitude is non-zero
sbcu = 0.0_hp          ! bedload component, m direction (here dummy since sbc_total is true)
sbcv = 0.0_hp          ! bedload component, n direction (here dummy since sbc_total is true)
!
! There should be no suspended load.
!
equi_conc = .false. ! set flag to indicate that susp load magnitude is given
cesus = 0.0_hp        ! suspended load concentration (here dummy since equi_conc is false)
ssus = 0.0_hp          ! zero suspended load transport
!
! This formula does not include wave driven transport.
!
sbwu = 0.0_hp          ! bedload transport, m direction due to waves is zero
sbwv = 0.0_hp          ! bedload transport, n direction due to waves is zero
sswu = 0.0_hp          ! suspended load transport, m direction due to waves is zero
sswv = 0.0_hp          ! suspended load transport, n direction due to waves is zero
!
! Since there is no suspended load, also the relaxation time for depth-averaged models can be
! set to zero.
!
t_relax = 0.0_hp        ! relaxation time is zero
end subroutine enghan
!$
```

For the bedload (due to currents) the routine may either return the total transport rate (`sbc`) or the two transport components (`sbcu` and `sbcv`) in the two grid directions m and n . The logical flag `sbc_total` indicates which values the routine has set (obviously true for the former choice and false for the latter choice). In the former case the sediment transport direction is automatically set equal to the near bed velocity direction in a 3D model and equal to the depth-averaged velocity direction corrected for secondary flow (see [Equation 11.44](#)) in a 2D model.

For the suspended load (due to currents) there are again two options: the routine may either return the (equilibrium) suspended sediment concentration (`cesus`) or the (equilibrium) suspended transport rate (`ssus`). The logical flag `equi_conc` indicates which value the routine has set (obviously true for the former choice and false for the latter choice). For sand fractions the (equilibrium) suspended sediment concentration will be computed in the latter case as S_{sus}/uH where u is the depth-averaged velocity and H is the water depth. For bedload fractions the reverse is done: if `cesus` is given then the suspended sediment transport rate is computed by multiplying it with uH .

If you want to distinguish between flow and wave driven transports, you may set the wave driven bedload and suspended load rate components separately using `sbwu`, `sbwv`, `sswu` and `sswv`. It should be noted that the suspended load due to currents (specified as either `cesus` or `ssus`) will be used to force the advection-diffusion equation for the suspended load. The suspended load due to waves (specified using `sswu` and `sswv`) is treated as a regular explicit sediment flux.

Remark:

- ◊ All transport quantities should be given in [m^3] solid material, i.e. excluding any pore volumes.



In 3D simulations the advection and diffusion of sediment in the vertical direction results in a relaxation behaviour; the concentrations are imposed near the bed and it takes time for the model to reach the equilibrium concentration. In 2D simulations the processes in the vertical direction are lacking; for such applications a relaxation time is needed to reproduce this behaviour. You can specify it here as `t_relax`; if `t_relax` is equal to zero or if it is negative then the program will use the [Galappatti \(1983\)](#) formula.

B.9.5 User defined transport routine for mud fractions

This section gives an example code for a user-defined transport routine for mud fractions. The interface of the routine for sand and bedload fractions is slightly different; it is described in [subsection B.9.4](#). This particular example implements the source and sink terms of [Partheniades \(1965\)](#) formulae; the code can be found in the source directory of the Delft3D installation on your computer.

The order and way in which arguments are passed to external functions differs among FORTRAN compilers. Most compilers allow you to change these settings, however, if these settings do not match the settings of the calling program the program will crash when trying to call the subroutine in the supplied library. It is, therefore, important to make sure that the settings that you use to compile the library match the settings used when compiling delftflow. It is therefore recommended to generate the library using the same compiler as delftflow; at this moment this is the Intel Fortran 11 compiler³ using default settings for a multithreaded program.

Most of the code is straightforward and standard Fortran, so it should be fairly easy to understand. Some statements may, however, require some explanation. The `!DEC$` statement is read by the compiler and makes sure the subroutine name is visible to the outside world. The real type “hp” is defined as a double precision quantity: even if the simulation runs in single precision, the communication with the library occurs in double precision hence all input and output arguments of the subroutine should be defined to be of the type “real(hp)”.

The input quantities are provided using three linear arrays, namely a real array, an integer array and a character array. Each element of these arrays represents a different quantity; see the example for a description of the various input items. The input parameters for the mud transport formula are the same as those for the sand and bedload fractions, however, a number of input parameters are not defined in case of mud, namely the characteristic grain diameters D_{10} , D_{50} , D_{90} , D_s , D_* , the mud fraction and the hiding and exposure factor ξ . You may either use the various elements directly in your formulae or define local variables as done in the example; the latter approach keeps your own code more readable and separate from the Delft3D interface. The array approach for the input items allows us to add extra input

³If you can't use this compiler then make sure that the following minimum conditions are met: multi-threaded library with calling convention “C, by reference” and “string length arguments after all arguments”.

parameters to the end of the arrays without breaking compatibility with older routines.

```

subroutine parkro(dll_integers, max_integers, &
                  dll_reals   , max_reals   , &
                  dll_strings , max_strings , &
                  sink        , source      , &
                  error_message )
!DEC$ ATTRIBUTES DLLEXPORT, ALIAS: 'PARKRO' :: PARKRO
!--description-----
!
! Computes source and sink terms according to
! Partheniades and Krone (1965)
!
!--pseudo code and references-----
! NONE
!--declarations-----
implicit none
!
! Local constants
! Interface is in high precision
!
integer      , parameter :: hp    = kind(1.0d0)
!
! Subroutine arguments: input
!
integer                      , intent(in)  :: max_integers
integer                      , intent(in)  :: max_reals
integer                      , intent(in)  :: max_strings
integer      , dimension(max_integers), intent(in) :: dll_integers
real(hp)       , dimension(max_reals)  , intent(in) :: dll_reals
character(len=256), dimension(max_strings), intent(in) :: dll_strings
!
! Subroutine arguments: output
!
real(hp)      , intent(out) :: source      ! source term [kg/m2/s]
real(hp)      , intent(out) :: sink        ! sink term [-] (to be multiplied with concen
character(len=256), intent(out) :: error_message ! not empty: echo and stop run
!
! Local variables for input parameters
!
integer      :: l
integer      :: m
integer      :: n, nm
real(hp)     :: ag
real(hp)     :: chezy
real(hp)     :: d10, d50, d90, dss, dstar
real(hp)     :: h, hidexp, hrms
real(hp)     :: mudfrac
real(hp)     :: rhosol, rhowat, rlabda
real(hp)     :: sal
real(hp)     :: taub, tem, teta, timsec, tp
real(hp)     :: u, umod, uorb, utot, uuu
real(hp)     :: v, vicmol, vvv
real(hp)     :: ws
real(hp)     :: zumod
character(len=256) :: runid
!
! Local variables
!
real(hp)     :: eropar
real(hp)     :: taum, tcrero, tcrcdep
!
!-- extract array variables -----
!
if (max_integers < 4) then
  error_message = 'Insufficient integer values provided by delftflow'

```

```

        return
    endif
nm      = dll_integers( 1 ) ! nm index of the grid cell
m       = dll_integers( 2 ) ! m index of the grid cell
n       = dll_integers( 3 ) ! n index of the grid cell
l       = dll_integers( 4 ) ! number of the sediment fraction in the computation
!
if (max_reals < 30) then
    error_message = 'Insufficient real values provided by delftflow'
    return
endif
timsec = dll_reals( 1 )      ! current time since reference time [s]
u       = dll_reals( 2 )      ! m component of effective depth-averaged velocity [m/s]
v       = dll_reals( 3 )      ! n component of effective depth-averaged velocity [m/s]
utot   = dll_reals( 4 )      ! magnitude of effective depth-averaged velocity [m/s]
uuu    = dll_reals( 5 )      ! m component of characteristic velocity [m/s]
vvv    = dll_reals( 6 )      ! n component of characteristic velocity [m/s]
umod   = dll_reals( 7 )      ! magnitude of characteristic velocity [m/s]
zumod  = dll_reals( 8 )      ! height above bed of characteristic velocity [m]
h      = dll_reals( 9 )      ! water depth [m]
chezy  = dll_reals(10)       ! local Chezy value [m1/2/s]
hrms   = dll_reals(11)       ! wave height [m]
tp     = dll_reals(12)       ! wave period [s]
teta   = dll_reals(13)       ! angle between wave dir and local grid orientation [deg]
rlabda = dll_reals(14)       ! wave length [m]
uorb   = dll_reals(15)       ! orbital velocity at the bed [m/s]
!
!---- the following parameters are undefined for mud fractions -----
d50    = dll_reals(16)       ! sediment diameter of fraction [m]
dss    = dll_reals(17)       ! sediment diameter of fraction when in suspension [m]
dstar  = dll_reals(18)       ! critical dimensionless grain size parameter [-]
d10    = dll_reals(19)       ! 10-percentile diameter of local sediment mixture [m]
d90    = dll_reals(20)       ! 90-percentile diameter of local sediment mixture [m]
mudfrac = dll_reals(21)      ! mud fraction [-]
hidexp = dll_reals(22)       ! hiding & exposure factor [-]
!---- the parameters above are undefined for mud fractions -----
!
ws     = dll_reals(23)       ! settling velocity [m/s]
rhosol = dll_reals(24)       ! solid sediment density [kg/m3]
rhowat = dll_reals(25)       ! local water density [kg/m3]
sal    = dll_reals(26)       ! local salinity [ppt]
tem    = dll_reals(27)       ! local water temperature [degC]
ag     = dll_reals(28)       ! gravitational acceleration [m/s2]
vicmol = dll_reals(29)       ! molecular viscosity of water [m2/s]
taub   = dll_reals(30)       ! bed shear stress [N/m2]
!
if (max_strings < 2) then
    error_message = 'Insufficient strings provided by delftflow'
    return
endif
runid  = dll_strings( 1 )    ! user-specified run-identification
filenm = dll_strings( 2 )    ! user-specified file name (keyword: InputFile)
!
!! executable statements -----
!
write(*,*) 'plugin_delftflow_traform.dll : parkro : called'
! The output argument error_message MUST have value '' to continue the calculation.
!
error_message = ''
!
! If you want to indicate that this subroutine has encountered some invalid input or
! encountered some unexpected situation, you can set the error_message to a non-empty
! string. This error_message will then be shown in the log file of the calling program
! and the simulation will abort. This is shown by the next line, remove it to enable
! this subroutine.
!
```

```

! error_message = 'Use the Partheniades Krone formula inside \DFLOW'
!
! Set some parameters and compute derivative quantities.
!
tcrero = 0.1_hp
tcrdep = 0.1_hp
eropar = 2.5e-6_hp
!
! Compute source and sink terms
!
taum = max(0.0_hp, taub/tcrero - 1.0_hp)
source = eropar * taum
if (tcrdep > 0.0_hp) then
    sink = max(0.0_hp, 1.0_hp-taub/tcrdep)
else
    sink = 0.0_hp
endif
end subroutine parkro

```

Compared to the subroutine for sand and bedload fractions (described in subsection B.9.4) this routine has very few output arguments, namely only `source` and `sink`. The source term will be used explicitly and should be specified in kg (mud) per unit area (m^2) per unit of time (s). The sink term is treated implicitly and reads $\text{sink} \times c^{t+1} w_s$ where c^{t+1} is the concentration at the new time level in [kg/m^3] and w_s is the local settling velocity in [m/s]; hence the quantity `sink` to be returned by the routine is dimensionless.



Remarks:

- ◊ The unit for the source term is [$\text{kg}/\text{m}^3/\text{s}$].
- ◊ The sink quantity is dimensionless since it will be multiplied with the concentration in [kg/m^3] and the settling velocity in [m/s] to give the actual sink term in the equation.

B.9.6 User defined routine for the settling velocity

This section gives an example code for a user-defined routine that computes the settling velocity. This particular example implements a simple salinity dependent settling formula for mud, but the same routine interface can also be used for sand fractions.

The order and way in which arguments are passed to external functions differs among FORTRAN compilers. Most compilers allow you to change these settings, however, if these settings do not match the settings of the calling program the program will crash when trying to call the subroutine in the supplied library. It is, therefore, important to make sure that the settings that you use to compile the library match the settings used when compiling delftflow. It is therefore recommended to generate the library using the same compiler as delftflow; at this moment this is the Intel Fortran 11 compiler⁴ using default settings for a multithreaded program.

Most of the code is straightforward and standard Fortran, so it should be fairly easy to understand. Some statements may, however, require some explanation. The `!DEC$` statement is read by the compiler and makes sure the subroutine name is visible to the outside world. The real type "hp" is defined as a double precision quantity: even if the simulation runs in single precision, the communication with the library occurs in double precision hence all input and output arguments of the subroutine should be defined to be of the type "real(hp)".

The input quantities are provided using three linear arrays, namely a real array, an integer

⁴If you can't use this compiler then make sure that the following minimum conditions are met: multi-threaded library with calling convention "C, by reference" and "string length arguments after all arguments".

array and a character array. Each element of these arrays represents a different quantity; see the example for a description of the various input items. The input parameters for the mud transport formula are the same as those for the sand and bedload fractions, however, two input parameters are not defined in case of mud, namely D_{50} and D_s . You may either use the various elements directly in your formulae or define local variables as done in the example; the latter approach keeps your own code more readable and separate from the Delft3D interface. The array approach for the input items allows us to add extra input parameters to the end of the arrays without breaking compatibility with older routines.

```

subroutine settle(dll_integers, max_integers, &
                 dll_reals , max_reals , &
                 dll_strings , max_strings , &
                 ws         , &
                 error_message )
!DEC$ ATTRIBUTES DLLEXPORT, ALIAS: 'SETTLE' :: SETTLE
!!--description-----
!
! Computes simple salinity dependent settling velocity.
!
!!--pseudo code and references-----
! NONE
!!--declarations-----
implicit none
!
! Local constants
! Interface is in high precision
!
integer      , parameter :: hp    = kind(1.0d0)
!
! Subroutine arguments: input
!
integer          , intent(in)  :: max_integers
integer          , intent(in)  :: max_reals
integer          , intent(in)  :: max_strings
integer          , dimension(max_integers), intent(in)  :: dll_integers
real(hp)        , dimension(max_reals)   , intent(in)  :: dll_reals
character(len=256), dimension(max_strings) , intent(in)  :: dll_strings
!
! Subroutine arguments: output
!
real(hp)      , intent(out) :: ws           ! settling velocity [m/s]
character(len=256), intent(out) :: error_message ! not empty: echo and stop run
!
! Local variables for input parameters
!
integer      :: k
integer      :: l
integer      :: m
integer      :: n, nm
real(hp)     :: ag
real(hp)     :: cconc, chezy, conc, csoil
real(hp)     :: d50, dss
real(hp)     :: epstur
real(hp)     :: h
real(hp)     :: ktur
real(hp)     :: rhosol, rhowat
real(hp)     :: sal
real(hp)     :: tem, timsec
real(hp)     :: u, um
real(hp)     :: v, vicmol, vm
real(hp)     :: w
character(len=256) :: runid
!
! Local variables

```

```

!
real(hp)      :: f
real(hp)      :: pi
real(hp)      :: salmax
real(hp)      :: ws0, ws1
!
!! extract array variables -----
!
if (max_integers < 5) then
    error_message = 'Insufficient integer values provided by delftflow'
    return
endif
nm      = dll_integers( 1 ) ! nm index of the grid cell
m       = dll_integers( 2 ) ! m index of the grid cell
n       = dll_integers( 3 ) ! n index of the grid cell
k       = dll_integers( 4 ) ! layer number of the grid cell
l       = dll_integers( 5 ) ! number of the sediment fraction in the computation
!
if (max_reals < 21) then
    error_message = 'Insufficient real values provided by delftflow'
    return
endif
timsec   = dll_reals( 1 )      ! current time since reference time [s]
u        = dll_reals( 2 )      ! m component of the local velocity [m/s]
v        = dll_reals( 3 )      ! n component of the local velocity [m/s]
w        = dll_reals( 4 )      ! vertical component of the local velocity [m/s]
sal      = dll_reals( 5 )      ! local salinity [ppt]
tem      = dll_reals( 6 )      ! local water temperature [degC]
rhowat  = dll_reals( 7 )      ! local water density [kg/m3]
conc    = dll_reals( 8 )      ! sediment concentration [kg/m3] of considered fraction
cconc   = dll_reals( 9 )      ! total sediment concentration [kg/m3] over all fractions
ktur    = dll_reals(10)       ! turbulent kinetic energy [m2/s2]
epstur  = dll_reals(11)       ! turbulent dissipation [m2/s3]
!
!---- the following parameters are undefined for mud fractions -----
d50      = dll_reals(12)      ! sediment diameter of fraction [m]
dss      = dll_reals(13)      ! sediment diameter of fraction when in suspension [m]
!---- the parameters above are undefined for mud fractions -----
!
rhosol   = dll_reals(14)      ! solid sediment density [kg/m3]
csoil    = dll_reals(15)      ! reference density for hindered settling calculations [kg/m3]
ag       = dll_reals(16)      ! gravitational acceleration [m/s2]
vicmol   = dll_reals(17)      ! molecular viscosity of water [m2/s]
h        = dll_reals(18)      ! water depth [m]
um       = dll_reals(19)      ! m component of effective depth-averaged velocity [m/s]
vm       = dll_reals(20)      ! n component of effective depth-averaged velocity [m/s]
chezy   = dll_reals(21)      ! local Chezy value [m1/2/s]
!
if (max_strings < 2) then
    error_message = 'Insufficient strings provided by delftflow'
    return
endif
runid   = dll_strings( 1 )    ! user-specified run-identification
filenm  = dll_strings( 2 )    ! user-specified file name (keyword: SettleInput)
!
!! executable statements -----
!
write(*,*) 'plugin_delftflow_traform.dll : settle : called'
! The output argument error_message MUST have value '' to continue the calculation.
!
error_message = ''
!
! If you want to indicate that this subroutine has encountered some invalid input or
! encountered some unexpected situation, you can set the error_message to a non-empty
! string. This error_message will then be shown in the log file of the calling program
! and the simulation will abort. This is shown by the next line, remove it to enable

```

```

! this subroutine.
!
! error_message = 'Use the settling velocity routine inside \DFLOW'
!
! Set some parameters and compute derivative quantities.
!
ws1      = 5.0e-4_hp      ! settling velocity at S >= S_max
ws0      = 5.0e-4_hp      ! settling velocity at S = 0
salmax   = 30.0_hp
!
pi = 4.0_hp * atan(1.0_hp)
!
! Compute settling velocity
!
if (sal > salmax) then
    f = 1.0_hp
else
    f = 0.5_hp * (1 - cos(pi*sal/salmax))
endif
ws = ws0 + (ws1 - ws0) * f
end subroutine settle

```

Compared to the subroutine for sand and bedload fractions (described in subsection B.9.4) this routine has very few output arguments, namely only `source` and `sink`. The source term will be used explicitly and should be specified in kg (mud) per unit area (m^2) per unit of time (s). The sink term is treated implicitly and reads `sink` times $c^{t+1}w_s$ where c^{t+1} is the concentration at the new time level in [kg/m^3] and w_s is the local settling velocity in [m/s]; hence the quantity `sink` to be returned by the routine is dimensionless.

Remarks:



- ◊ The unit for the source term is [$\text{kg}/\text{m}^3/\text{s}$].
- ◊ The sink quantity is dimensionless since it will be multiplied with the concentration in [kg/m^3] and the settling velocity in [m/s] to give the actual sink term in the equation.

B.9.7 Sediment transport and morphology boundary condition file

File contents	Time-series for sediment transport and morphology boundary conditions
Filetype	ASCII
File format	Fix format for header information; free format for time-series data.
Filename	< <i>name.bcm</i> >
Generated	manually offline

Record description:

For each open boundary segment that according to the boundary characteristics given in the `<*.mor>` file requires boundary data, the data is given in two related blocks:

- ◊ A header block containing a number of compulsory and optional keywords accompanied by their values.
- ◊ A data block containing the time dependent data.

Description header block:

Text	Required	Value
table-name	no	arbitrary string
location	yes	'name boundary section' (see < <i>name.bnd</i> > file)
time-function	no	{'non-equidistant'} or 'equidistant'
time-step	yes	time step only in case of time-function 'equidistant'
reference-time	yes	yyyymmdd, yyymmdd hhmmss or 'from model'
time-unit	no	'years', 'decades', 'days', 'hours', {'minutes'}, 'seconds', 'ddhhmmss' or 'date'
interpolation	no	{'linear'} or 'block'
extrapolation	no	'periodic', 'constant' or {'none'}
parameter	yes	'parameter name and location' units '[ccc]'
records-in-table	no	number of times/lines in the data block

**Remarks:**

- ◊ Default parameter values are indicated in braces.
- ◊ Reference-time not required if time-unit equals 'date'.
- ◊ Unit strings are currently not interpreted by Delft3D-FLOW.

The 'parameter name and location' strings depend on the boundary type chosen, i.e. quantity type to be specified. The following table lists the base parameter names. The full 'parameter name and location' string is a concatenation of the indicated base parameter name, optionally followed by a single space character and the user-defined sediment name, optionally followed by 'end A' and 'end B' location strings. Bedload transport should be specified for only the non-mud fractions (i.e. sand and bedload fractions only) whereas bed composition should be specified for all fractions. The parameters for multiple sediment fractions must occur in the same order as in which the sediment fractions have been defined. The location strings 'end A' and 'end B' can be used to indicate that the specified quantity linearly varies along the boundary; in case of a parameter for multiple sediment fractions, all parameter-sediment pairs must be specified for both 'end A' and 'end B' and all 'end A' conditions should precede the 'end B' conditions.

Boundary type	Base name	parameter	Unit	Multiplicity
sediment transport and bed level				
0: free	none		-	-
1: fixed	none		-	-
2: time series	depth		m	1
3: depth change prescribed	depth change		m/s	1
4: transport incl pores prescribed	transport incl pores		m ³ /s/m	#nonmud
5: transport excl pores prescribed	transport excl pores		m ³ /s/m	#nonmud
bed composition				
0: free	none		-	-
1: fixed	none		-	-
2: mass fraction prescribed	mass fraction		-	#sediment
3: volume fraction prescribed	volume fraction		-	#sediment

Description data block:

Record	Record description
each record	Time in time-units after the reference-time and followed by as many values as parameters have been defined in the description block (all reals). In case of time-function 'equidistant', the first (time) column should be dropped. In case of time-unit 'date' the date and time should be specified as one string using the format: yyyyymmddhhmmss.

Remarks:

- ◊ Maximum record length is 512.
- ◊ The morphological boundary conditions will only be used at inflow boundaries.
- ◊ Bed composition boundary conditions are only relevant if you do not impose incoming bedload transport.
- ◊ The parameter name of the column should read 'time'.



Example:

Model with morphological boundary conditions imposed at two boundaries. At both boundaries the depth and mass fractions for two sediment fractions 'Sediment fine' and 'Sediment coarse' respectively are prescribed. The file structure allows you to either combine boundary conditions in one table, or to use individual tables. The example shows both approaches.

```

location          'East Boundary'
time-function    'non-equidistant'
reference-time   19941001
time-unit        'minutes'
interpolation    'linear'
parameter        'time' unit '[min]'
parameter        'depth' unit '[m]'
parameter        'mass fraction Sediment fine' unit '[-]'
parameter        'mass fraction Sediment coarse' unit '[-]'
      0.0000  5.0     0.3     0.7
      2880.0000 6.0     0.4     0.6
location          'West boundary'

```

```

time-function      'non-equidistant'
time-unit          'date'
interpolation      'linear'
parameter          'time' unit '[]'
parameter          'depth end A' unit '[m]'
parameter          'depth end B' unit '[m]'
19941001000000    5.0      5.5
19941003000000    6.0      6.5
location           'West boundary'
time-function      'equidistant'
reference-time     19941001
time-unit          'days'
time-step          2
interpolation      'linear'
parameter          'mass fraction Sediment fine end A' unit '[-]'
parameter          'mass fraction Sediment coarse end A' unit '[-]'
parameter          'mass fraction Sediment fine end B' unit '[-]'
parameter          'mass fraction Sediment coarse end B' unit '[-]'
0.3                0.7      0.35     0.75
0.4                0.6      0.45     0.65

```

B.9.8 Morphological factor file

File contents	Time-series for morphological factor		
Filetype	ASCII		
File format	Fix format for header information; free format for time-series data.		
Filename	<name.mft>		
Generated	manually offline		

Record description:

The data is given in two related blocks:

- ◊ A header block containing a number of compulsory and optional keywords accompanied by their values.
- ◊ A data block containing the time dependent data.

Description header block:

Text	Required	Value
table-name hline location	no no	arbitrary string should be blank if specified
time-function	no	{‘non-equidistant’} or ‘equidistant’
time-step	yes	time step only in case of time-function ‘equidistant’
reference-time	yes	yyyymmdd, yyymmdd hhmmss or ‘from model’
time-unit	no	‘years’, ‘decades’, ‘days’, ‘hours’, {‘minutes’}, ‘seconds’, ‘ddhhmmss’ or ‘date’
interpolation	yes	‘block’ (default value ‘linear’ should not be used)
extrapolation	no	‘periodic’, ‘constant’ or {‘none’}
parameter	yes	‘parameter name and location’ units ‘[ccc]’
records-in-table	no	number of times/lines in the data block

Remarks:

- ◊ Maximum record length is 512.
- ◊ The parameter name of the column should read ‘time’.
- ◊ The ‘parameter name and location’ string should read ‘MorFac’.
- ◊ Default parameter values are indicated in braces.
- ◊ Interpolation method should be ‘block’ for MorFac!
- ◊ Reference-time not required if time-unit equals ‘date’.
- ◊ Unit strings are currently not interpreted by Delft3D-FLOW.

Description data block:

Record	Record description
each record	Time in time-units after the reference-time and followed by as many values as parameters have been defined in the description block (all reals). In case of time-function ‘equidistant’, the first (time) column should be dropped. In case of time-unit ‘date’ the date and time should be specified as one string using the format: yyyymmddhhmmss.

Example:

The following file prescribes a morphological factor of 5 until Jan 2, 1994 0:00 and a factor of 10 thereafter.

```

time-function      'non-equidistant'
reference-time    19941001
time-unit         'minutes'
interpolation     'block'
extrapolation     'constant'
parameter          'time' unit '[min]'
parameter          'MorFac' unit '[-]'
      0.0000   5.0
      1440.0000 10.0
  
```

B.9.9 Initial bed composition file

The initial bed composition file allows you to specify the initial composition of multiple layers in the bed. If the bed is initially vertically homogeneous, you can use the `IniSedThick` (or older `SdBUni`) keywords in the `<*.sed>` file. If the bed is initially not completely homogeneous in the vertical direction, you should use this initial bed composition file. This `<*.inb>` file is keywords based.

Table B.11: Initial bed composition file keywords

Keyword	Record description
BedCompositionFileInfo	
FileCreatedBy	contains version number of FLOW-GUI
FileCreationDate	creation date and time of the <code><*.inb></code> file
FileVersion	version number, currently 01.00
Layer	
Type	Layer type (1 string) options: mass fraction: requires <code>Thick</code> and <code>Fractionx</code> keywords volume fraction: requires <code>Thick</code> and <code>Fractionx</code> keywords sediment mass: requires <code>SedBedx</code> keywords sediment thickness: requires <code>SedBedx</code> keywords
Thick	thickness of sediment layer [m] in case of either 'mass fraction' or 'volume fraction' type layer uniform value (1 real) or filename <code><*.thk></code> with non-uniform values at cell centres (1 string)
Fraction1, Fraction2, ...	mass or volume fraction in case of 'mass fraction' and 'volume fraction' type layer respectively uniform value (1 real) or filename <code><*.frc></code> with non-uniform values at cell centres (1 string)
SedBed1, SedBed2, ...	mass or thickness in case of 'sediment mass' and 'sediment thickness' type layer respectively uniform value (1 real) or filename <code><*.sdb></code> with non-uniform values at cell centres (1 string)
The Layer block can be repeated as often as needed.	

Example of an initial bed composition file:

The example file contains two layers. The first (top) layer has a spatially varying thickness as given by the QUICKIN file `<layer1.thk>` while the composition is uniform 30 % of sediment fraction 1 and 70 % of sediment fraction 2. The characteristics of the second layer are given as sediment mass [kg/m^2]: a constant 1600 kg m^{-2} of sediment fraction 2 and a spatially varying amount of sediment fraction 1 (as specified in the QUICKIN file `<layer2fraction1mass.sdb>`).

```
[BedCompositionFileInfo]
```

```

FileCreationDate = Thu Dec 08 2005, 14:47:50
FileVersion      = 01.00
[Layer]
Type            = mass fraction
Thick           = layer1.thk
Fraction1       = 0.3
Fraction2       = 0.7
[Layer]
Type            = sediment mass
SedBed1         = layer2fraction1mass.sdb
SedBed2         = 1600

```

B.10 Fluid mud (2-layer approach)

This feature (beta functionality) concerns the simulation of a fluid mud layer together with an overlying suspension layer using the 3D sediment transport, see the previous section. The modelling of fluid mud requires two separate MDF files, both with different additional keywords.

Suspension layer

Keywords	Value	Description
Filsed	#name.sed#	Sediment input file; defines sediment fractions and other input quantities
Flumud	#YES#	Indicates that 3D sediment simulation is run parallel to a fluid mud simulation

Restrictions:

- ◊ Only one sediment type can be specified (most logically “mud”)
- ◊ Morphological changes cannot be computed.



Details about the sediment input file can be found in Appendix B.9.1.

Fluid mud layer

Keywords	Value	Description
Filmud	#name.mud#	Mud input file; defines properties of fluid mud

File contents See the table below
 Filetype ASCII
 File format Free Format
 Generated Manually offline



Remark:

- ◊ The keywords are case sensitive.

Record description:

Record	Record description
1	CBED: sediment concentration of the bed [g/l]
2	CMUD: sediment concentration of the fluid mud layer [g/l]
3	FMUD: friction coefficient between consolidated bed and fluid mud layer [-]
4	FWAT: friction coefficient between fluid mud layer and suspension layer [-]
5	MERS: bulk erosion coefficient [kg/m ² /s]
6	RHOSUS: density of the suspension layer [kg/m ³]
7	RHOMUD: density of the mud layer [kg/m ³]
8	TAUBNG: Bingham yield stress [Pa]
9	TAUERS: critical shear stress for erosion [Pa]
10	TAUSET: critical shear stress for settling [Pa]
11	VDEW: dewatering velocity [m/s]

Example:

```

400.0      cbet     =
75.0       cmud     =
0.05      fmud     =
0.032     fwat     =
0.000     mers     =
1020      rhosus   =
1065      rhomud   =
0.2       taubng   =
0.5       tauers   =
0.1       tauset   =
0.000     vdew     =

```

**Restrictions:**

- ◊ The values of the parameters are not checked against their domains.
- ◊ The fluid mud simulation is carried out using the same schematisation as the suspension layer simulation is carried out.
- ◊ The fluid mud layer must be modelled as a 2DH problem.
- ◊ The modelling of a fluid mud layer requires a non-standard method of using the Delft3D software. This aspect is discussed in detail further on in this appendix.

Detailed description:

In some (consultancy) projects (one of) the question(s) to be answered is the generation and fate of fluid mud layers in estuaries. These questions may be related to optimising dredging strategies or environmental problems concerning the transport of contaminated mud. Fluid mud layers may be formed from deposition of sediment from the water column. Once formed, the fluid mud layer may be forced to flow by gravity or shear stresses induced by the overlying

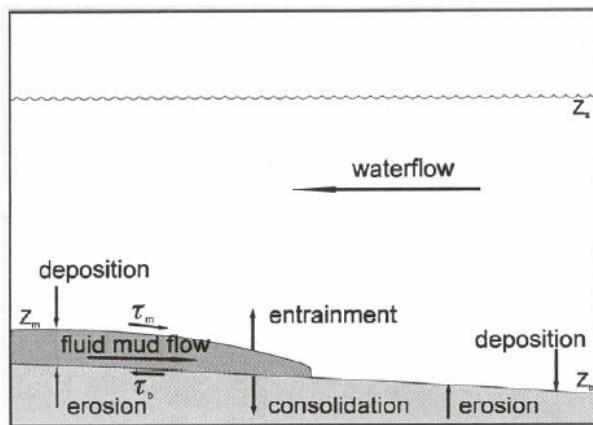


Figure B.12: Schematic representation of the governing processes between suspension layer and fluid mud layer [Winterwerp et al. \(1999\)](#).

water. If shear stresses are high enough, material from the mud layer may be entrained into the water column. When the flow in the mud layer is sufficiently high, the mud layer may erode the consolidated bed. Material thus eroded contributes to the fluid mud layer. When the fluid mud layer is at rest, the fluid mud layer is subjected to consolidation. The processes described above, are depicted in [Figure B.12](#).

B.10.1 Two layer system

The system of a suspension layer and a fluid mud layer is considered as a two layer system. A suspension layer is driven by tidal forces and possibly by wind shear stresses. Sediment may be transferred from the bottom to the fluid mud layer by erosion and the other way around by consolidation, and sediment may be transferred from the fluid mud layer to the suspension layer by means of entrainment and the other way around by deposition.

B.10.1.1 Suspension layer

The water motion in the fluid mud layer is assumed to be governed by tidal forces, wind shear stresses, bottom shear stresses and sediment concentration. Since the sediment concentration is assumed to have a significant effect the 3D sediment version of Delft3D-FLOW is used for the simulation of the suspension layer.

It is assumed that the velocities in the suspension layer are much larger than the velocities in the fluid mud layer. In addition it is assumed that the thickness of the fluid mud layer is negligible compared to the thickness of the water layer. Consequently, the fluid mud layer is therefore assumed not to influence the suspension layer hydrodynamically. For the present system, the bed roughness is also independent of the presence of the fluid mud layer. The fluid mud layer, however, acts as a sediment source or sink for the suspension layer.

B.10.1.2 Fluid mud layer

The fluid mud flow is assumed to be driven by the shear stresses acting in the interface of suspension layer and fluid mud. The non-Newtonian rheological properties of the fluid mud layer are accounted for by the formulation of the shear stresses in the interfaces of suspension and fluid mud layer, and of fluid mud layer and the consolidated bed, if the fluid mud layer is modelled as 2DH problem.

The shear stresses in the fluid mud - water interface depend on the velocity difference between the two layers using a material specific friction coefficient. The shear stresses in the bed - fluid mud layer depend on the speed of the fluid mud flow, again using a material specific friction coefficient and a Bingham yield stress. Sediment is transported from the bed to the fluid mud layer, to the suspension layer and *vice versa*. The density of the fluid mud layer is assumed to be constant, as is the density of the consolidated bed. In the present formulations, the thickness of the consolidated bed is assumed not to change with time. Furthermore, the mud bed is assumed to be an infinite source of mud.

B.10.1.3 Mathematical modelling of fluid mud layer

The mathematical model for the fluid mud layer is given by [Wang and Winterwerp \(1992\)](#). Herein only a concise description is provided. The mass balance reads:

$$\frac{\partial d_m}{\partial t} + \frac{\partial u_m d_m}{\partial x} + \frac{\partial v_m d_m}{\partial y} = \frac{1}{c_m} \frac{dm}{dt}, \quad (\text{B.14})$$

where t is time, x and y are spatial co-ordinates, d_m is the thickness of the fluid mud layer, u_m is the flow speed of the fluid mud layer in x -direction, v_m is the flow speed of the fluid mud layer in y -direction and c_m the (constant) sediment concentration in the mud layer. The right-hand term in Eq. B.14 represents the source term in the mass balance.

The equations of motion for both directions read:

$$\frac{\partial u_m}{\partial t} + u_m \frac{\partial u_m}{\partial x} + v_m \frac{\partial u_m}{\partial y} + g \frac{\rho_m - \rho}{\rho_m} \frac{\partial \eta_m}{\partial x} - \Omega v_m + \frac{1}{\rho_m d_m} (\tau_{bx} - \tau_{sx}) = -\frac{1}{\rho_m} \rho g \frac{\partial \eta}{\partial x} \quad (\text{B.15})$$

and

$$\frac{\partial v_m}{\partial t} + u_m \frac{\partial v_m}{\partial x} + v_m \frac{\partial v_m}{\partial y} + g \frac{\rho_m - \rho}{\rho_m} \frac{\partial \eta_m}{\partial y} - \Omega u_m + \frac{1}{\rho_m d_m} (\tau_{by} - \tau_{sy}) = -\frac{1}{\rho_m} \rho g \frac{\partial \eta}{\partial y} \quad (\text{B.16})$$

where ρ_m is the density of the fluid mud layer, ρ the density of the suspension layer, Ω the coefficient of the Coriolis acceleration, τ_{bx} the shear stress between the fluid mud layer and the underlying bed in x -direction, τ_{by} the shear stress between the fluid mud layer and the underlying bed in y -direction, τ_{sx} the shear stress between the suspension layer and the fluid mud layer in x -direction, τ_{sy} the shear stress between the suspension layer and the fluid mud layer in y -direction, η the elevation of the surface (of the suspension layer), η_m the elevation of the mud-suspension interface and g the acceleration due to gravity.

The shear stresses are given by:

$$\begin{pmatrix} \tau_{bx} \\ \tau_{by} \end{pmatrix} = \begin{pmatrix} u_m \\ v_m \end{pmatrix} \frac{\tau_m}{\sqrt{u_m^2 + v_m^2}} \quad (\text{B.17})$$

where

$$\tau_m = \tau_B + \frac{f_m \rho_m}{8} (u_m^2 + v_m^2), \quad (\text{B.18})$$

and

$$\begin{pmatrix} \tau_{sx} \\ \tau_{sy} \end{pmatrix} = \begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix} \frac{f_s \rho \sqrt{\Delta u^2 + \Delta v^2}}{8}. \quad (\text{B.19})$$

Herein is:

f_m	friction coefficient between the bed and the fluid mud layer,
f_s	friction coefficient between the suspension layer and the fluid mud layer,
τ_B	Bingham yield strength,
Δu	$u - u_m$,
Δv	$v - v_m$,
u	x -component of the flow velocity in the suspension layer ($k = k_{max}$)
v	y -component of the flow velocity in the suspension layer ($k = k_{max}$).

The sediment exchange rate between the suspension layer and the fluid mud layer includes:

- ◊ settling
- ◊ entrainment

and between the fluid mud layer and the (consolidated) bed:

- ◊ erosion
- ◊ consolidation, when the fluid mud layer is at rest ($u_m = v_m = 0$).

The source term in the mass balance equation B.14 is given by:

$$\frac{dm}{dt} = \text{Settling} - \text{Entrainment} + \text{Erosion} - \text{Dewatering} \quad (\text{B.20})$$

The settling term is given by:

$$\text{Settling} = H\left(\frac{\tau_{dm} - \tau_s}{\tau_{dm}}\right) w_s C \left(\frac{\tau_{dm} - \tau_s}{\tau_{dm}}\right) \quad (\text{B.21})$$

Herein $H()$ is the Heaviside function, and τ_{dm} is the critical shear stress for settling.

The entrainment term is given by [Winterwerp et al. \(1999\)](#)

$$\text{Entrainment} = \frac{2C_s \langle |\vec{u}_{*,m}|^2 - \frac{\tau_B}{\rho_m} \rangle |\vec{u} - \vec{u}_m| + C_\sigma \langle |\vec{u}_{*,s}|^2 - \frac{\tau_B}{\rho_m} \rangle |\vec{u}_{*,s}|}{\frac{gh\Delta\rho}{\rho} + C_s |\vec{u} - \vec{u}_m|^2} c_m \quad (\text{B.22})$$

in which the terms between the angular brackets are relevant only if their values are positive. Furthermore, $u_{*,s}^3 = u_*^3 + u_{*,m}^3$, u_* is shear velocity of the flow of the suspension layer, and $u_{*,m}^2 = f_s(u - u_m)^2$. The empirical coefficients are $C_s = 0.25$ and $C_\sigma = 0.42$.

The erosion term is given by:

$$\text{Erosion} = H\left(\frac{\tau_b - \tau_e}{\tau_e}\right) M_e \left(\frac{\tau_b - \tau_e}{\tau_e}\right) \quad (\text{B.23})$$

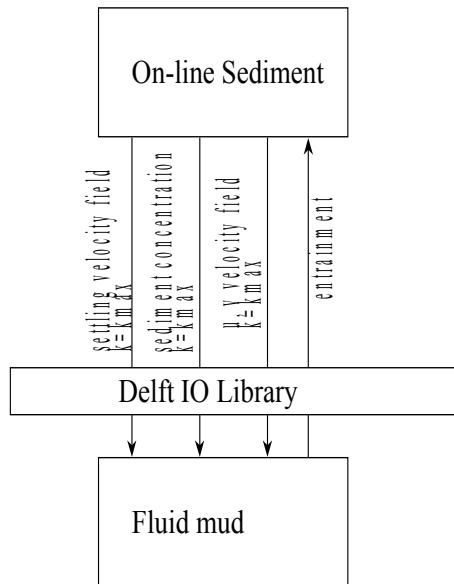


Figure B.13: A schematic representation of two Delft3D-FLOW modules running simultaneously simulating a fluid mud problem

where M_e is the bulk erosion coefficient and τ_e is the critical shear stress for erosion.

The dewatering term is given by:

$$\text{Dewatering} = V_0 c_0 \quad (\text{B.24})$$

where V_0 is the consolidation speed and c_0 is the sediment concentration in the fluid mud layer.

In (Wang and Winterwerp, 1992) it is shown that the governing equations for the fluid mud layer are quite similar to the governing equations for the water phase in the 2DH case, so that only minor modifications to the standard Delft3D-FLOW code are required to apply this code to fluid mud simulations.

B.10.2 Applying fluid mud

In practice, the standard code for Delft3D-FLOW with 3D sediment is modified such that the same code can be applied to both the suspension layer (possibly in 3D) and the fluid mud layer (only 2DH). A simulation of a fluid mud problem requires the execution of two Delft3D-FLOW modules, one for the suspension layer and one for the fluid mud layer. The data transfer between the two modules is handled by DelftIO library calls. These library calls also take care of a proper synchronisation of both Delft3D-FLOW modules. A schematic representation of a fluid mud simulation is shown in Figure B.13.

The procedure followed is that two MDF-files are created: one for the suspension layer and one for the fluid mud layer. The FLOW module determines from the MDF file whether the FLOW module is run for the suspension layer or the fluid mud layer. Both modules are executed simultaneously and communicate the following parameters:

- ◊ velocity field of the bottom layer of the suspension layer.

- Direction: suspension layer → fluid mud layer
- Purpose: calculating shear stresses between suspension layer and fluid mud layer
- ◊ sediment concentration and settling velocity in the bottom layer of the suspension phase
 - Direction: suspension layer → fluid mud layer
 - Purpose: calculating the settling flux from the suspension layer to the fluid mud layer
- ◊ entrainment flux
 - Direction: fluid mud layer → suspension layer suspension layer to the fluid mud layer

The parameters described above are communicated each half time step.

B.10.2.1 DelftIO library

The communication between two delftflow executables is handled by DelftIO library calls. The behaviour of these communication lines are to some extent user configurable. This can be done by editing the file <dioconfig.ini>, which follows the syntax of common Windows <.ini> files. A <dioconfig.ini> file may contain a Defaults section in which the global settings for all streams can be specified.

The option StreamType defines the type of the data stream. This can be “ASCII”: a human readable format (useful for debugging), “BINARY”: a fast binary stream, human unreadable format and platform dependent format, “NEFIS”: a fast binary stream, human unreadable and platform independent format.

The option OnLine may take the values True or False. If True, then the data streams are synchronised. This means that a data stream may contain only one block of data of a predefined size, such as a flow field. When putting data onto the stream, the program waits until the data are taken up by the receiving program. Similarly, when receiving data, the program only continues when all data are received. The offline mode (OnLine=False) may be useful for calibrating the fluid mud module for large modules. In this way, the suspension layer needs to be simulated only once. The draw back is that there is no interaction between suspension layer and fluid mud layer.

The option Active can be used to exclude a data stream to be used during a simulation (Active=False). This is useful if a fluid mud simulation is run offline. For both the 3D sediment FLOW module and the fluid mud module the incoming data streams should be neglected.

The global settings may be altered for particular data streams. If this is required the Defaults section contains a line NumberOfDatasets = x where x represents the number of data sets with different settings. For each data set the options above can be altered. A data stream is identified with the keyword Name=...

Below follows an example of the <dioconfig.ini> file.

```
[Defaults]
StreamType=ASCII
OnLine=True
NumberOfDatasets=2
[Dataset1]
Name=Dataset_A
Desc=description of dataset A
StreamType=Binary
[Dataset2]
Name=Dataset_B
```

```
Active=False
```

If no <dioconfig.ini> file exists, the default values are used:

```
StreamType=BINARY
OnLine=False
Active=True
```

B.10.2.2 Running a simulation in foreground

The procedure to run a fluid mud simulation in the foreground is straight forward: two flow models are executed directly after another by using the FLOW Selection menu.

B.10.2.3 Running a simulation in background

For a fluid mud simulation it is essential that two delftflow executables are run simultaneously. The batch file prepare options in the FLOW Selection window currently do not provide in launching executables simultaneously. However, the Selection menu generated scripts can easily be adapted to be applied to fluid mud simulations. If more than one fluid mud simulation is to be run, it is essential that the batch file is generated such that the delftflow.exe's that should be run simultaneously are added to the batch file one after another.

- ◊ Unix shell scripts: The unix shell scripts run delftflow.exe, and repeat this sequence for all models specified. In order to execute delftflow.exes simultaneously you need to add an "&" sign to the command line executing *only the first* delftflow program:

```
 ${D3D_HOME} / ${ARCH} / flow/bin/delftflow.exe >> flow-inp.scr 2>&1 &
```

- ◊ Windows batch files: The Windows batch files are similar to the Unix shell scripts except for the syntax. The command line executing the first delftflow.exe should be such that the delftflow.exe is executed by the start command

```
 start %D3D_HOME%\%ARCH%\flow\bin\delftflow.exe >> flow-mud.scr
```

B.10.2.4 Pitt falls

The present version of the program does not check for possible inconsistencies between the problem definition of the suspension layer and the fluid mud layer. You are considered to be responsible for preserving the consistency. Some mistakes are easily made and may result in incomprehensible error messages. The following checklist may be used to track down the problem in case.

- ◊ consistent grid definition;
- ◊ consistent depth schematisation;
- ◊ consistent time frame (including time spacing);
- ◊ consistent water density suspension phase and suspension density in fluid mud attribute file;
- ◊ only one delftflow simulation can be run with online visualisation checked on;
- ◊ the fluid mud problem must be modelled 2DH (i.e. Kmax=1);
- ◊ DelftIO communication stream <dio-*> files have to be removed before running a fluid mud simulation. These files may exist if a previous run did not terminate normally.

B.10.3 Post-processing

As a result of the similarity in solving the equations pertaining to the suspension layer and those pertaining to the fluid mud layer, the standard post-processing can be used to visualise the computational results of the fluid mud layer. It is remarked that the nomenclature of some

parameters may be somewhat confusing, for example *water level* should be read as *mud level*.

B.11 Z-model

B.11.1 Grid definition and construction

The vertical grid in the Z -model is based on horizontal layers with constant z co-ordinate intersecting the water column. The (maximum) layer thickness is defined as the distance between two consecutive surfaces and is independent of space and time.

Near the free surface and the bed the computational grids may be partially filled depending on the local depth and the free surface elevation. Furthermore, the free surface in the Z -grid-ordinate model is not restricted to the upper most grid cell. Consequently, the number of active grid cells may vary in space and time. The values k_{min} and k_{max} , indicating the index of the lower and upper most active grid cell, are not constant. Note that the Z -model indexes its layer number, is in contrast to the σ -grid, from bottom to surface; $k = k_{min}$ is the bottom layer, $k = k_{max}$ is the surface layer.

The vertical grid is constructed upon the specification of the following input parameters (see Figure B.14):

Keywords	Value	Description
Zmodel	#Y#	a (character) flag denoting that a Z -model is defined instead of a σ -model
Zbot	ZBOT	A real value representing the maximum depth in the model (relative to the reference level; unit: metres; positive upwards). It marks the lower boundary of the grid.
Ztop	ZTOP	A real value representing the ‘imaginary’ maximum water level in the model (relative to the reference level; unit: metres; positive upwards). This imaginary level is used only to determine the grid distribution. It does not mark the maximum surface level.

Record description:

- ◊ The distance $d(Ztop, Zbot)$ is used to determine the thickness of each layer with the help of the layer distribution specified by you (array THICK). The thickness of layer k is referred to as ΔZ_k .
- ◊ The thickness of the top layer is determined every half time step using the fixed layering and the newly computed surface level. The top of the top layer is set equal to the surface level. If the surface level exceeds Ztop by a (relatively) large distance, this can lead to a thick top layer, reducing the local accuracy. If this happens during a simulation, a warning is issued (to the <tri-diag> file) of the following form:

```
*** WARNING Maximum water level 0.146E-03 (m) exceeds ZTOP specified in input;
changing ZTOP to above this value is strongly advised
```

This top layer will always be sufficiently large to contain the computed water level, but the vertical resolution at the top level may be reduced when these warnings occur.

- ◊ Near the bed, the layer thickness is not uniform as it depends on the actual depth. Layers may also be situated (partly) under the bed level.

**Remarks:**

- ◊ Ztop must be greater than Zbot.
- ◊ Zbot must be at least equal (in absolute term) to the maximum depth value specified, otherwise an error message will be generated of the form:

```
*** ERROR Depth value -0.350E+01 (m) exceeds ZBOT specified in input;
change ZBOT to this value
```

- ◊ Ztop must be at least equal (in absolute term) to the initial water level specified, otherwise an error message will be generated.
- ◊ If a bottom level is specified above Ztop (dry points with depth set to e.g. -100.0 m) an error is given at the start of the simulation, with the message to change the concerned cells to permanently dry points. This message has the following form:

```
*** ERROR One or more depth values exceed ZTOP specified in input
Options:
```

```
1: Change ZTOP to above the highest value: 0.100E+02
2: Copy the following lines into the .dry file to make these points permanently dry:
3, 2, 3, 2
10, 39, 10, 39
End of data-block for dry points file.
```

- ◊ *The sequence of layer number is reversed in comparison to the σ -co-ordinate model*

Example - input parameters:

This example concerns a simulation where some layers become partly or completely dry. It is a 5-layered model with a non-uniform vertical grid. The reference level is at 0.0 m. The parameter values that have been specified in the input file are:

```
Zmodel= #Y#
Zbot   = -4.0
Ztop   =  1.0
Thick  = 12.0
      23.0
      35.0
      20.0
      10.0
```

For an arbitrary time instant where the maximum water level is approximately 1.8 m — exceeding Ztop — the resulting vertical grid is shown in [Figure B.14](#).

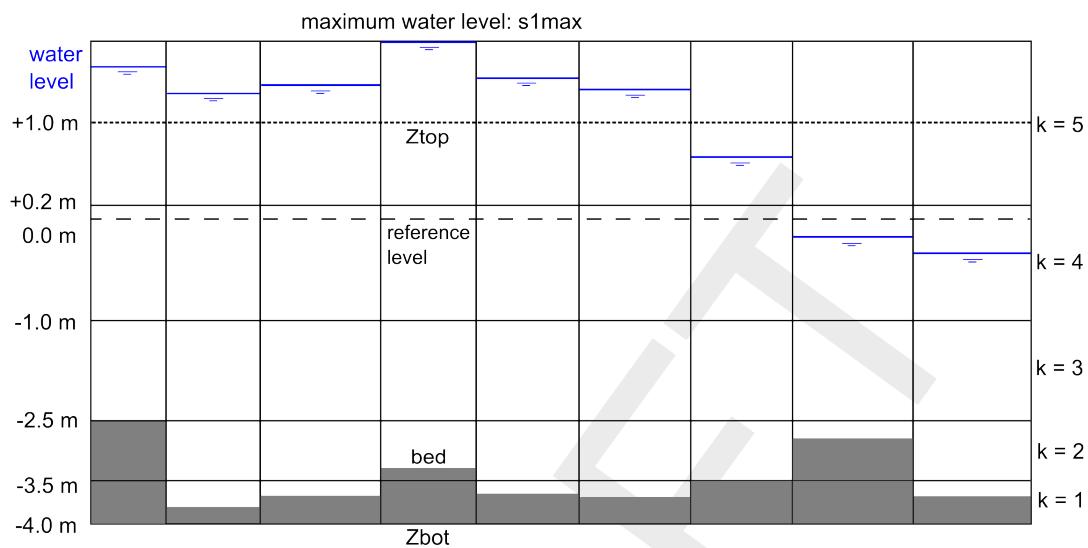
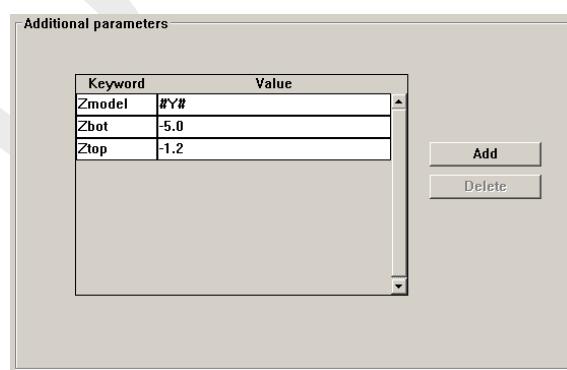
B.11.2 Defining the keywords in the FLOW-GUI

Thick corresponds to the layer array in the window Datagroup *Domain*, subgroup *Grid parameters*. Keywords Zmodel, Zbot and Ztop and their respective values can be added/edited via the Data Group *Additional parameters* (see [Figure B.15](#)):

B.11.3 Restrictions and limitations

The following functions are not allowed when used in combination with the Z -model:

- ◊ 3D Morphology (on line SED), incl. Roller model
- ◊ Coupled FLOW-WAQ simulations
- ◊ Internal Wave option (IWE)
- ◊ Wave effect (Wave induced flow)
- ◊ Fluid Mud
- ◊ Q2E 2D Turbulence closure model

**Figure B.14:** Vertical grid construction, Z-model**Figure B.15:** Inserting appropriate keywords to switch on the Z-grid-model

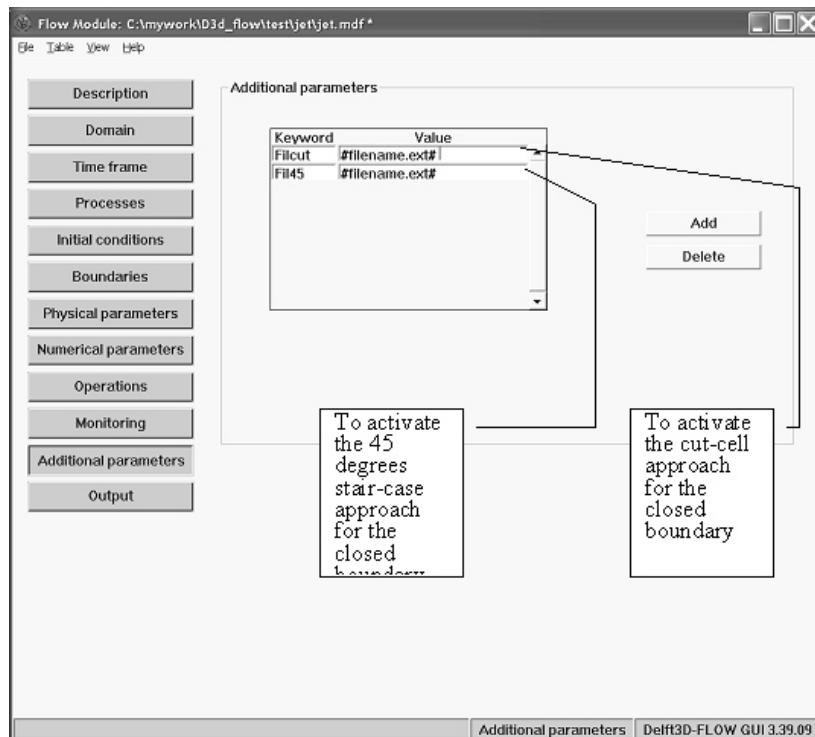


Figure B.16: Defining cut-cell and 45 degree closed boundaries

- ◊ Secondary Flow
- ◊ Waqua option - structure
- ◊ 2D Weir
- ◊ Momentum solver other than Multi Direction Upwind Explicit (default)
- ◊ Cyclic Transport solver - will be overridden Van Leer 2 method (default)
- ◊ DPUOPT other than MIN (minimum) which is the default or UPW (upwind)

The following (combinations) of functions have not been (fully) tested (WARNINGS will be issued, but computation will continue):

- ◊ Spherical co-ordinate
- ◊ Tide generating forces
- ◊ Droges
- ◊ Structures

B.11.3.1 Defining Cut-cells and 45 degrees closed boundaries

To activate the cut-cell approach or the removal of 45 degrees staircase effect along the closed boundaries the following case sensitive keywords may be added/edited in the MDF-file via the Data Group *Additional parameters* respectively as follows:

The format of a 45 degree closed boundary file is identical to that of the dry points file. Similar rules for defining these points are applicable.

The format of a Cut-cell closed boundary file is identical to that of the thin dams file. Similar rules for defining these points are applicable.

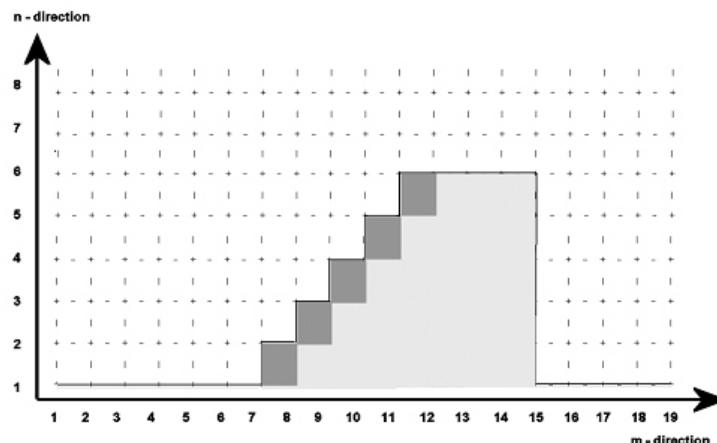


Figure B.17: 45 degrees staircase closed boundary

B.11.4 45 degrees staircase closed boundary points (Z -model only)

File contents	Location of (permanently) 45 degrees staircase closed boundary points with the purpose of removing the staircase effect (Z -model only)
Filetype	ASCII
File format	Free formatted.
Filename	<name.cut>
Generated	FLOW-GUI (treat it as dummy dry point) or manually offline.

Record description:

Record	Record description
each record	The grid indices of the begin and end point of a 45 degrees closed boundary point section (4 integers).

Restrictions:

- ◊ The angle of a line of dry points and the horizontal numerical grid axis can be an integer multiple of 45 degrees.
- ◊ Points forming 45 degrees closed boundary must be defined on active internal points.
- ◊ The input items are separated by one or more blanks.
- ◊ 45 degrees staircase point may not be Cut-cell as well



Example:

Five sets of 45 degrees closed boundary points in a model area of $19 * 8$ grid points, see [Figure B.17](#).

```

8   1   8   1
9   2   9   2
10  3  10  3
11  4  11  4
12  5  12  5
  
```

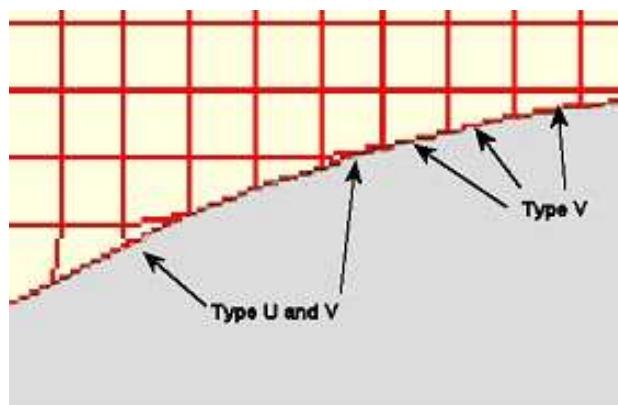


Figure B.18: Cut-cell closed boundary (not related to the data specified in the example above)

B.11.5 Cut-cell closed boundary points (Z -model only)

File contents	Location and type of cut-cell closed boundary points (Z -model only)
Filetype	ASCII
File format	Free formatted.
Filename	<name.cut>
Generated	manually offline.

Record description:

Record	Record description
each record	The grid indices of the begin and end point of a line of thin dams (4 integers). A character indicating the type of thin dams (U or V).



Restrictions:

- ◊ Cut-cell closed boundary must be an active internal point but adjacent to closed boundary.
- ◊ Cut-cell closed boundary point must have an area that is > 0 .
- ◊ Cut-cell point in a corner is not allowed.
- ◊ Cut-cell point may not be 45 degrees staircase cell as well.

Example:

Twelve points in a model are marked as cut-cell points, see [Figure B.18](#).

```

4    15    4    15  V
3    15    3    15  U
3    13    3    13  V
2    13    2    13  U
1     9    1     9  U
2     9    2     9  V
11   7    11    7  U
11   6    11    6  V
12   11   12   11  U
12   10   12   10  V
13   13   13   13  U
13   12   13   12  V

```

B.12 Non-hydrostatic solver

B.12.1 The use of hydrostatic and non-hydrostatic models

The application of three-dimensional numerical models for understanding and assessing free surface flows has become widely used in the past decades. In most of these cases the flows are of boundary layer type, which means that the vertical acceleration component is relatively small. Under these circumstances, the pressure distribution in the vertical may be assumed to be hydrostatic, and hence the hydrodynamic pressure may be neglected.

Within the Delft3D-FLOW model, this hydrostatic pressure assumption has been adopted, which implies that the numerical model is based on the shallow-water equation. The approximation of hydrostatic pressure or, equivalently, shallow water simplifies the coupled set of three-dimensional equations to be solved and therefore allows the development of a computationally efficient three-dimensional numerical hydrodynamic model.

Delft3D-FLOW has been used successfully in many practical test cases in describing horizontal velocity profiles in vertical direction or stratified currents in predominantly horizontal flows. However, there are situations where the vertical acceleration, and thereby the non-hydrostatic pressure component, can not be neglected. This, for example, is the case when flows over abruptly changing bottom topography, orbital movements in short wave motions, or intensive vertical circulations are considered. Generally, the application of hydrostatic approximation is disputable, when the vertical scales of the flow (i.e. velocities and distances) are of the same order of magnitude as the horizontal scales of the flow. When the requirements for shallow-water flow are no longer met, the hydrodynamic component of the pressure has to be resolved in order to get physically realistic flow patterns.

In order to capture non-hydrostatic flow phenomena, the hydrostatic version of the Cartesian grid (Z -model), which is part of the Delft3D modelling suite, is extended with a non-hydrostatic module. The numerical method employed in this model is presented in the following.

B.12.2 Governing equations

Before discussing the numerical method used for computing the non-hydrostatic pressure, the underlying equations are given.

For small density differences ($\Delta\rho/\rho_0 \ll 1$, where ρ_0 is a reference density), the Reynolds-averaged Navier-Stokes equation may be written as:

$$\frac{\partial u_i}{\partial t} + u_i \frac{\partial u_j}{\partial x_i} + \frac{1}{\rho_0} \frac{\partial p}{\partial x_i} - \frac{1}{\rho_0} \frac{\partial \tau_{ij}}{\partial x_j} + \varepsilon_{ijk} 2\Omega_j u_k = \frac{\rho}{\rho_0} g \delta_{i3} \quad (\text{B.25})$$

where δ_{ij} is the Kronecker delta, ε_{ijk} is the permutation symbol, Ω_j is the planetary vorticity and τ_{ij} are the turbulent stresses. The pressure p can be split into a hydrostatic ($\rho g z$) and hydrodynamic (q) part:

$$p = p_{atm} + g \int_z^\zeta \rho dz' + q \quad (\text{B.26})$$

Using this expression, the momentum equations in the x , y , and z -direction read respectively:

$$\frac{\partial u}{\partial t} + \frac{1}{\rho_0} \frac{\partial}{\partial x} \int_z^\zeta \rho g dz' + \frac{1}{\rho_0} \frac{\partial q}{\partial x} = RHS_x \quad (\text{B.27})$$

$$\frac{\partial v}{\partial t} + \frac{1}{\rho_0} \frac{\partial}{\partial y} \int_z^\zeta \rho g dz' + \frac{1}{\rho_0} \frac{\partial q}{\partial y} = RHS_y \quad (\text{B.28})$$

$$\frac{\partial w}{\partial t} + \frac{1}{\rho_0} \frac{\partial q}{\partial z} = RHS_z \quad (\text{B.29})$$

where the right-hand-side, RHS_i , contains the acceleration due to advection, turbulent stresses and Coriolis. The shallow-water equations are obtained by reducing the vertical momentum equation to the hydrostatic pressure assumption, which implies $q = 0$.

The continuity equation is given by:

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (\text{B.30})$$

By integrating the continuity equation over the depth, and applying the kinematical boundary condition

$$\frac{D\zeta}{Dt} \equiv \frac{\partial \zeta}{\partial t} + u \frac{\partial \zeta}{\partial x} + v \frac{\partial \zeta}{\partial y} = w \quad (\text{B.31})$$

this equation may be written as:

$$\frac{\partial \zeta}{\partial t} + \frac{\partial}{\partial x} \int_{-d}^\zeta u dz + \frac{\partial}{\partial y} \int_{-d}^\zeta v dz = 0 \quad (\text{B.32})$$

B.12.3 A pressure correction technique for computing the non-hydrostatic pressure

The hydrodynamic pressure is computed by using a pressure correction technique that is similar to the method presented by [Casulli \(1999\)](#) and [Busnelli \(2001\)](#). The pressure correction method is employed in combination with an ADI-technique which differs slightly from the Delft3D σ -model ([Bijvelds, 2001](#)).

The provisional horizontal velocities, u^{n+1*} , v^{n+1*} and w^{n+1*} are computed from the ADI-steps:

First step ADI

$$\frac{u^{n+1/2} - u^n}{\Delta t} = g \frac{\zeta_{i+1}^{n+1/2} - \zeta_i^{n+1/2}}{\Delta x} + \frac{1}{\rho_0} \frac{q_{i+1}^n - q_i^n}{\Delta x} + RHS_x \quad (\text{B.33})$$

$$v^{n+1/2} = v^n \quad (\text{B.34})$$

$$\begin{aligned} \frac{\zeta^{n+1/2} - \zeta^n}{\Delta t} + \frac{\sum_{k=1}^K \left(\Delta z_k^n u_k^{n+1/2} \right)_i - \sum_{k=1}^K \left(\Delta z_k^n u_k^{n+1/2} \right)_{i-1}}{\Delta x} + \\ \frac{\sum_{k=1}^K \left(\Delta z_k^n v_k^{n+1/2} \right)_j - \sum_{k=1}^K \left(\Delta z_k^n v_k^{n+1/2} \right)_{j-1}}{\Delta y} = 0 \end{aligned} \quad (\text{B.35})$$

Second step ADI

$$u^{n+1*} = u^{n+1/2} \quad (\text{B.36})$$

$$\frac{v^{n+1*} - v^{n+1/2}}{\Delta t} = g \frac{\zeta_{j+1}^{n+1*} - \zeta_j^{n+1*}}{\Delta y} + \frac{1}{\rho_0} \frac{q_{j+1}^n - q_j^n}{\Delta y} + RHS_y \quad (\text{B.37})$$

$$\begin{aligned} \frac{\zeta^{n+1*} - \zeta^{n+1/2}}{\Delta t} &+ \frac{\sum_{k=1}^K (\Delta z_k^n u_k^{n+1*})_i - \sum_{k=1}^K (\Delta z_k^n u_k^{n+1*})_{i-1}}{\Delta x} + \\ \frac{\sum_{k=1}^K (\Delta z_k^n v_k^{n+1*})_j - \sum_{k=1}^K (\Delta z_k^n v_k^{n+1*})_{j-1}}{\Delta y} &= 0 \end{aligned} \quad (\text{B.38})$$

The provisional vertical velocity is computed from

$$\frac{w^{n+1*} - w^n}{\Delta t} = \frac{1}{\rho_0} \frac{q_{k+1}^n - q_k^n}{\Delta z} + RHS_z \quad (\text{B.39})$$

The hydrodynamic pressure q is located in the cell-centre of the staggered C-grid. It is noted that in the implementation of the above equations the acceleration due to Coriolis is omitted in the vertical momentum equation.

The barotropic pressure gradient is computed implicitly in order to avoid time step restrictions due to stability conditions. This yields a tri-diagonal set of equations which can be solved efficiently by using a double sweep technique. For computational efficiency, the remainder of the terms are treated explicitly. Further details on the spatial discretisation of the differential equations can be found in (Bijvelds, 2001). Recently it was found out that the discretisation of the advection terms in the Z -model was not robust enough in combination with the pressure correction method. The discretisation was changed to first order upwind instead of the multi-directional upwind method of the original implementation (Bijvelds, 2001).

It is noted that at this point of the computation the velocity field is no longer free of divergence, hence $\nabla \cdot \vec{u} \neq 0$. By applying a pressure correction in the next step of the computation, the requirement $\nabla \cdot \vec{u} = 0$ is met at all computational cells.

Hydrodynamic pressure correction

The momentum equations in the pressure correction step read:

$$\frac{u^{n+1} - u^{n+1*}}{\Delta t} = \frac{1}{\rho_0} \frac{\tilde{q}_{i+1}^{n+1} - \tilde{q}_i^{n+1}}{\Delta x} \quad (\text{B.40})$$

$$\frac{v^{n+1} - v^{n+1*}}{\Delta t} = \frac{1}{\rho_0} \frac{\tilde{q}_{j+1}^{n+1} - \tilde{q}_j^{n+1}}{\Delta y} \quad (\text{B.41})$$

$$\frac{w^{n+1} - w^{n+1*}}{\Delta t} = \frac{1}{\rho_0} \frac{\tilde{q}_{k+1}^{n+1} - \tilde{q}_k^{n+1}}{\Delta z} \quad (\text{B.42})$$

Where $\tilde{q} = q^{n+1} - q^n$

It may be verified that the pressure correction applied in the above discretisation is a consistent approximation of the underlying differential equation, by substituting the following expression in the momentum equation in x -direction:

$$\frac{\partial u}{\partial t} - \frac{1}{\rho_0} \frac{\partial q}{\partial x} \approx \frac{u^{n+1} - u^* + u^* - u^n}{\Delta t} - \frac{1}{\rho_0} \left[\frac{q_{i+1}^{n+1} - q_i^{n+1}}{\Delta x} + \frac{q_{i+1}^n - q_i^n}{\Delta x} - \frac{q_{i+1}^n - q_i^n}{\Delta x} \right] \quad (\text{B.43})$$

For the sake of convenience the horizontal grid spacing, Δx and Δy , are assumed to be constant. The discretised continuity equation *below* the free surface (i.e. $k = kfsmin, \dots, kfsmax - 1$) reads

$$\frac{u_{i+1/2,j,k}^{n+1} - u_{i-1/2,j,k}^{n+1}}{\Delta x} + \frac{v_{i,j+1/2,k}^{n+1} - v_{i,j-1/2,k}^{n+1}}{\Delta y} + \frac{w_{i,j,k+1/2}^{n+1} - w_{i,j,k-1/2}^{n+1}}{\Delta z} = 0 \quad (\text{B.44})$$

At the free surface, the finite difference approximation of Eq. B.32 is

$$\frac{\zeta_{i,j}^{n+1} - \zeta_{i,j}^n}{\Delta t} + \frac{\sum_{k=1}^K (\Delta z^n \bar{u}^{n+1})_{i+1/2,j,k} - \sum_{k=1}^K (\Delta z^n \bar{u}^{n+1})_{i-1/2,j,k}}{\Delta x} + \frac{\sum_{k=1}^K (\Delta z^n \bar{v}^{n+1})_{i,j+1/2,k} - \sum_{k=1}^K (\Delta z^n \bar{v}^{n+1})_{i,j-1/2,k}}{\Delta y} = 0 \quad (\text{B.45})$$

Using Eq. B.44 this may be written as:

$$\frac{\zeta_{i,j}^{n+1} - \zeta_{i,j}^n}{\Delta t} + \frac{(\Delta z^n \bar{u}^{n+1})_{i+1/2,j,k} - (\Delta z^n \bar{u}^{n+1})_{i-1/2,j,k}}{\Delta x} + \frac{(\Delta z^n \bar{v}^{n+1})_{i,j+1/2,k} - (\Delta z^n \bar{v}^{n+1})_{i,j-1/2,k}}{\Delta y} + w_{K-1}^{n+1} = 0 \quad (\text{B.46})$$

Note that for consistency with the approximation of the continuity equation in the ADI-steps, the vertical grid spacing Δz should be taken at the time level n .

By inserting the momentum Equations B.40 to B.42 in the continuity equation for the inner field ($k = kfsmin, \dots, kfsmax - 1$), we arrive at a system of equations where the pressure correction q^* is the only unknown:

$$\begin{aligned} & \frac{1}{\rho_0} \left[\frac{(q_{i+1,j,k}^{n+1} - q_{i,j,k}^{n+1}) \Delta_{i+1/2,j,k}^n - (q_{i,j,k}^{n+1} - q_{i-1,j,k}^{n+1}) \Delta_{i-1/2,j,k}^n}{\Delta x^2} \right] + \\ & \frac{1}{\rho_0} \left[\frac{(q_{i,j+1,k}^{n+1} - q_{i,j,k}^{n+1}) \Delta_{i,j+1/2,k}^n - (q_{i,j,k}^{n+1} - q_{i,j-1,k}^{n+1}) \Delta_{i,j-1/2,k}^n}{\Delta y^2} \right] + \\ & \frac{1}{\rho_0} \left[\frac{(q_{i,j,k+1}^{n+1} - q_{i,j,k}^{n+1}) - (q_{i,j,k}^{n+1} - q_{i,j,k-1}^{n+1})}{\Delta z_{i,j,k+1/2}^n} \right] = \frac{u_{i+1/2,j,k}^{n+1*} \Delta z_{i+1/2,j,k}^n - u_{i-1/2,j,k}^{n+1*} \Delta z_{i-1/2,j,k}^n}{\Delta x} + \\ & \frac{v_{i,j+1/2,k}^{n+1*} \Delta z_{i,j+1/2,k}^n - v_{i,j-1/2,k}^{n+1*} \Delta z_{i,j-1/2,k}^n}{\Delta y} + w_{i,j,k+1/2}^{n+1*} - w_{i,j,k-1/2}^{n+1*} \end{aligned} \quad (\text{B.47})$$

At the top layer K , both q^* and ζ^{n+1} need to be determined:

$$\begin{aligned} \frac{\zeta_{i,j}^{n+1} - \zeta_{i,j}^n}{\Delta t} + \frac{1}{\rho_0} \left[\frac{(q_{i+1,j,K}^{n+1} - q_{i,j,K}^{n+1}) \Delta_{i+1/2,j,K}^n - (q_{i,j,K}^{n+1} - q_{i-1,j,K}^{n+1}) \Delta_{i-1/2,j,K}^n}{\Delta x^2} \right] + \\ \frac{1}{\rho_0} \left[\frac{(q_{i,j+1,K}^{n+1} - q_{i,j,K}^{n+1}) \Delta_{i,j+1/2,K}^n - (q_{i,j,K}^{n+1} - q_{i,j-1,K}^{n+1}) \Delta_{i,j-1/2,K}^n}{\Delta y^2} \right] + \\ \frac{1}{\rho_0} \left[\frac{(q_{i,j,K+1}^{n+1} - q_{i,j,K}^{n+1})}{\Delta z_{i,j,K+1/2}^n} - \frac{(q_{i,j,K}^{n+1} - q_{i,j,K-1}^{n+1})}{\Delta z_{i,j,K-1/2}^n} \right] = \frac{u_{i+1/2,j,K}^{n+1*} \Delta z_{i+1/2,j,K}^n - u_{i-1/2,j,K}^{n+1*} \Delta z_{i-1/2,j,K}^n}{\Delta x} + \\ \frac{v_{i,j+1/2,K}^{n+1*} \Delta z_{i,j+1/2,K}^n - u_{i,j-1/2,K}^{n+1*} \Delta z_{i,j-1/2,K}^n}{\Delta y} - w_{i,j,K-1/2}^{n+1*} \quad (\text{B.48}) \end{aligned}$$

In order to close the system of equations, the transient $(\zeta^{n+1} - \zeta^n)/\Delta t$ needs to be approximated. By assuming a excursion of the water level $\zeta^{n+1} - \zeta^{n+1*}$ as a result of the pressure correction step, the total pressure in the top cell may be written as

$$q_{i,j,K}^{n+1} = \rho g (\zeta_{i,j}^{n+1} - \zeta_{i,j}^{n+1*}) + q_{i,j,K}^{n+1*} \quad (\text{B.49})$$

Hence, the hydrodynamic pressure at the top cell now contains a hydrostatic pressure component equal to $\rho g (\zeta^{n+1} - \zeta^{n+1*})$. Assuming the pressure to be hydrostatic in the upper most grid cell, which implies $q_K^{n+1} = 0$, the pressure correction at the old geometry (based on ζ^{n+1*}), may be written as:

$$q_{i,j,K}^{n+1} = \rho g (\zeta_{i,j}^{n+1} - \zeta_{i,j}^{n+1*}) + q_{i,j,K}^{n+1*} \quad (\text{B.50})$$

Using this equation ζ can be eliminated from Eq. B.48 and the pressure correction is the only unknown variable. Substituting Equation B.50 in Equation B.48 yields a symmetric and positive definite system of equations, which is solved by using the standard conjugate gradients (CG) method.

The computed pressure correction field $q_{i,j,k}^{n+1*}$ contains a hydrostatic pressure component (proportional to $\zeta^{n+1} - \zeta^{n+1*}$) which originates from the change of water level. The barotropic pressure gradient hereby introduced is accounted for in the following ADI step. Therefore the total pressure field needs to be corrected for by a constant value equal to $\zeta^{n+1} - \zeta^{n+1*}$, according to Equation B.50.

For the upper most grid cells this yields

$$q_{i,j,K}^{n+1} = 0 \quad (\text{B.51})$$

which leads to the hydrostatic pressure assumption in the top layer. The new free surface elevation can readily be determined from Eq. B.50. The free surface boundary condition is shown graphically in Figure B.19.

B.12.4 Boundary conditions

In order to get a closed system of equations, boundary conditions for the pressure correction and the free surface elevation need to be imposed.

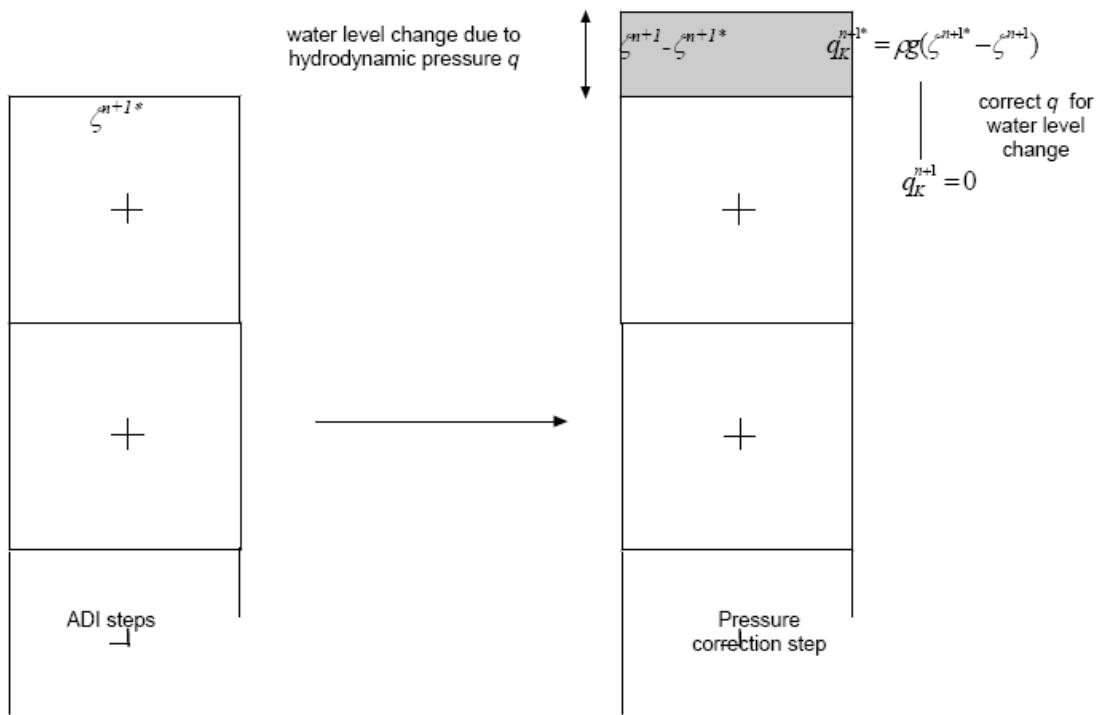


Figure B.19: Schematic representation of the free surface boundary condition for the pressure correction.

Near closed walls and at the boundaries of the non-hydrostatic area we assume homogeneous Neumann boundary conditions for the pressure correction:

$$\frac{\partial q^{n+1*}}{\partial n} = 0 \quad (B.52)$$

where n is the direction normal to the closed boundary. Physically this means there is no flow through closed boundaries due to hydrodynamic pressure gradients. At open boundaries where a water level is imposed the hydrodynamic pressure (correction) is assumed to be zero:

$$q^{n+1*} = 0 \quad (B.53)$$

B.12.5 Conjugate gradient method (CG)

In the non-hydrostatic model, the pressure correction is solved by means of the conjugate gradient method (CG). The CG method is an effective method for symmetric positive definite systems. It is the oldest and best known of the non-stationary methods. The method proceeds by generating vector sequences of iterates (i.e. successive approximations to the solution), residuals corresponding to the iterates, and search directions used in updating the iterates and residuals. Although the length of these sequences can become large, only a small number of vectors need to be kept in memory. At each iteration stage two inner products are performed in order to compute update scalars that are defined to make the sequences satisfy certain orthogonality conditions. On a symmetric positive definite linear system these conditions imply that the distance to the true solution is minimized in some norm.

The convergence rate of the CG method (and iterative methods in general) depends on the spectral properties of the coefficient matrix. Therefore one may attempt to transform the linear system into one that is equivalent in the sense that it has the same solution, but that has more favourable spectral properties. A pre-conditioner is a matrix that effects such a transformation.

In the present implementation, local scaling of the system of equations is used to improve the matrix properties, being one of the simplest pre-conditioners.

For further details on the CG method, the reader is referred to [Golub and Van Loan \(1989\)](#).

B.12.6 Practical aspects of using the non-hydrostatic flow module

The non-hydrostatic module provides you with a tool that is capable of simulating small-scale flow phenomena that cannot be captured with a ordinary hydrostatic model. In order to model these non-hydrostatic phenomena properly, some requirements regarding the model set-up and the model parameters that are used need to be met. These requirements stem from the physical nature of the flow, and are partly due to the fact that the original Delft3D-FLOW model was developed for stably-stratified shallow-water flow. In the following some practical aspects on the use of the non-hydrostatic model will be discussed.

B.12.6.1 Switches in MDF-file

The non-hydrostatic model can be switched on by you by including the key-word `Nonhyd = #YES#` in the MDF-file. It is noted that the non-hydrostatic option is available within the Z -model only. Therefore, the keywords of the Z -model need to be set, `Zbot`, `Ztop` and the keyword `Zmodel` must be also set to “`#YES#`” to activate the non-hydrostatic module (see [subsection B.11.1](#)).

Compared to hydrostatic simulations, including the non-hydrostatic pressure in the simulations increases the computational effort typically by a factor 4–10, depending on the type of problem and on the stopping criterion of the CG solver. In order to restrict the computational effort, the computation of the non-hydrostatic pressure can be limited to a part of the computational domain. In many practical applications non-hydrostatic phenomena are only locally of importance. As a consequence, the non-hydrostatic pressure may be neglected in the largest part of the computational domain. The neglect of the non-hydrostatic pressure may either ensue from the local physical nature of the flow, or may be due to the fact that the specific area lies outside the region of interest. Insight in the physics is therefore required when using this model feature.

To limit the region where the non-hydrostatic pressure is computed, the key-word `Nharea` must be included in the MDF-file, followed by two sets of indices that define a square region (in terms of grid administration). For example the statement

```
Nharea =      128      1      176      26
```

creates the following area (with indices (128,1)–(176,26)) within the curvilinear grid in which the non-hydrostatic pressure is taken into account, see [Figure B.20](#).

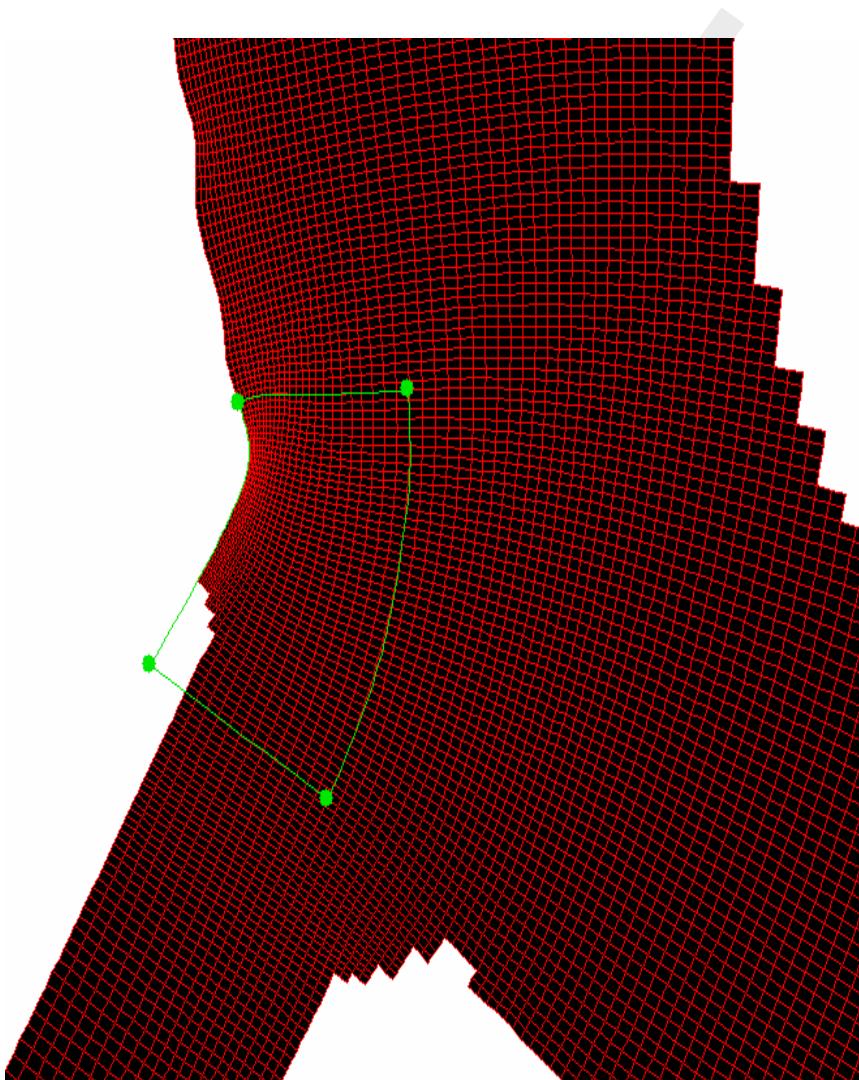


Figure B.20: Area where a non-hydrostatic pressure is taken into account

B.12.6.2 Grid spacing

The grids used for modelling shallow water flow are usually characterised by the fact that the horizontal grid spacing is much larger than the vertical grid spacing. Hence, the shallowness of the flow, characterised by $L_{hor} \gg L_{vert}$, is reflected by the grid's aspect ratio $\Delta x, \Delta y \gg \Delta z$. For non-hydrostatic flows, the typical length scales of the flow or of the same order of magnitude in both the horizontal and vertical direction. This means that the horizontal grid spacing has to be reduced significantly if non-hydrostatic flow phenomena need to be resolved.

If, for example, we consider the flow computed near a vertical wall in a hydrostatic and non-hydrostatic model, differences are only significant if we meet the above mentioned requirement, despite the fact that we apply an appropriate model for the physical phenomenon we are studying. In a hydrostatic model the vertical flow will typically be concentrated in the first grid cell from the wall, which is due to the fact that the vertical flow is determined by continuity only. (the wall is not “felt” in the model because there is no pressure built-up in the vertical direction). This means that the vertical velocities that are computed in the model are directly related to the horizontal grid spacing since $w = Q/(\Delta x \Delta y)$. In a low-resolution non-hydrostatic model, these low velocities lead to an under estimation of the non-hydrostatic pressure, and hence, the flow computed resembles the hydrostatic flow.

B.12.6.3 Vertical mixing

The hydrostatic version of Delft3D-FLOW is developed for stably stratified shallow-water flow. The presence of unstable stratification gives rise to strong vertical circulation, which, by definition, cannot be captured by the hydrostatic model, and results in flow patterns that are physically unrealistic. Since the model is not suited to model these types of flow, measures can be taken to suppress the presence of unstable stratification.

Among these measures, the so-called Forester filter may have a large impact on the results that are obtained with the non-hydrostatic Z -model. This filter, that may be switched on by you in the MDF-file, smears out the computed scalar variable (e.g. salinity, temperature), in such a way that a stably stratified monotonous solution is obtained for this quantity.

This filter is available in both the horizontal (`Forfuv`) and vertical direction (`Forfww`). When modelling stratified flows with the non-hydrostatic model in which unstable stratification may be expected, the filter in the vertical direction must be switched off (value field “no”). It is noted that wiggles which are due to numerical treatment of the differential equations, may arise when the Forester filter is not activated.

B.12.6.4 Convergence criterion CG solver

The system of equations is solved by using the CG algorithm. The number of iterations performed by the CG algorithm depends on the stopping criterion and on the type of norm that is used for the determination of the residual.

The value for the stopping criterion can be set by you by specifying a value for the keyword `Nheps` in the MDF-file. The default value for this parameter is set equal to 10^{-2} . If the criterion is set too small, the number of iterations may exceed the maximum number of iterations in CG, which may be defined by you by means of the key word `Nhiter`. The default value for this parameter is set equal to 50.

In that case a warning will be written to the diagnostic file, together with the maximum of the residual at the end of the last CG iteration. The way in which the residual of the CG-algorithm

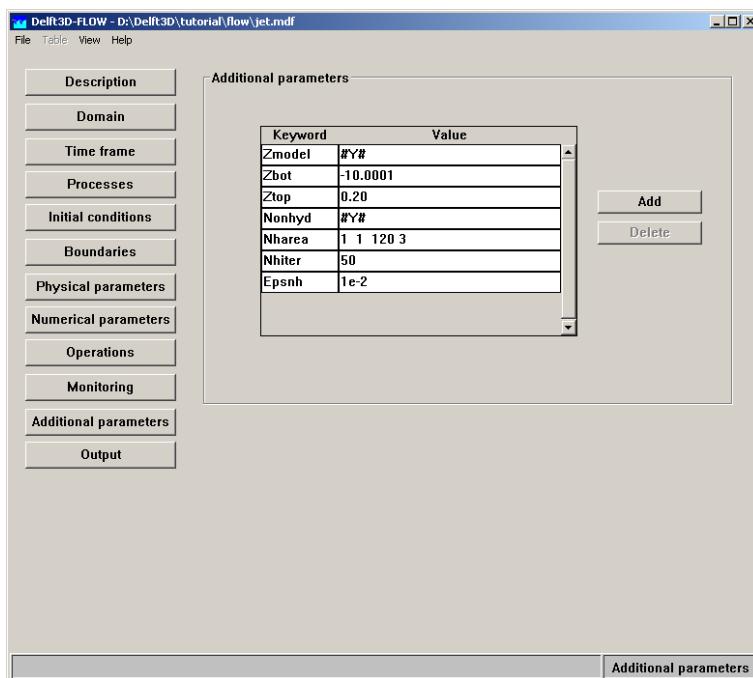


Figure B.21: Defining the Non-hydrostatic solver using the Z -model in the FLOW-GUI

is determined can be specified by you as well. This residual determines the measure of divergence of the computed velocity field. The default value for this parameter is the L_∞ -norm. For non-hydrostatic phenomena that have a strong local character it is advised to use the L_∞ -norm. In other cases the L_2 -norm will be sufficiently accurate. The L_2 -norm is used when the keyword L2norm in the MDF-file is set equal to #YES#.

B.12.6.5 Defining the input (keywords) for the non-hydrostatic pressure approach

To run the Z -model in the non-hydrostatic mode the following case sensitive keywords need to be added in the MDF-file via the window *Additional parameters*, for Z -model keywords see section B.11:

Keyword	Value	Description	Default
Nonhyd	#YES#	Activate non-hydrostatic module	–
Nharea	$1 \leq M_1 < M_2 \leq M_{\max}$ $1 \leq N_1 < N_2 \leq N_{\max}$	Define non-hydrostatic area	–
Nhiter	$5 \leq N \leq 500$	Maximum number of iterations	50
Epsnh	$0.0 < \varepsilon < 0.1$	Stop criterion CG-method	10^{-2}
L2norm	#Y# yields L_2 -norm #N# yields L_∞ -norm	Type of norm used for determination residual	#Y#

B.13 User defined functions

Keyword	Value	Description	Default
Nprcus	{1 to 4}	Number of User Defined Functions used	0

To use one or more User Defined Functions (UDF) you *must* specify the number of UDFs with the keyword Nprcus and its value. Next you must specify for each UDF a UDF-specific number of keywords and values; this input data is defined with the UDF sections. For each UDF you must define the keyword Prcusr which defines the UDF to be used.

The currently available User Defined Functions are:

Functionality	Description
bc turbulence model	Boundary conditions for the turbulence model
diagnostic mode	Specify to use the diagnostic mode
particle wind factor	An additional wind speed working on particles

Finally you must specify the keyword Nprinp; its value consists of 4 integers which defines:

- ◊ the number of files to be used (first integer)
- ◊ the number of reals to be used (second integer)
- ◊ the number of integers to be used (third integer)
- ◊ the number of string variables to be used (fourth integer)

B.13.1 Boundary conditions for turbulence models

Keyword	Value	Description	Default
Prcusr	#bc turbulence model #	Boundary conditions for the k - L and k - ε turbulence models	none
Nprinp	1 0 0 0	One filename will be specified No reals No integers No strings	1 0 0 0
Filusr	#name.ext#	Filename with boundary conditions	none

In most cases the generation and dissipation turbulence is dominated by the processes inside the model area. The boundary conditions for both the turbulent kinetic energy and the turbulent energy dissipation rate are set to zero by default. However, in small area models the boundary conditions might be important. You can specify a vertical profile but otherwise uniform value for each boundary section.

File contents	The steady boundary conditions for the turbulence kinetic energy k and (optional) the energy dissipation ε for each boundary section concerned.
File format	Free formatted.
Generated	Manually offline.

**Remarks:**

- ◊ Keywords are case sensitive!
- ◊ Values of character-type *must* be enclosed between two number signs (#).
- ◊ More than one values on a line are space separated.

Record description:

Record	Record description
each record	Sequence number of the open boundary section (Data Group <i>Boundaries</i>) (1 integer), Interface number (1 integer), Time independent boundary condition for k (k - L turbulence model; 1 real) or k and ε (k - ε turbulence model; 2 reals).

**Restrictions:**

- ◊ The open boundary section must be specified in the same sequence as specified in the Data Group *Boundaries* (i.e. section 2 must always precede section 3 and so on).
- ◊ You may skip sections which do not require boundary condition for k and ε .
- ◊ The turbulence kinetic energy k and the energy dissipation ε are defined at interfaces between the layers. For a model with Kmax layers there are Kmax+1 interfaces. Consequently, Kmax+1 boundary condition values must be specified for each segment.
- ◊ Input items in a record must be separated by one or more blanks.

Example:

For a model containing four open boundary sections and three layers, you want to prescribe k and ε at open boundary sections two (variable k and uniform ε over the vertical) and boundary section four (k and ε uniform), for interface 0 to 4. The comments (not in the file) is enclosed between brackets.

```

2 0 0.000039 0.00005 {k and ε for open boundary section 2 at the surface}
2 1 0.000041 0.00005 {k and ε for open boundary section 2 between layers 1 and 2}
2 2 0.000047 0.00005 {item at interface between layer 2 and layer 3}
2 3 0.000051 0.00005 {item at interface between layer 3 and layer 4}
2 4 0.000052 0.00005 {item at the bottom layer}
4 0 0.000048 0.00005 {k and ε for open boundary section 4 at the surface}
4 1 0.000048 0.00005 {k and ε for open boundary section 4 between layers 1 and 2}
4 2 0.000048 0.00005 {item at interface between layer 2 and layer 3}
4 3 0.000048 0.00005 {item at interface between layer 3 and layer 4}
4 4 0.000048 0.00005 {item at the bottom layer}

```

B.13.2 Diagnostic mode

Keyword	Value	Description	Default
Prcusr	#diagnostic mode#	Use diagnostic mode	none
Nprinp	0 1 0 0	No filename One real No integers No strings	0 1 0 0
Rcousr	1 real [min]	Time in minutes after the simulation Reference Date, subsection 4.5.3	0.0

In the diagnostic mode you keep the distributions of the salinity, temperature and the concentrations of the conservative substances the same for all times after the time specified by the parameter Rcousr. Rcousr is given in minutes after the Reference Date, see Data Group *Time frame*, [subsection 4.5.3](#). If Rcousr is set to zero the initial condition is applied during the whole computation.

You can apply the diagnostic mode to investigate the influence of the quantities mentioned above, especially the salinity and the temperature, on the flow without having the feedback of the flow on the parameter of interest.

Restriction:



- ◊ You can not combine the diagnostic mode with:
 - ◊ secondary flows
 - ◊ 3D sediment transport
 - ◊ morphological computations

B.13.3 Particle wind factor

Keyword	Value and unit	Description	Default
Prcusr	#particle wind factor#	Additional wind drift of drogues	none
Nprinp	0 3 0 0	No filename Three reals No integers No strings	0 3 0 0
Rcousr	1 real [m/s]	Wind speed	0.0
Rcousr	1 real [degrees]	Wind direction relative to North	0.0
Rcousr	1 real [-]	Wind factor α_w	0.0

With the user defined function 'particle wind factor' you can add an additional drift to drogues.

This can be useful to take into account an additional wind effect on drogues, for instance because they are more exposed to the wind than drifting with the flow.

The wind direction is relative to North, positive clockwise and the nautical definition is used.

The velocity components of a drogue are increased by the additional wind drift as given by:

$$\vec{u}|_{new} = \vec{u}|_{old} + \alpha_w \cdot \vec{W}, \quad (\text{B.54})$$

where \vec{u} is the drogue velocity vector, \vec{W} is the wind vector and α_w is the wind factor specified as input.

B.14 Domain decomposition

B.14.1 Introduction

Domain decomposition is a technique in which a model is divided into several smaller model domains. The subdivision is based on the horizontal and vertical model resolution required for adequately simulating physical processes. Then, the computations can be carried out separately on these domains. The communication between the domains takes place along internal boundaries, or so-called *DD-boundaries*. If these computations are carried out concurrently, we speak of *parallel computing*. Parallel computing will reduce the turn around time of multiple domain simulations.

Domain decomposition allows for local grid refinement, both in horizontal direction and in vertical direction. Grid refinement in horizontal direction means that in one domain smaller mesh sizes (fine grid) are used than in other domains (coarse grid). In case of vertical grid refinement one domain e.g. uses ten vertical layers and an other domain five layers, or a single layer (depth-averaged).

Functionality

In Delft3D-FLOW domain decomposition can be applied, including local grid refinement in both horizontal and vertical direction. Domain decomposition is available in the (standard) operational version of Delft3D-FLOW. Thus, with the operational version both single domain and multiple domain simulations can be carried out. This appendix contains a detailed description of how domain decomposition in Delft3D-FLOW should be applied.

Hardware platforms

Multiple domain simulations with Delft3D-FLOW run on computers on which P-threading is available. This is the case for Windows and Linux.

The current implementation does not allow yet parallel computing on distributed memory systems, such as on a cluster of PCs.

B.14.2 Motivations for domain decomposition

Domain decomposition is widely recognised as an *efficient* and *flexible* tool for the simulation of complex physical processes. The advantages of a multi-domain modelling approach for flow and transport problems can be summarised by:

- 1 Modelling flexibility
 - ◊ coupling of different models
 - ◊ coupling of models of different dimension
 - ◊ coupling of models with different, independently generated grids
- 2 Modelling accuracy

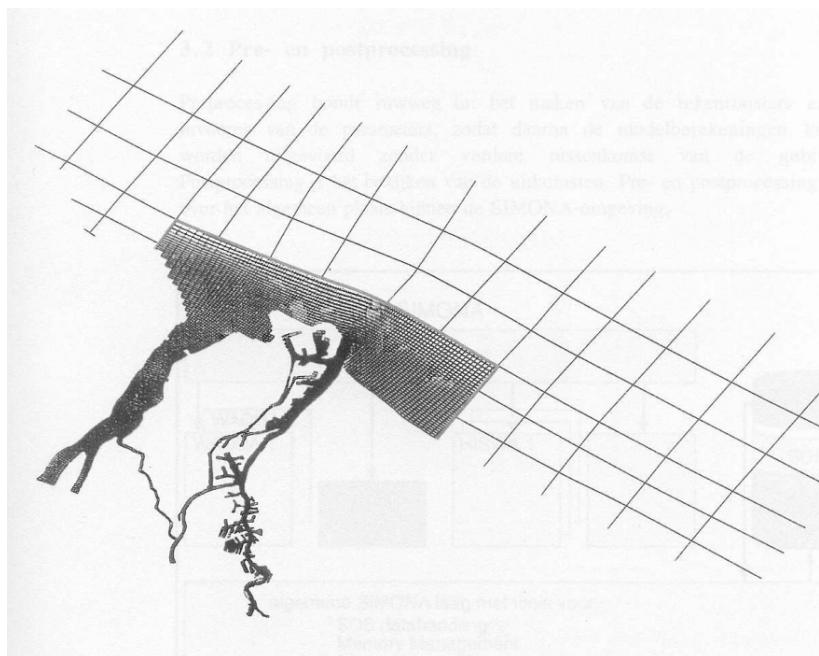


Figure B.22: Example of grid refinement

- ◊ smooth, structured grids per domain
- ◊ local grid refinements

3 Efficiency

- ◊ reduced memory demands by decomposition into smaller domains
- ◊ parallel execution of domains
- ◊ better software engineering and maintenance due to modular approach

It should be noted that numerical techniques other than domain decomposition may also offer some (but not all) of the above-mentioned advantages. The use of a fully unstructured grid, for example, would offer as much or even more flexibility in the specification of the computational grid. However, the drawbacks that come with that approach may be severe: the numerical scheme would be rather inaccurate or otherwise highly complex, and may turn out to be inefficient. At Deltares this and other considerations have led to the conclusion that a domain decomposition approach based on a subdivision of the model into non-overlapping domains, each covered by its own structured grid, is the numerical approach that meets best the demands of efficiency, accuracy and general applicability. The structured multi-domain approach combines the advantages of the modelling flexibility of the single-domain unstructured approach with the efficiency and accuracy of the single-domain structured approach.

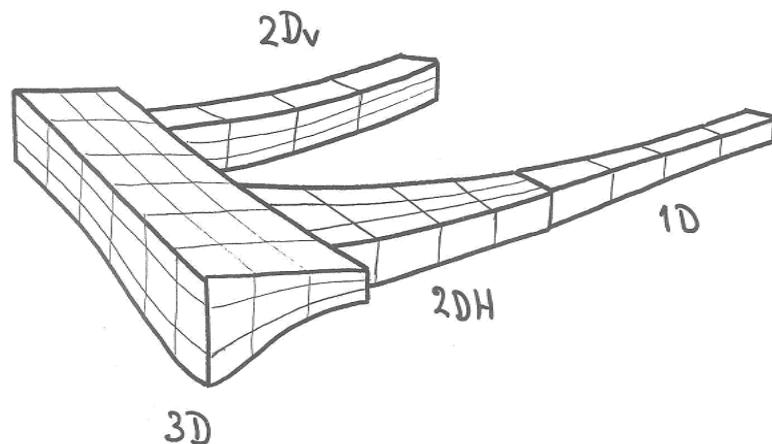


Figure B.23: Example of coupling of models with a different dimension

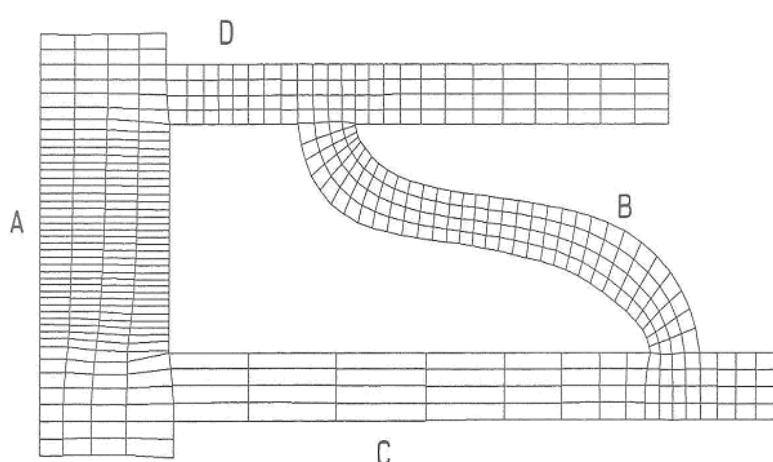


Figure B.24: Schematised island without domain decomposition

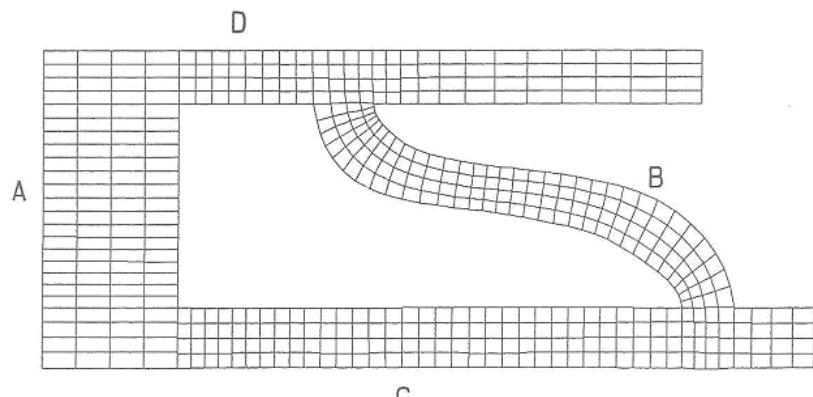


Figure B.25: Schematised island with domain decomposition

Flexibility of domain decomposition implementation

At Deltares the main motivation was increasing the modelling flexibility of the system rather than focussing on increasing the efficiency of the system by using many processors. The most well known example is the use of local grid refinement (see [Figure B.22](#)) and the use of domains with different dimensions. This is illustrated in [Figure B.23](#), in which a 3D model is coupled to a 2Dv (two-dimensional vertical), a 2Dh (two-dimensional horizontal) and a 1D (one-dimensional) model.

Less well known, but also quite important is the increased flexibility when modelling complex geometries. In practice, the model area can be quite complex due to, for example, irregular land boundaries and the existence of (a lot of) islands. A complex geometry in combination with the fact that very often a curvilinear grid is applied in Delft3D-FLOW may lead to a situation in which the set-up of an accurate and efficient model grid can be very tedious. This is illustrated in [Figure B.24](#) in which the geometry is even rather simple because there is only one island. When using the single domain version of Delft3D-FLOW the grid should be such that at both sides of the island the number of grid lines should be equal. Thus, at the left side A of the island the number of grid lines should be equal to the number at the right B (similarly, for the top and bottom side). As a result a lot of unnecessary grid points might be needed. In [Figure B.24](#) this is, for example, the case at the left side of the model domain. In [Figure B.25](#) the situation with domain decomposition is sketched. By defining at each side of the island a separate domain (called A, B, C and D) it is no longer required that the number of grid lines at both sides should be equal. Then, the number of grid lines can be chosen on the basis of accuracy considerations.

When using domain decomposition one often thinks of local grid refinement. However, the increased flexibility, such as illustrated in [Figure B.25](#), is also a major step forward.

Accuracy of domain decomposition implementation

The domain decomposition approach implemented in Delft3D-FLOW is based on a subdivision of the domain into *non-overlapping domains*, with the possibility for grid refinement in both the horizontal and vertical direction. This allows for a nearly optimal distribution of grid points. Furthermore, an efficient iterative method has been used for solving the discretised equations over the domains.

Owing to the overlap of only one grid cell, the spatial discretisation at or near the interface may be of lower order than the discretisations in the interior. Since, in general only a few domains are used, it is assumed that this approach is acceptable. In case of grid refinement a special approach at or near the interface is always required.

Multiple domain simulations with Delft3D-FLOW are not able to identically reproduce the results of a single domain simulation. Owing to the iteration procedure over the domains and the possibly lower order discretisations at the interface will lead to small differences. However, these differences are acceptable, which is confirmed in numerical experiments.

Robustness of domain decomposition implementation

In case of an explicit coupling of domains, the implementation can be done straightforwardly. This only requires an exchange of data between neighbouring domains. However, such an approach would lead to a reduced robustness of the system compared to the (single domain) Delft3D-FLOW system. Therefore, domain decomposition has been implemented in such a way that it has a similar robustness as the single domain Delft3D-FLOW code.

This has been achieved by using a direct iterative solver for the continuity equation that is comparable to the single domain implementation. For the momentum equations, the transport equation and the turbulence equations we used the so-called additive Schwarz method, which allows for parallelism over the domains. Upon convergence, almost the same numerical solution is obtained compared to the single domain case. This explicit type of iteration process is comparable to the corresponding iterative solution methods in the single domain code. Thus, the domain decomposition implementation of Delft3D-FLOW is based on a comparable robustness.

B.14.3 Local refinement horizontal and vertical

Implementation of local grid refinement requires special numerical techniques, in particular for interpolation between domains. We will distinguish grid refinement in vertical and horizontal direction. Grid refinement in the vertical is implemented rather straightforward, whereas grid refinement in the horizontal direction is rather complex.

Both for horizontal and vertical grid refinement it is required that grid lines in the coarse domain are continued in the fine domain. Thus, only “regular” couplings are allowed. In [Figure B.26](#) an example of grid refinement in the horizontal direction is shown (a 1-to-3 refinement).

Horizontal grid refinement

In Delft3D-FLOW local grid refinement along domain interfaces has been based on a 1-to-N refinement with N an integer number. This means that a grid cell near the interface in the coarse domain corresponds to N grid cells in the other refined domain.

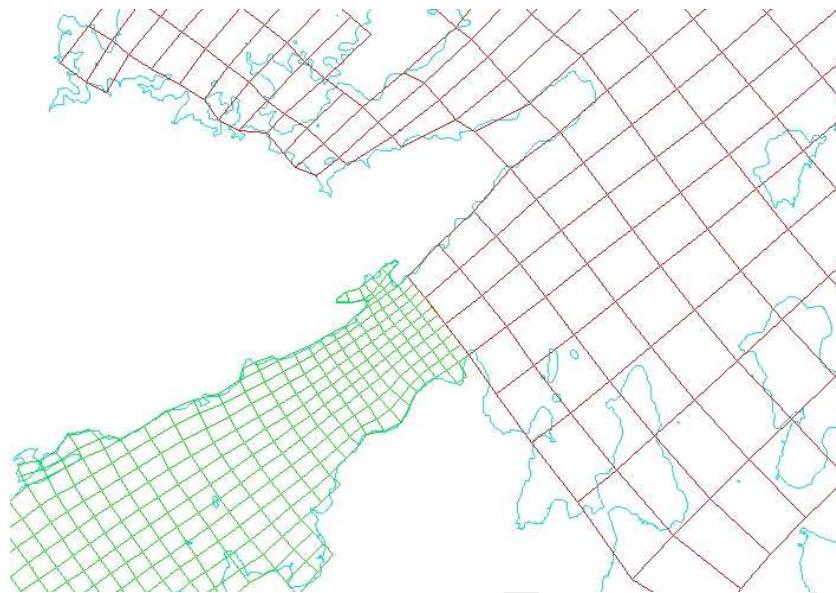


Figure B.26: Example of grid refinement in the horizontal direction

Vertical grid refinement

In case of grid refinement in the vertical, the implications are rather limited. Only in the momentum equations and the transport equation some interpolation has to be carried out. For the continuity equation, the implementation of vertical grid refinement does not require any adaptations.

In case of vertical grid refinement it is assumed that the number of layers in the refined domain is a integer multiple of the number of layers in the coarse domain. Moreover, the layer distribution is such that for each layer in the coarse domain its layer thickness is identical to the sum of the layer thicknesses of the corresponding layers in the neighbouring domain. In [Table B.12](#) an example of a possible refinement is given.

B.14.4 Pre-processing, processing and post-processing

Pre-processing

When running a single domain simulation with Delft3D-FLOW all relevant input data is stored in the so-called Master Definition Flow (MDF) file. For preparing such an input file we refer to [Chapter 4](#). When using domain decomposition in Delft3D-FLOW for each domain an MDF-file has to be specified, similarly to a single domain simulation. Thus, exactly the same input parameters have to be specified. It is advised to run all domains separately (a few time steps) in order to check the input files, before you start with coupling the domains into a domain decomposition simulation.

Furthermore, the following additional input file is required, which specifies which domains and where they are coupled:

A configuration file (`<name.ddb>`), which specifies all domains couplings and the location of the DD-boundaries.

How this file should be generated, is described in the RGFGRID User Manual, [RGFGRID UM \(2016\)](#).

Table B.12: Example of vertical grid refinement

Layer number		Layer thickness	
coarse	fine	coarse	fine
1	1	0.1	0.03
	2		0.07
2	3	0.2	0.09
	4		0.11
3	5	0.4	0.20
	6		0.20
4	7	0.2	0.11
	8		0.09
5	9	0.1	0.07
	10		0.03

Processing

The `<name.ddb>` file as output from RGFGRID has to be adjusted before this file can be used by FLOW: the grid filenames have to be replaced by the MDF-filename in which the grid is used. For example: if `<dom1.grd>` is used by `<runid1.mdf>`, then you have to replace all `<dom1.grd>` references in `<name.ddb>` by `<runid1.mdf>`.

A multi-domain simulation with Delft3D-FLOW is started with the appropriate input file. An example can be found at https://svn.oss.deltares.nl/repos/delft3d/trunk/examples/02_domaindecomposition. Note the *DDB* specification in the `<.ini>` and/or `<.xml>` configure file.

Post-processing

Not only an input file is required for each domain, also for each domain output files will be generated. In case of a single-domain simulation we end-up with one history file (`<trih-runid.def/dat>`), one map file (`<trim-runid.def/dat>`) and possibly other output files such as restart and communication files. In case of domain decomposition, for each domain (with its own runid) output files are generated. For example, in case of five domains we arrive at five history and map files. This means that post-processing will require more effort compared to a single domain simulation. For example, in case of post-processing with Delft3D-QUICKPLOT, to generate an overall contour map of water levels at a certain point in time, you should define for each domain a water level data set and assign all these data sets in the figure.

Summarising, when using domain decomposition with Delft3D-FLOW the standard way of plotting is used (Delft3D-QUICKPLOT).

B.14.5 Restrictions

As described in [subsection B.14.4](#) for each domain an MDF-file has to be prepared. However, this does not mean that arbitrary combinations of input parameters are allowed. For example, a restriction is that in all domains the same time step is applied. This and other restrictions are listed below:

Grid restrictions

- ◊ At domain interfaces the grids should be nicely connected (no overlap and “no holes” between domains).
- ◊ The maximum number of domain interfaces is limited to 50.
- ◊ In case of horizontal and vertical grid refinement, grid lines (or layers) in the coarse domain should be continued in the fine domain, see [Figure B.26](#) and [Table B.12](#). Thus, there should be a 1-to-N refinement, with N an integer number.
- ◊ In case of horizontal refinement it is advised to have an equidistant refinement.
- ◊ Grids must be of the same type (thus, all in spherical co-ordinates, or all in Cartesian co-ordinates).
- ◊ The grid orientation should be the same (increasing M- and N-numbering in the same direction).
- ◊ No coupling of columns to rows or vice versa.
- ◊ Domain interfaces should be straight lines (no staircase interfaces).
- ◊ When coupling a subdomain inside a coarse domain (as a way of nesting), the hole in the coarse domain must be at least 3 rows and 3 columns.

Bathymetry

- ◊ At domain interfaces the depth values in corresponding depth points should be identical in both domains.
- ◊ In case of grid refinement corresponding depth points should have identical depth values (see also previous point). Furthermore, for the refined model, it is proposed that the depth values in the intermediate points are determined by linear interpolation.

Model input

- ◊ Each MDF-file should satisfy the requirements for a single domain Delft3D-FLOW simulation.
- ◊ In all domains the same simulation period and time step should be used.
- ◊ In all domains the same processes for transport should be used (e.g. salinity and temperature). Note that turbulence processes may vary (e.g. $k-\varepsilon$ in one domain and an Algebraic turbulence model in another domain).
- ◊ No permanently dry points are allowed along domain interfaces. Note that temporary dry points are allowed.
- ◊ Thin dams may be specified perpendicular to a domain interface, but not parallel.
- ◊ Part of Delft3D-FLOW functionality does not work in combination with domain decomposition:
 - Direct coupling with water quality module
 - Z-model (strictly horizontal layers)
 - Internal Wave model
 - Fluid Mud
 - Drogue
- ◊ It is advised to avoid special points (see [section 10.9](#)) at domain interfaces.
- ◊ Though not strictly necessary it is advised to have the output timings the same in all domain models.

Table B.13: Example of a `<name.ddb>` file

DOM1.GRD	9	31	21	31	DOM2.GRD	12	1	16	1
DOM4.GRD	1	4	1	10	DOM1.GRD	9	1	9	2
DOM3.GRD	5	61	5	65	DOM2.GRD	1	1	1	5
DOM1.GRD	1	1	1	5	DOM3.GRD	5	14	5	26

- ◊ Online visualisation can be activated for each domain.
- ◊ If you want to visualise all domains in the online visualisation, you have to set *Output* → *Refresh screen at each step* off.

B.14.6 How to set-up a domain decomposition model

When applying domain decomposition with Delft3D-FLOW there are two different options for the user:

Subdivision of an existing model: You start with an existing model and subdivide this overall domain into smaller domains. Next, one or more domains may be refined to enhance the accuracy of the model application.

Combining separate models: You start with separate, independent models that are combined into a multi domain configuration. Some of the domains may be more refined at the interface than other domains, as long as all interface points in the coarse domain coincide with interface points in the refined domain. Often, the corresponding grid points do not coincide and have to be adapted.

Subdivision of an existing model

Generating grids of domains

When dividing an existing model into several domains a few actions have to be carried out manually. At first, the grid of the model area has to be divided into smaller sub-grids. The grid generator RGFGRID should be applied to create a grid for each domain.



Remark:

- ◊ We remark that it is your responsibility to define sub-grids that are connected at the interfaces. Moreover, there should be no overlap or “gaps” between the domains.

Defining domain interfaces

RGFGRID has been extended with the option to define domain interfaces. We presume that the grids already satisfy the rules for domain decomposition.

An example of a `<name.ddb>` file is given in Table B.13.

In this example there is a subdivision into four sub-grids (dom1.grd, dom2.grd, etc.). The grid indices of corresponding interface segments are specified. For example, the interface at the first line indicates that ($M=9-21$, $N=31$) in domain dom1.grd corresponds to ($M=12-16$, $N=1$) in domain dom2.grd. So, this is a 3-to-1 coupling.

The `<name.ddb>` file is the starting point for a Delft3D-FLOW multi-domain simulation.

The `<name.ddb>` file as output from RGFGRID has to be adjusted before this file can be used by FLOW: the grid filenames have to be replaced by the MDF-filename in which the grid

is used. For example: if `<dom1.grd>` is used by `<runid1.mdf>`, then you have to replace all `<dom1.grd>` references in `<name.ddb>` by `<runid1.mdf>`.

Combining separate models

When combining independently generated domain grids to one multi-domain model, the actions to be carried out are quite similar to the ones for subdividing an existing model. The most important difference is that, when combining independent models, the grid boundaries may not coincide. Therefore, at first, the model grid has to be adapted so that grid points at the proposed domain interfaces coincide. This can be done in RGFGRID.

B.15 Surfbeat/roller model

This extension of the original Delft3D program allows the modelling of the effect of short-wave groups on long waves. This effect is caused by spatial variations in the radiation stresses and causes long waves to travel along with groups of short waves (called carrier waves). The long waves attached to the groups of short waves are called locked waves or bound waves. The groups of short waves and thus the long waves attached to these groups travel with the group velocity (C_g) of the carrier waves.

The Delft3D program does not model the individual carrier waves but it does model the forcing caused by the short waves. It can only be used in cases where the wave spectrum is narrow-banded both with respect to frequency as with respect to direction. Given these limitations it is possible to determine a dominant frequency (the peak frequency) and a dominant direction from the wave field information. These two parameters are then used to determine a directional field which indicates the direction of propagation of the carrier waves. This is also the direction in which the short-wave energy propagates. This short-wave energy is a quantity that is modelled by Delft3D and it is directly determined from the short-wave components provided at the boundaries (we will come to these later). The short-wave energy is transported into the domain travelling with the group velocity which is based on the peak frequency.

The mean wave direction field along which wave energy and roller energy are transported is not predicted by the surfbeat model but imported from SWAN via the Delft3D-WAVE module.

B.15.1 Domain

A typical domain for the application of the roller model is given by a stretch of sea along the coast. From the sea side the coast is attacked by a directionally spread wave field. The domain may contain a harbour basin, in which case the model can be used to predict harbour oscillations caused by wave groups and the long waves released from these groups as they approach the harbour.

A sketch of a typical problem layout is given in [Figure B.27](#).

In [Figure B.27](#) Boundary A represents the sea boundary. At this boundary the depth must be such that incoming bound and free waves can be considered long waves as to allow the shallow water approximation to be valid. Boundary C represents the coast. It is modelled as an impermeable boundary at a small depth. Boundaries B and D are the lateral boundaries. In the sketch we show the case where a locked wave enters the domain at a. Due to refraction of the carrier wave groups (which force the locked wave) the trajectory curves toward the normal at the coast. Wave breaking causes the carrier waves to vanish thereby releasing the locked wave. This is a gradual process that takes place along the shallow part of the trajectory a-b. The free (released) wave reflects at the coast and travels seaward along trajectory b-c. As the waves are released gradually and the free waves refract more due to bottom changes

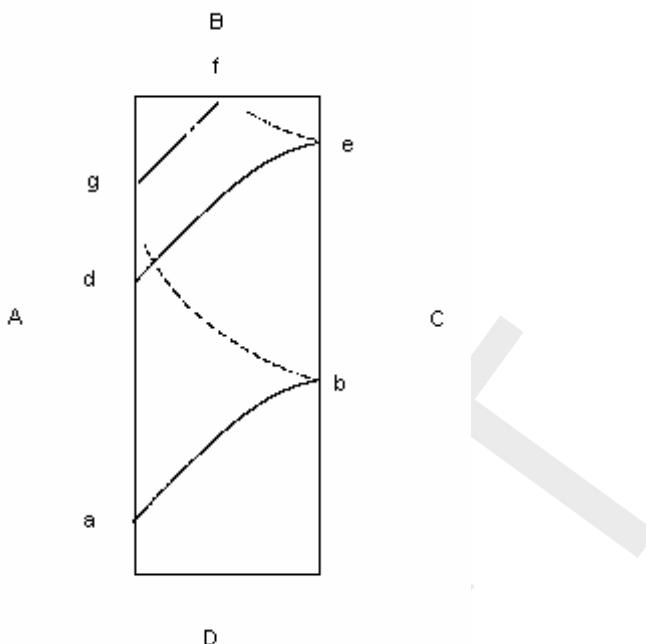


Figure B.27: Problem layout sketch

(they are long waves) than the carrier waves there will be a range of directions to the right of the curve a–b in which free waves propagate. The curve b–c represents these waves after reflection at the coast. At the sea boundary the free waves must be allowed to leave the computational domain without (significant) reflections. At d bound waves also enter the domain. They propagate along curve d–e. Reflected free waves have to leave at the lateral boundary B now (curve e–f). Bound waves entering the domain at g should leave the domain at the lateral boundary B directly (curve g–f). If wave breaking takes place along the path g–f some of the waves will be released and free waves will also need to leave the domain at lateral boundary B.

B.15.2 Formulations

The short wave energy balance reads:

$$\frac{\partial E}{\partial t} + \frac{\partial}{\partial x} (EC_g \cos(\alpha)) + \frac{\partial}{\partial y} (EC_g \sin(\alpha)) = -D_w \quad (\text{B.55})$$

(α is user-defined)

Here E is the short-wave energy, C_g the group velocity, α the wave direction and D_w the dissipation of wave energy, for which we use the expression by [Roelvink \(1993\)](#), which was derived for the situation of propagating wave groups where the energy varies slowly:

$$D_w = 2\alpha f_m \left(1 - \exp \left(- \left(\frac{\sqrt{8E/(\rho g)}}{\gamma h} \right)^n \right) \right) E \quad (\text{B.56})$$

(γ and n are user-defined)

Through the process of wave breaking the wave energy is reduced and transformed into roller energy. This energy is located in the down-wave region after wave breaking. Spatial

variations in the roller energy also generate forces on the water. The propagation of this form of energy is not yet well understood. Ignoring the effect it has on the generation of long waves is not an option as recent studies have shown. We model the propagation of roller energy by transporting it with twice the local celerity of the carrier waves. The roller energy is rapidly dissipated in shallow regions. The energy that is lost from the organised wave motion is converted to roller energy through the roller energy balance:

$$\frac{\partial E_r}{\partial t} + \frac{\partial}{\partial x} (2E_r C \cos(\alpha)) + \frac{\partial}{\partial y} (2E_r C \sin(\alpha)) = D_w - D_r \quad (\text{B.57})$$

where C is the wave celerity. The roller energy dissipation D_r is a function of the roller energy E_r :

$$D_r = 2\beta g \frac{E_r}{C} \quad (\text{B.58})$$

Here β is a user-defined coefficient of approximately 0.1 and g the acceleration of gravity.

The time-and space-varying wave energy and roller energy cause a variation in radiation stresses (vertically averaged), through the following relations:

$$S_{xx} = \left(\frac{C_g}{C} (1 + \cos^2(\alpha)) - \frac{1}{2} \right) E + 2 \cos^2(\alpha) E_r \quad (\text{B.59})$$

$$S_{xy} = S_{yx} = \sin(\alpha) \cos(\alpha) \left(\frac{C_g}{C} E + 2E_r \right) \quad (\text{B.60})$$

$$S_{yy} = \left(\frac{C_g}{C} (1 + \sin^2(\alpha)) - \frac{1}{2} \right) E + 2 \sin^2(\alpha) E_r \quad (\text{B.61})$$

The radiation stresses are subdivided into surface stresses and depth-uniform stresses. Here, the surface stresses will always generate a (vertical) circulation. This is only allowed in regions where dissipation takes place. As the roller model has been introduced to delay the transfer of wave energy to the current, the shear stress related to the roller is the only surface stress that is applied. This surface stress reads:

$$F_{x,r} = \frac{D_r}{C} \cos(\alpha) \quad (\text{B.62})$$

$$F_{y,r} = \frac{D_r}{C} \sin(\alpha) \quad (\text{B.63})$$

The depth-invariant part of the radiation stresses is now simply the total radiation stress gradient minus the surface stress:

$$F_{w,x} = - \left(\frac{\partial S_{xx}}{\partial x} + \frac{\partial S_{xy}}{\partial y} \right) - F_{x,r} \quad (\text{B.64})$$

$$F_{w,y} = - \left(\frac{\partial S_{xy}}{\partial x} + \frac{\partial S_{yy}}{\partial y} \right) - F_{y,r} \quad (\text{B.65})$$

B.15.3 Boundary conditions

At the boundary, the bound wave is determined from the local short-wave energy field using the horizontal-bottom approximation:

$$\zeta_b = \frac{2n - \frac{1}{2}}{\rho (C_g^2 - gh)} (E - \bar{E}) \quad (\text{B.66})$$

$$u_b = \frac{C_g}{h} \cos(\theta_b) \zeta_b \quad (\text{B.67})$$

$$v_b = \frac{C_g}{h} \sin(\theta_b) \zeta_b \quad (\text{B.68})$$

Here we have assumed that the average of ζ_b is zero. The time averaged wave energy is given by

$$\bar{E} = \frac{1}{2} \rho g \sum_{j=j_s}^N a_j^2 \quad (\text{B.69})$$

The user-defined parameter f_s is used to distinguish long waves from short waves. Wave components with a frequency smaller than f_s are considered long and are prescribed at the boundary as free waves (provided the direction of these waves is incoming at the boundary).

The free surface elevation and the x and y components of the velocity of the free incoming waves are given respectively by:

$$\eta_i = \sum_{j=1}^{j_s} a_j \cos((\cos(\theta_j) x_b + \sin(\theta_j) y_b) k_j - \omega_j t + \varphi_j) \quad (\text{B.70})$$

$$u_i = \sum_{j=1}^{j_s} \frac{a_j \omega_j}{k_j h} \cos(\theta_j) \cos((\cos(\theta_j) x_b + \sin(\theta_j) y_b) k_j - \omega_j t + \varphi_j) \quad (\text{B.71})$$

and

$$v_i = \sum_{j=1}^{j_s} \frac{a_j \omega_j}{k_j h} \sin(\theta_j) \cos((\cos(\theta_j) x_b + \sin(\theta_j) y_b) k_j - \omega_j t + \varphi_j) \quad (\text{B.72})$$

In these expressions it is assumed that all of the free waves are incoming at the boundary. If this is not the case the outgoing components have to be ignored in the summations.

The Riemann boundary condition uses the information of the incoming free and bound waves and the outgoing reflected wave (which is also assumed to be a free wave). The direction of the outgoing wave is computed from the velocity and surface elevation in the previous time step.

The parameters θ_i , θ_r and θ_b are the angles between the direction of propagation and the normal at the boundary. \vec{n} is the inward pointing normal unit vector at the boundary. We use the abbreviations $c_i = \cos(\theta_i) = \vec{n} \cdot \vec{e}_i$, $c_r = \cos(\theta_r) = -\vec{n} \cdot \vec{e}_r$, $c_b = \cos(\theta_b) = \vec{n} \cdot \vec{e}_b$. The weakly reflective boundary condition can be written as:

$$u + \zeta \sqrt{\frac{g}{h}} c_r = \sqrt{\frac{g}{h}} \left(\zeta_i (c_r + c_i) + \zeta_b \left(c_r + \frac{C_g}{\sqrt{gh}} c_b \right) \right). \quad (\text{B.73})$$

Here u denotes the normal component of the depth averaged velocity vector: $\vec{u} \cdot \vec{n}$.

Since the boundary condition only allows reflected (free) waves to leave the domain, situations where bound waves approach the boundary have to be avoided. This can be achieved by defining strips along such boundaries where the wave forces (which drive the bound waves) are gradually reduced towards the boundary. The bound waves travelling towards the open boundary are thus gradually transformed into free waves which can be dealt with using Eq. B.73. The location of the strips is provided by you through the input file as described in subsection B.15.6. It is advised to choose for the width of the strip at least the length of a typical wave group.

References: [Van Dongeren and Svendsen \(1997\)](#) and [Roelvink \(1993\)](#).

B.15.4 Coupling with other modules

The model needs wave direction and period information from SWAN, which it reads from the communication file in the same way as it does this without the surfbeat/roller option.

The surfbeat/roller model can be combined with the online sediment and morphology features.

B.15.5 Modes of operation

The surfbeat/roller model can be run with quasi-stationary boundary conditions or with the instationary boundary conditions described above. In the first case, the boundary conditions are taken over from the wave information on the communications file; the main purpose of the surfbeat/roller model is then to include the roller equations, which leads to a shoreward shift of the wave set-up and the longshore and cross-shore flow. In this case, very little extra information is needed, except to switch the roller model on and to specify some coefficients.

In the case of instationary boundary conditions, spectral components of the surface elevation at a point with given depth are converted to boundary conditions for short-wave energy and free long wave elevation at all boundaries as outlined above. In this case an additional file with the spectral components and some additional information must be specified.

B.15.6 Input description

To set up a model including the wave energy and roller energy balance, first a standard wave-flow model must be set up, as described in the WAVE and FLOW manuals. A set-up must be chosen where a dummy FLOW call sets up the administration, after which a WAVE computation is carried out for a single time point, followed by a FLOW run that takes into account wave forcing.

For the typical small coastal models where we apply the roller model in **stationary** mode we recommend a combination of water level boundaries for the offshore boundary and Neumann boundaries for the lateral boundaries. Also add the keyword:

```
Cstbnd=#yes#
```

This helps avoiding the formation of artificial boundary layers along the domain boundaries. These may occur due to normal components of the advection terms.

Now we can turn on the roller model in stationary mode by specifying in the MDF-file the extra keyword:

```
Roller =#yes#
```

Optionally, the roller model parameters can be specified if values other than their default values as given below are required, by adding the keyword/value combinations:

```
Alfaro   = 1.0
Betaro   = 0.1
Gamdis   = 0.55
Ndis     = 10
```

(Alfaro see [Equation B.55](#), Betaro see [Equation B.58](#), Gamdis and Ndis see [Equation B.56](#))

These keywords can be entered using a text editor or (preferably) through the FLOW user interface, in Data Group *Additional parameters*.

In *instationary* mode, for the boundary conditions of the surfbeat model to work properly it is necessary to set all flow boundaries to Riemann type. The input for the Riemann boundary conditions will be generated automatically based on the given wave spectrum; a dummy boundary condition file <*.bct> or <*.bch> has to be created, i.e. by doing a ‘save all’ in the FLOW user interface.

The input wave spectrum that is used to create the instationary boundary conditions has to be given in a file specified in the mdf file through a keyword:

```
Filwcm= #wavcmp#
```

Here <wavcmp> may be any legitimate filename. The contents of this file are described below.

Wave components file

In this section we will describe the structure of the file <wavcmp> that prescribes the Fourier components of the incoming short- and long- wave signals. The following records must be provided in free format; after the prescribed number of values in each record, any comment can be added as only the required values per record are read.

Record	Record description
1	The file may start with an arbitrary number of description records starting with an asterisk (*). These lines will be ignored except for the line containing the version number which should read “* version 1”.
2	N_c
3	h
4	f_{split}
5 to $4 + N_c$	$f_j \quad a_{bc,j} \quad \varphi_{bc,j} \quad \theta_{bc,j}$
$5 + N_c$	nmskf nmskl mmskf mmskl
$6 + N_c$	timtap [sec]

N_c is the number of spectral components.
 h is the characteristic water depth in [m] at the open boundaries.
 f_{split} is the splitting frequency in [Hz] between free long waves and short waves. Only the components which have a frequency lower than f_{split} will be prescribed at the boundary as free waves. The other components are used to determine the incoming high frequency wave energy field which will generate forced waves in the model.

f_j is the frequency of wave component j in [Hz].

Warning:

- ◊ The wave components should be specified in order of increasing wave frequency, i.e. $f_j < f_p$ if $j < p$.



$a_{bc,j}$ is the amplitude of wave component j in [m].
 $\varphi_{bc,j}$ is the phase angle of the incoming wave component j in [deg] at $x = 0, y = 0$.
 $\theta_{bc,j}$ is the direction of the incoming wave component j in [deg] in Cartesian convention, i.e. $\theta_{bc,j} = 90$ represents a wave travelling to the north.
The wave components generate the following free surface elevation relative to the tidal elevation:

$$\eta(x, y, t) = \sum_{j=1}^{N_c} a_{bc,j} \exp \{i (k_j \cos(\theta_{bc,j}) x + k_j \sin(\theta_{bc,j}) y - \omega_{bc,j} t + \varphi_{bc,j})\} \quad (B.74)$$

Here k_j is related to f_j and prescribed water depth h through the linear dispersion relation and $\omega_j = 2\pi f_j$.
nmskf is the grid number in η dir below which the wave forces are gradually and artificially reduced to zero at the lower boundary. Set to -1 if the wave forces should not be reduced. At boundaries at which waves enter, this parameter should be -1.

nmskl	is the grid number in η dir above which the wave forces are gradually and artificially reduced to zero at the upper boundary. Set to -1 if the wave forces should not be reduced. At boundaries at which waves enter, this parameter should be -1.
mmskf	is the grid number in ξ dir below which the wave forces are gradually and artificially reduced to zero at the left boundary. Set to -1 if the wave forces should not be reduced. At boundaries at which waves enter, this parameter should be -1.
mmskl	is the grid number in ξ dir above which the wave forces are gradually and artificially reduced to zero at the right boundary. Set to -1 if the wave forces should not be reduced. At boundaries at which waves enter, this parameter should be -1.
timtap	is the time in seconds that is used by the taper for the incoming signals. It is advised to define timtap at least several times as large as the period of a typical group in the signal.

**Remark:**

- ◊ The parameters nmskf, nmskl, mmskf and mmskl can in general be set to -1.

Determining wave components from a given spectrum

Here we distinguish between an analytically prescribed spectrum and a numerically provided spectrum. As an example of the first we choose the combination of a JONSWAP spectrum and a cos-m distribution for the directional spreading of the waves.

JONSWAP-cos-m

We assume the wave energy spectrum to be of the following form:

$$S(f, \theta) = D(\theta) E(f) \quad (\text{B.75})$$

where

$$D(\theta) = \frac{\Gamma\left(\frac{1}{2}\right) \Gamma\left(\frac{m+1}{2}\right)}{\Gamma\left(\frac{m}{2} + 1\right)} \cos^m(\theta - \theta_0) \quad (\text{B.76})$$

and JONSWAP frequency spectrum:

$$E(f) = A_0 \nu^{-5} \exp\left(-\frac{5}{4} \nu^{-4}\right) \gamma_0^{\exp(-(\nu-1)^2/(2\sigma^2))} \quad (\text{B.77})$$

$$\text{with } \nu = f/f_m \text{ and } \sigma = \begin{cases} 0.07, & \nu \leq 1 \\ 0.09, & \nu > 1 \end{cases}$$

The parameters θ_0 , m , γ_0 , f_m and A_0 are assumed to be known constants. The parameter A_0 is related to the significant wave height through:

$$H_{\text{sig}}^2 = 16 \int_0^\infty E(f) df \quad (\text{B.78})$$

The method described here is the 'single summation method' where one direction is assigned to each frequency bin. This direction is determined by using the directional spreading function for the frequency bin to define a probability density function and then randomly draw from this distribution.

In the case of a cos-m directional spreading this can be transformed to drawing from a uniform distribution on [0,1], yielding a value for p , and then solving the problem:

$$\frac{\Gamma\left(\frac{1}{2}\right)\Gamma\left(\frac{m+1}{2}\right)}{\Gamma\left(\frac{m}{2}+1\right)} \int_{-\pi/2}^{\alpha} \cos^m(\beta) d\beta = p \quad (\text{B.79})$$

The solution α is then used to determine the direction $\theta_j = \theta_0 + \alpha$ which is assigned to this frequency bin. If we use an equidistant grid (with grid size Δf) for the frequency, the energy in one frequency bin can be approximated by $\Delta f E(j\Delta f) = \frac{1}{2}a_j^2$, where a_j is the amplitude assigned to the bin. The phase φ_j of the j -th component is drawn from a uniform distribution on $[0, 2\pi]$

The wave component to be used in the file <wavcmp> is now given by the string: $j\Delta f, a_j, \varphi_j, \theta_j$

Determining the components in the case of a numerical spectrum

In the case where the spectrum is not given analytically as in the former section but numerically we can follow the same procedure to find the single summation components. Here we assume that the spectrum is given on a equidistant grid: $[\theta_i, f_j], i = 0, 1, \dots, M$ and $j = 0, 1, \dots, N$. We furthermore assume that the spectrum has one peak inside the domain $[\theta_0, \theta_M] \times [f_0, f_N]$ around which most of the energy is concentrated. At the boundary of this domain the spectrum is assumed to be small. For each frequency bin we approximate the directional spreading $D(\theta, f_j)$ through linear interpolation such that

$$D(\theta_i, f_j) = S(\theta_i, f_j) \left/ \left(\Delta\theta \left(\frac{1}{2}S(\theta_0, f_j) + \frac{1}{2}S(\theta_M, f_j) + \sum_{p=1}^{M-1} S(\theta_p, f_j) \right) \right) \right. \quad (\text{B.80})$$

We again have to solve:

$$\int_{\theta_0}^{\theta} D(\tau, f_j) d\tau = p \quad (\text{B.81})$$

where p is drawn from a uniform distribution on $[0, 1]$.

We find i_0 such that

$$\frac{1}{2}\Delta\theta \sum_{i=1}^{i_0} (D(\theta_i, f_j) + D(\theta_{i-1}, f_j)) \leq p < \frac{1}{2}\Delta\theta \sum_{i=1}^{i_0+1} (D(\theta_i, f_j) + D(\theta_{i-1}, f_j)) \quad (\text{B.82})$$

and interpolate linearly to find:

$$\theta_j = i_0 \Delta\theta + \frac{p - \Delta\theta \left(\frac{1}{2}(D(\theta_0, f_j) + D(\theta_{i_0}, f_j)) + \sum_{i=1}^{i_0-1} D(\theta_i, f_j) \right)}{\frac{1}{2}(D(\theta_{i_0}, f_j) + D(\theta_{i_0+1}, f_j))}. \quad (\text{B.83})$$

The amplitude of the component corresponding with the frequency bin is found from:

$$\Delta f \Delta\theta \left(\frac{1}{2}S(\theta_0, f_j) + \frac{1}{2}S(\theta_M, f_j) + \sum_{p=1}^{M-1} S(\theta_p, f_j) \right) = \frac{1}{2}a_j^2. \quad (\text{B.84})$$

The phase φ_j is again drawn from a uniform distribution on $[0, 2\pi]$.

The wave component is again given by the string: $j\Delta f, a_j, \varphi_j, \theta_j$.

B.16 Bedform heights

The bedform height functionality allows you to include a dynamic dune height predictor or ripple, mega-ripple, and dune roughness height predictors in the simulation. The formulations have been described in Section 9.11.1. Several keywords in the MDF file influence the functioning; these are indicated in Table B.14. If `Bdf` is false, then the keyword `BdfRpC`, `BdfRpR`, `BdfMrC`, `BdfMrR`, `BdfDnC`, `BdfDnR` and `BdfOut` will still be processed for use by the [Van Rijn \(2007\)](#) formulations.

Table B.14: Bedform keywords in mdf file

Keyword	Value	Description	Default
<code>Bdf</code>	#Y# or #N#	Switch for dune height predictor	#N#
<code>BdfMor</code>	#Y# or #N#	Switch for time scale to be used for relaxation processes; by default the bedform adapt based on the hydrodynamic time step. Set #Y# if the morphological time step should be used (equal to the flow time step multiplied by the morphological factor). If <code>BdfMor</code> is true, then bedforms will not adapt during the initial morphological period as specified by <code>MorStt</code> .	#N#
<code>BdfH</code>	#vanrijn84# #fredsoempm# #fredsoeeh# #powerrelation#	Dune height predictor	#vanrijn84#
<code>BdfEps</code>	pos. real	Multiplication factor ε in case <code>BdfH</code> equals #vrijn84#, #fredsoempm# or #fredsoeeh#	1.0
<code>BdfThc</code>	pos. real	Critical bed shear stress used in dune height predictor. Set to a positive value to overrule the default value based on the Shields curve.	Shields curve
<code>BdfaH</code>	pos. real	Multiplication factor a_H in case <code>BdfH</code> equals #powerrelation#	0.0
<code>BdfbH</code>	pos. real	Exponent b_H in case <code>BdfH</code> equals #powerrelation#	1.0
continued on next page			

Table B.14 – continued from previous page

Keyword	Value	Description	Default
BdfRlx	#none# #thconst# #lhconst# #lhfactor# #lhchhmax#	Relaxation behaviour	#none#
BdfT_H	pos. real	Relaxation time scale T_H in [Tunit] in case BdfRlx equals #thconst#	0.0
BdfL_H	pos. real	Relaxation distance L_H in [m] in case BdfRlx equals #lhconst#	0.0
BdfLHc	pos. real	Multiplication factor f_{LH} in case BdfRlx equals #lhfactor#	0.0
BdfHmx	pos. real	Critical water depth H_{\max} in [m] in case BdfRlx equals #lhchhmax#	1.0
BdfLHp	pos. real	Exponent b_{LH} in case BdfRlx equals #lhchhmax#	2.5
BdfPmx	pos. real	Limiter ϕ_{\max} in case BdfRlx equals #lhchhmax#	3.0
BdfADV	#Y# or #N#	Switch for dune migration (this option becomes only active if dune relaxation is included, which is not the case by default)	#Y#
BdfCFL	#Y# or #N#	CFL check for dune migration	#N#
BdfaC	pos. real	Multiplication factor a_C in case BdfRlx equals #thconst#, #lhconst# or #lhfactor#	0.0
BdfbC	pos. real	Exponent b_C in case BdfRlx equals #thconst#, #lhconst# or #lhfactor#	1.0
BdfChn	pos. real	Multiplication factor a_{cH} in case BdfRlx equals #lhchhmax#	5.5
BdfChp	pos. real	Exponent b_{cH} in case BdfRlx equals #lhchhmax#	3.5
BdfGmn	pos. real	Limiter γ_{\min} in case BdfRlx equals #lhchhmax#	0.2
continued on next page			

Table B.14 – continued from previous page

Keyword	Value	Description	Default
BdfUni	pos. real or file-name	Initial dune height in [m]: uniform value or filename <*.dnh> with non-uniform values at cell centres. In case of restarting from a previous simulation with simulated dune height, the results of the previous simulation overrule the value(s) specified here.	0.0
BdfL	#vanrijn84# #powerrelation#	Dune length predictor	#vanrijn84#
BdfaL	pos. real	Multiplication factor a_L in case BdfL equals #powerrelation#	0.0
BdfbL	pos. real	Exponent b_L in case BdfL equals #powerrelation#	1.0
BdfRou	#vanrijn07# #vanrijn84# #powerrelation#	Roughness height predictor. The default is #vanrijn84# if dune height predictor is included in the simulation (Bdf is true), and #vanrijn07# otherwise since it predicts the roughness height directly.	#vanrijn84# or #vanrijn07#
BdfRpc	pos. real	Ripple calibration factor α_r in case BdfRou equals #vanrijn07#	1.0
BdfRpR	pos. real	Ripple relaxation time T_r in [Tunit] in case BdfRou equals #vanrijn07#	0.0
BdfMrC	pos. real	Mega-ripple calibration factor α_{mr} in case BdfRou equals #vanrijn07#	0.0
BdfMrR	pos. real	Mega-ripple relaxation time T_{mr} in [Tunit] in case BdfRou equals #vanrijn07#	0.0
BdfDnC	pos. real	Dune calibration factor α_d in case BdfRou equals #vanrijn07#	0.0
BdfDnR	pos. real	Dune relaxation time T_d in [Tunit] in case BdfRou equals #vanrijn07#	0.0
continued on next page			

Table B.14 – continued from previous page

Keyword	Value	Description	Default
BdfaR	pos. real	Multiplication factor a_R in case BdfRou equals #power-relation#	0.0
BdfbR	pos. real	Exponent b_R in case BdfRou equals #powerrelation#	1.0
BdfOut	#Y# or #N#	Flag to indicate whether dune height/length and/or bedform roughness height data should be written to the Delft3D-FLOW map-file. If you want to reuse dune height/length information in a subsequent simulation upon restart, then you need to set this flag to #Y#.	#N#

B.17 Trachytopes

The trachytope functionality allows for the usage of different types of roughness formulations at different locations within the computational domain. Multiple formulation may be active in the same grid cell. Several keywords in the MDF file influence the functioning.

Keyword	Value	Description	Default
Trtrou	#Y# or #N#	Trachytope option activated	#N#
Trtdef	#name#	Definition file trachytopes	
Trtu	#name#	Area file trachytopes u -direction	
Trtv	#name#	Area file trachytopes v -direction	
TrtDt	pos. real	Time step in minutes for updating roughness and resistance coefficients based on trachytopes. Must be a multiple of Dt.	1 Dt
TrtClu	#name#	Spatial calibration factor for trachytopes u -direction	uniform 1
TrtClv	#name#	Spatial calibration factor for trachytopes v -direction	uniform 1
TrtMnH	pos. real	Minimum water depth in roughness computation	Dryflc
TrtMth	1 or 2	Area averaging method	1
TrtAsr	real $\in [0, 1]$	Serial factor in averaging of area roughnesses	0.6

Using trachytypes does not eliminate the need for defining the standard roughness input. The standard roughness values will be overruled by the trachytype defined values for those locations for which trachytypes have been defined. When trachytypes cover less than 100% of a grid cell, the standard roughness value is used as a default roughness value for the remaining percentage of the grid cell. The roughnesses based on the trachytypes are updated based on the interval specified by `TrtDt`. When `TrtDt` is larger than 1, the roughnesses are kept constant expressed roughness formulation chosen for the default roughness (Roumet).

The keywords refer to three (types of) attribute files, namely

- 1 Trachytype definition file for defining the different types of trachytypes and the associated roughness formulations (`Trtdef`). This file is described in [subsection B.17.1](#).
- 2 Area files for defining the fraction of the grid cell covered by a certain area trachytype type (`Trtu/Trtv`). It also contains the data for the linear and point trachytype types. This file is described in [subsection B.17.2](#). If the trachytype definition file defines only one trachytype, then the area files are optional: if the area files are not specified in such a case, then all points will use that one trachytype.
- 3 Calibration files (`TrtClu/TrtClv`). These are files with positive numbers in QUICKIN/Delft3D DEP file format.

B.17.1 Trachytype definition file

The trachytype definition file contains lines of the following format defining the different types of trachytypes.

```
TrachytypeNr      FormulaNr      ...Parameters...
```

where `...Parameters...` indicates a space separated list of formula specific parameters; the parameters required and their order are specified in the following table. The user must specify for each trachytype (combination of formula number and parameters) a unique positive trachytype number. This trachytype number is used in the area files (see [subsection B.17.2](#)) to indicate the roughness types within a grid cell. The following roughness formulations have been defined (see [subsection 9.11.2](#) for the related formulae):

FormulaNr	Description	Parameters
Special classes (1–50)		
1	flood protected area, reduces the effective area of the grid cell. It has no influence on the continuity equation (i.e. it does not decrease the surface area of the grid cell).	–
2	composite trachytype: fraction α of type T_1 and fraction β (generally $\beta = 1 - \alpha$) of type T_2	T_1, T_2, α, β
Area trachytype classes: simple type (51–100)		
51	constant White-Colebrook/Nikuradse value	k [m]
52	constant Chézy value	C [$m^{1/2}/s$]
53	constant Manning value	n [$s/m^{1/3}$]
54	constant z_0 value	z_0 [m]

FormulaNr	Description	Parameters
Area trachytype classes: alluvial type (101–150)		
101	simplified Van Rijn	$A [m^{0.3}]$, $B [m^{0.3}]$
102	power relation	$A [m^{1/2}/s]$, $B [-]$
103 ⁵	Van Rijn predictor	-
104 ⁵	Struiksma predictor	$A_1 [m^{1/2}/s]$, $A_2 [-]$, $\theta_c [-]$, $\theta_m [-]$, $C_{min} [m^{1/2}/s]$
105 ⁵	bedforms quadratic	-
106 ⁵	bedforms linear	-
Area trachytype classes: vegetation type (151–200)		
151	Barneveld 1	$h_v [m]$, $n [1/m]$
152	Barneveld 2	$h_v [m]$, $n [1/m]$, $C_D [-]$, $k_b [m]$
153	Baptist 1	$h_v [m]$, $n [1/m]$, $C_D [-]$, $C_b [m^{1/2}/s]$
154	Baptist 2	$h_v [m]$, $n [1/m]$, $C_D [-]$, $C_b [m^{1/2}/s]$
Linear trachytype classes: various (201–250)		
201	hedges 1	$h_v [m]$, $n [1/m]$
202	hedges 2	$h_v [m]$, $n [1/m]$
Linear trachytype classes: various (201–250)		
251	trees	$h_v [-]$, $C_D [-]$

B.17.2 Area files

The area files (for u - and v -direction) contain lines of the following format defining the surface area of the grid cell covered by a certain trachytype.

```
N      M      TrachytypeNr    AreaFraction
```

The trachytypes numbers should be defined in the trachytype definition file as described in the previous section. Records referring to the same grid cell (m,n) should be consecutive. When records referring to the same grid cell (m,n) occur at different locations within the same input file separated by records referring to other grid cells, only the last block of data is used. The AreaFraction is a value between 0.0 and 1.0. For linear trachytypes (trachytype number 201) the AreaFraction should be the projected length of the hedge in the u - and v -direction.

The notation

⁵ Formulae 103, 104, 105 and 106 use the D_{50} and D_{90} data from the sediment transport and morphology module if available. In case of a simulation without sediment, these values are obtained from the keywords `BdfD50` and optionally `BdfD90` in the mdf-file. The values of `BdfD50` and `BdfD90` may be a constant value for the whole grid, or the filename of a file containing a spatial varying field. The default value for D_{50} is 0.2 mm, and the default value for D_{90} is $1.5D_{50}$.

```
N1    M1    N2    M2    TrachytopeNr1    AreaFraction1
N1    M1    N2    M2    TrachytopeNr2    AreaFraction2
```

can be used as a shorthand notation for

```
N1    M1    TrachytopeNr1    AreaFraction1
N1    M1    TrachytopeNr2    AreaFraction2
:
N2    M2    TrachytopeNr1    AreaFraction1
N2    M2    TrachytopeNr2    AreaFraction2
```

If the trachytope definition file defines only one trachytope (not necessarily trachytope number 1), then the area files are optional: if the area files are not specified in such a case, then all points will use that one trachytope, i.e. it will be like specifying

```
1    1    NMAX    MMAX    TrachytopeNr    1.0
```

for both area files.

B.18 Creating D-Water Quality input files

The output of Delft3D-FLOW is by default stored in a file that cannot be read using D-Water Quality. Therefore, a coupling program Coup203 was required to convert the Delft3D-FLOW output files into D-Water Quality input files. This is no longer required as Delft3D-FLOW can now optionally write D-Water Quality input files directly. The following three keywords have been introduced for this purpose.

Keyword	Value	Description	Default
Flwq	3 reals	start time, time step and end time for writing data to D-Water Quality input files	–
ilAggr	integers	vertical aggregation: number of flow layers aggregated per waq layer	all ones
WaqAgg	#name.dwq#	horizontal aggregation: name of aggregation file	–

When Flwq is present, the production of WAQ input files is switched on and execution of Coup203 is not needed any more. Flwq specifies the simulation times at which WAQ input is written (start time, time interval, stop time). ilAggr specifies the vertical aggregation; WaqAgg the horizontal aggregation.

More detailed description:

Flwq	This keyword activates the writing of output to the WAQ module for the 3D domain. Flwq specifies the simulation times at which WAQ input is written: start time, time interval, stop time. The start time, time interval and stop time are specified in minutes. Start and stop time are relative to the reference date (time 00:00:00). The time interval should be a whole multiple of the FLOW time step size and produces a mass conservative internal integration of FLOW time steps to the larger WAQ-file time step size. The start and stop time should be within the FLOW simulation timings range.
------	--

ilAggr

This keyword controls the vertical aggregation. t holds a set of integer numbers. Each integer denotes the number of vertical layers to be aggregated. The sum of the integers must be equal to the total sum of hydrodynamic layers ($kmax$). Example: For example 5 WAQ layers with aggregation “1 2 2 3 2” are specified, which means that the hydrodynamic layers 2/3, 4/5, 6/7/8 and 9/10 are aggregated.

Warning:

- ◊ The number of layers to be aggregated is always (so also for the Z-layer model) defined from top to bottom.

**WaqAgg**

This keyword controls the horizontal aggregation. It may have one of the following values:

- ◊ # # (empty): No horizontal aggregation is performed and the so called #active only# value is assumed.
- ◊ #name.dwq#: The name of an aggregation file as created with D-Waq DIDO to join computational elements and/or limit the WAQ files to a subset of the FLOW area.
- ◊ #full matrix#: Indicates that the WAQ files do contain the full-matrix representation of the flow grid. Otherwise the so called #active only# value is assumed.

Example lines in MDF-file:

```
Flwq = 7.5000000e+002 1.5000000e+001 1.5000000e+003
ilAggr= 1 2 2 3 2
WaqAgg= # #
```

B.19 Dry run

An option has been added to Delft3D-FLOW to perform a simulation without actually performing all computations, a so-called *dry run*. When the dry run option is switched on, Delft3D-FLOW reads all input data, generates all the required variables in the computational engine, performs all normal checks and writes to the output files. In this way, all time steps are performed, but very little actual computations are performed. Using this option, one can verify a scenario and determine total disk usage before running an actual computation. The dry run option can be switched on by adding the keyword `DryRun` to the *Additional parameters*.

Keyword	Value	Description	Default
DryRun	#Y# or #N#	Dry run option activated	#N#

B.20 Reuse temporary files

Before the actual calculation is started, a few temporary files are created, containing time dependent information. The creation of these temporary files normally takes a few seconds. But for some models (with a very big number of open boundaries or discharges) the creation of these files may take long. Reusing them may be an option when doing several similar calculations in a row. When switched on, a message is generated in the diagnostic file. The `ReTMP` option can be switched on by adding the keyword to the *Additional parameters*.



Warning:

- ◊ It is the users responsibility to check whether reusing the temporary files is applicable.

Keyword	Value	Description	Default
<code>ReTMP</code>	#Y# or #N#	Reuse temporary files	#N#

B.21 Change the update frequency of the nodal factors

When using astronomic boundary conditions, the nodal factors are updated every 6 hours by default, see [subsubsection 4.5.6.1](#). To change this update frequency, add the keyword `NodalT` to the *Additional parameters*.

Warning: It is the users responsibility to check what the influence is of changing the update frequency on the results.

Keyword	Value	Description	Default
<code>NodalT</code>	real in $[0, \infty)$	Time interval between two updates of the astronomic nodal factors [timeUnit]	360.0 min

B.22 Bubble screen

A so called bubble screen can be modelled by Delft3D-FLOW. Bubble screens are constructions that inject air in a water system and are located at the bottom (or higher). This results into a bubble plume in which water, heat and dissolved substances are mixed and transported towards the free water surface. In practice, this bubble induced circulation can be applied to suppress unwanted phenomena such as algae growth. Bubble screens can be used in nearly stagnant and non-stratified or very strongly stratified water systems, such as lakes and reservoirs. In all other application we recommend to contact the Delft3D helpdesk for further advice whether bubble screens can be applied or not.

We distinguish between a single nozzle bubble plume (axisymmetric) and a bubble screen, i.e. a (2D) line bubble plume. For a single nozzle bubble plume we rely on the analysis and experiments of [Milgram \(1983\)](#) in a 50 m deep non-stratified reservoir as well as on the experiments of [Fannelop and Sjoen \(1980\)](#) in about 2.8 m and 9 m water depth. The injected air flux ranges from 0.024 to 0.590 m^3/s under normal atmospheric conditions. [Figure B.28](#) shows the general concept of a bubble plume created by air injected from a single nozzle. The bubble plume is about 20% narrower than the plume of entrained water ([Milgram, 1983](#)).

An example of the use of bubble screens is Lake Nieuwe Meer in The Netherlands. For many years in summer months algae blooms occurred. The so-called *Microcystis* bacterium dominated the lake in these periods. This caused a disgusting smell and other inconveniences,

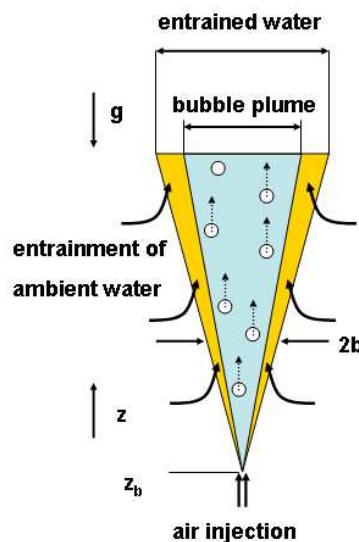


Figure B.28: Schematic axisymmetric bubble plume with entrainment of water by the rising bubbles

for example for water recreation. To suppress the algae blooms, in 1993, the Water Board Rijnland started artificial mixing of Lake Nieuwe Meer by means of air bubbling. A system of seven bubble screens, which can be seen as perforated air tubes, were put at the bottom of the lakes at different positions. This artificial mixing removed the thermal stratification and consequently the inconvenience caused by the algae.

Figure B.29 presents the definitions of the near-, mid- and far-field of the bubble screen and the associated horizontal length scales in relation to the axis of the bubble plume. The near-field and mid-field circulation due to a bubble screen is modelled in Delft3D-FLOW via the so-called subgrid approach. The far-field circulation, however, is computed by Delft3D-FLOW.

In practice, a bubble screen is applied to increase the vertical recirculation in a water system in order to break-down stratification of salinity or temperature. We note, however, that in Delft3D-FLOW a bubble screen can also be applied without any substances in the model schematisation.

Restriction:

- ◊ In Delft3D-FLOW the bubble screen functionality is only available for 3D modelling. So, it can not be applied in a 2DH (2D Horizontal) model.



B.22.1 Entrained water as function of the air injection

For a bubble screen in Delft3D-FLOW the user has to specify the water flux entrained by a bubble-plume and not the amount of air that is injected in the water system. This section presents engineering rules for both a single nozzle axisymmetric bubble plume and a (2D) line bubble plume. Conversion factors between the flux of entrained water and the amount of injected air are given. The user should take into account these conversion factors when specifying input values for bubble screens in Delft3D-FLOW (see [subsection B.22.4](#)).

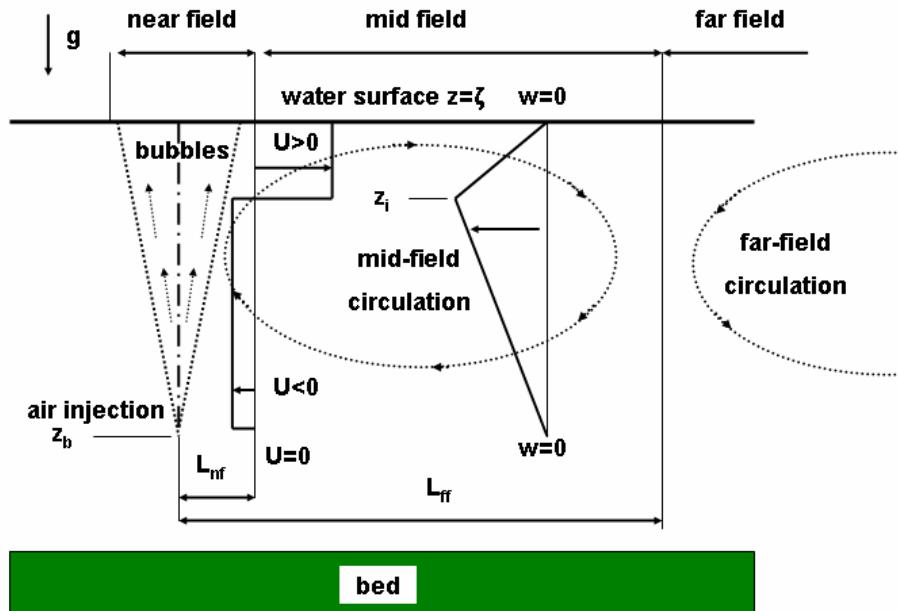


Figure B.29: Definition of near-, mid- and far-field of the circulation induced by a bubble screen and the vertical profile of the vertical (downward) velocity in the mid-field circulation cell

B.22.1.1 Single nozzle bubble plume

For a single nozzle plume we use an engineering rule for the ratio between the flux of entrained water and the amount of injected air that is based on [Figure B.30](#). In the experiments by [Milgram \(1983\)](#) was shown that the flux of entrained water Q_{ent} in [m³/s] scales with the injected air flux Q_{air} in [m³/s] to the power 0.66. This air flux is inhaled by the compressor and Q_{air} is expressed in [m³/s] under normal atmospheric conditions.

According to [Figure B.30](#) we advise, in practice for single nozzle bubble plumes, to apply a conversion factor α of 1000 between the amount of entrained water and the amount of injected air in a Delft3D-FLOW model with bubble screens, using:

$$Q_{ent} = (\alpha Q_{air})^{0.66} \quad (\text{B.85})$$

This means that the user has to multiply the amount of injected air with a conversion factor of 1000 and specify these values in the Delft3D-FLOW input file.

B.22.1.2 Bubble screen or line bubble plume

For bubble screens/line plumes, we rely on [Bulson \(1961\)](#) who instrumented and flooded an entire ship dock, located a 20 m long perforated horizontal pipe on its bed and varied the water depths from 2.8 to 11 m. In vertical and horizontal sections the vertical and horizontal velocity profiles were observed and reported by [Bulson \(1961\)](#). The injected air fluxes ranged from 0.0046 (m³/s)/m to 0.15 (m³/s)/m at the considered water depths.

Bulson defines, semi-empirical, the entrained water flux Q_{ent} by a bubble screen created by Q_{air} air flux under atmospheric conditions over a perforated tube of length L_{tube} at depth H_{inj} as

$$Q_{ent} = 0.47 H_{atm} L_{tube}^{\frac{2}{3}} (g Q_{air})^{\frac{1}{3}} \left(\frac{H}{H + H_{inj}} \right)^{\frac{1}{3}} \ln \left(\frac{H + H_{inj}}{H} \right) \quad (\text{B.86})$$

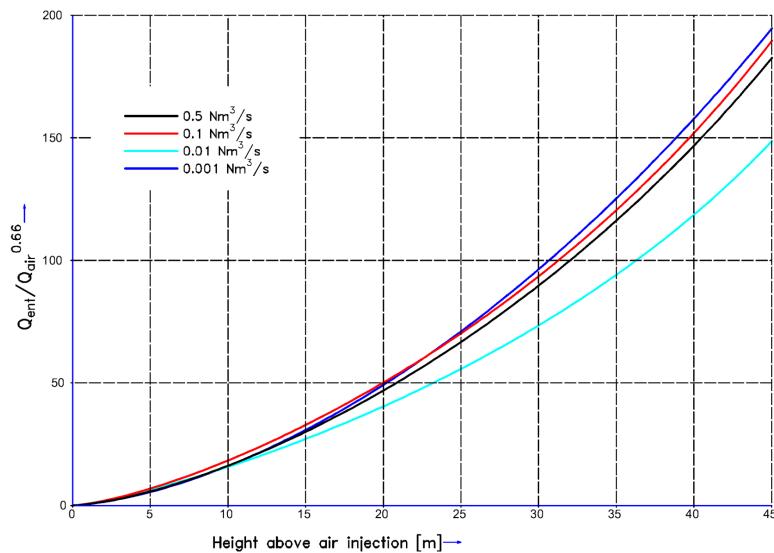


Figure B.30: Flux of entrained water as function or air flow under atmospheric conditions and height above the nozzle in stagnant non-stratified water based on experiments in (Milgram, 1983)
The injected air flux is given in $[Nm^3/s]$ where N stands for normal atmospheric conditions.

where H represents the height of a water column. [Equation B.86](#) is our second engineering rule. By applying [Equation B.86](#), it is possible to estimate the conversion factor between the amount of entrained water and the amount of injected air. As a first very rough estimate, the same conversion factor as for single nozzle bubble screens (i.e. 1000) might be used.

B.22.1.3 Bubble plume in stagnant stratified water

Depending on the air flux and stratification conditions, a bubble plume can experience one or more so-called plunge points where the entrained water flows away horizontally and above which the bubbles re-create a new plume, see [McDougall \(1978\)](#) and [Asaeda and Imberger \(1992\)](#). [Figure B.31](#) contains an illustration in which bubble plumes are applied for the de-stratification of a lake. The right graph of [Figure B.31](#) shows the typical temperature increase towards the water surface in a thermally stratified lake, where usually the hypolimnion has a uniform lower temperature. The entrainment Q_{ent} (see [Figure B.31](#)) of hypolimnion water cools the epilimnion and this process is simulated in Delft3D-FLOW.

The engineering rules of the previous paragraphs can be applied to the case of [Figure B.31](#), but the entrained water flux should be based on the typical height of the thermocline (separating hypolimnion from epilimnion) above the nozzle. This is our third engineering rule. If additional application specific assistance is required, please contact the Delft3D helpdesk.

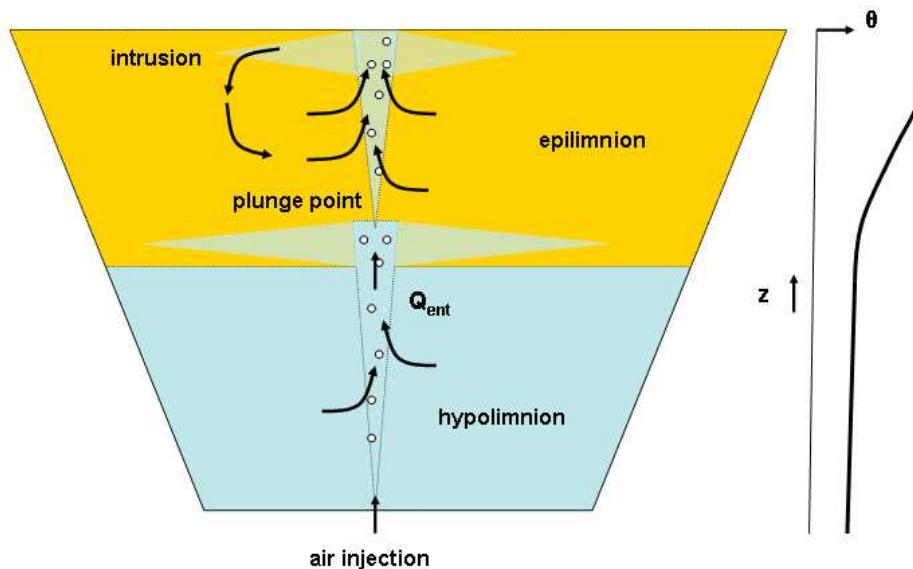


Figure B.31: Schematic overview of the introduction of cold hypolimnion water into the lower part of the warmer epilimnion by a bubble plume. Above the first plunge point the second plunge point creates an intrusion and recirculation inside the epilimnion.

B.22.2 Theory

Bubble-induced circulation patterns are often based on the following characteristics:

- 1 The water entrained into the bubble plume is diverted laterally into a horizontal flow.
- 2 The rising bubbles drag along water and transfer their potential energy to kinetic energy of the entrained water.
- 3 Upon reaching the water surface, this kinetic energy is first converted into an increase in potential energy i.e. the level of the water surface increase above the rising bubble plume.
- 4 Subsequently, this locally increased water level induces a horizontal, lateral current away from the vertical symmetry axis/plane of the bubble plume. This lateral current forms another jet that also entrains water from below.
- 5 The latter is a secondary entrainment that creates a large amplification factor compared to the injected air flow.
- 6 Experiments of 2D bubble screens show that the lateral, horizontal jet initially has a thickness that is roughly 20–25 % of the injection depth (see “ $U > 0$ ” in [Figure B.29](#)).
- 7 The conversion from kinetic energy of the rising flow into potential energy (local surface elevation) and the latter converted into horizontal flow implies that the initial momentum in the lateral jet equals the momentum of the entrained water in the rising jet.

Based on the above-mentioned characteristics, it is evident that for simulating the bubble-induced recirculation patterns the vertical water flow due to the rising bubble jet should be taken into account. Therefore, in Delft3D-FLOW we apply the approach of redistributing of sinks and sources in a certain water column for simulation bubble screens. Because of these changes in vertical flow, the horizontal flow is influenced as well.

In the remainder of this section we will describe the conceptual model for bubble screens for a 1DV (1-Dimensional Vertical) situation. The equation for the transport of heat content H is considered and reduced to its application in the 1DV model including recirculation of cooling water as well as by the recirculation by a bubble screen. The heat content H is defined by

$$H = \rho c_p(T + 273.15) \quad (\text{B.87})$$

The Reynolds-averaged version of the conservation equation of heat in an incompressible fluid reads

$$\frac{\partial H}{\partial t} + \frac{\partial uH}{\partial x} + \frac{\partial wH}{\partial z} = \frac{\partial}{\partial z} \left(D_V \frac{\partial H}{\partial z} \right) + Q \quad (\text{B.88})$$

The right-hand side in [Equation B.88](#) contains the most dominant contribution to diffusion i.e. vertical diffusion with the sum of molecular and eddy diffusivity, the latter due to mixing by turbulent fluid motions. Q represents the local or distributed heat source or sink.

For application in numerical codes, the contribution due to horizontal advection should be taken into account. Below, this is derived for the recirculation by a bubble screen or bubble plume. To that purpose, we consider [Equation B.88](#) without additional heat sources/sinks and without any diffusion terms. It then reduces to the following conservation equation

$$\frac{\partial H}{\partial t} + \frac{\partial uH}{\partial x} + \frac{\partial wH}{\partial z} = 0 \quad (\text{B.89})$$

Near and in the bubble screen the fluid moves upward. We define this fluid volume as the near-field. Its horizontal extent from the centre of the plume is defined by L_{nf} , see [Figure B.29](#). Adjacent to the near field, we define the mid-field encompassing the recirculation induced by the near-field. In this mid-field, the fluid has a downward velocity profile. This mid-field ends at some remote distance from the near field. This distance is defined as L_{ff} , see [Figure B.29](#). We remark that this represents the distance from the bubble plume to the mid-field/far-field interface. By definition, at the transition from mid-field to the far-field the velocity contributions due to recirculation vanish (i.e. $u(L_{ff}) = w(L_{ff}) = 0$). Subsequently, the far-field is affected by vertical profiles of temperature.

We assume a profile is presented with a downward velocity increasing in magnitude to some level z_i and then decreasing to zero at the level $z = z_b$ at which the bubbles are released, see also [Figure B.29](#). The question is how this recirculation of heat can be represented in the 1DV-version of [Equation B.89](#). To that purpose, we introduce the horizontal averaging operator

$$\overline{(\dots)}^{mf} = \frac{1}{L_{ff} - L_{nf}} \int_{L_{nf}}^{L_{ff}} (\dots) dx \quad (\text{B.90})$$

We define this as a mid-field averaging operator based on the transition length L_{nf} from near field to mid field as well as to the transition L_{ff} from the mid field to the far field. At the mid field to far field transition, by definition $u(x = L_{ff}) = 0$ and $w(x = L_{ff}) = 0$ hold. Now, we derive a model for the mid-field recirculation and this model aims at its application in a 1DV model or a 3D hydrostatic-pressure (shallow-water) model. The horizontal velocity in the mid-field and at some distance L from the plume's centre line follows from the mass conservation equation for incompressible fluid ([Equation B.89](#) with $H = 1$ m; take the integral $\int_{L_{ff}}^L dx$)

$$\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} = 0 \quad (\text{B.91})$$

Using the definition $u(L_{ff}) = 0$, inserting [Equation B.90](#) into [Equation B.91](#) yields

$$u(L) = (L_{ff} - L) \overline{\frac{\partial w}{\partial z}}^{mf} \quad (\text{B.92})$$

[Equation B.92](#) shows a positive horizontal velocity i.e. directed away from the bubble plume for positive vertical velocity gradient such as the profile above the level $z = z_i$. Below the level $z = z_i$ to the injection point level $z = z_b$ we have the opposite situation with a negative horizontal velocity and a negative vertical velocity gradient. [Figure B.29](#) shows an example of the vertical profile of the vertical (downward) velocity and the related horizontal velocity profile according to [Equation B.92](#).

Similarly, we consider the mid-field average of the horizontal advection of heat, which reads

$$\overline{\frac{\partial uH}{\partial x}}^{mf} = -\frac{(uH)_{L_{nf}}}{L_{ff} - L_{nf}} \quad (\text{B.93})$$

The mid-field average operator defined by [Equation B.90](#) and applied to [Equation B.89](#) while using ([Equation B.93](#)) yields

$$\overline{\frac{\partial H}{\partial t}}^{mf} + \overline{\frac{\partial wH}{\partial z}}^{mf} = \frac{(uH)_{L_{nf}}}{L_{ff} - L_{nf}} \quad (\text{B.94})$$

After substitution of [Equation B.92](#) in [Equation B.94](#), we arrive at

$$\overline{\frac{\partial H}{\partial t}}^{mf} + \overline{\frac{\partial wH}{\partial z}}^{mf} = H(L_{nf}, z, t) \overline{\frac{\partial w}{\partial z}}^{mf} \quad (\text{B.95})$$

The result [Equation B.95](#) shows that the mid-field-averaged effect of the recirculation, induced by the bubble plume, can be represented by a vertically distributed heat source/sink. [Equation B.86](#) is a significant step towards a model for the mid-field effect applicable in 1DV or 3D shallow-water codes. This source/sink distribution represents the lateral inflow or outflow of water at the near field – mid field transition, see [Figure B.29](#). The vertically distributed heat source/sink is positive above the level $z = z_i$ of maximum downward velocity, negative below this level and zero below the level $z = z_b$ of the injection of the bubbles, which is also shown in [Figure B.29](#).

The total depth-integrated heat content of the water body is not changed by the bubble plume. Consequently, the depth-integral of the right-hand side of [Equation B.95](#) yields zero. The latter constraint requires the integral of the distributed heat source above $z = z_i$ equals the sum of the distributed heat sinks below $z = z_i$, thus

$$\int_{z_i}^{\zeta} H(L_{nf}, z, t) \overline{\frac{\partial w}{\partial z}}^{mf} dz = - \int_{z_b}^{z_i} H(L_{nf}, z, t) \overline{\frac{\partial w}{\partial z}}^{mf} dz \quad (\text{B.96})$$

We note that the heat content $H(L, z, t)$ below z_i is transported toward the bubble plume i.e. flowing out of the mid-field volume $x > L_{nf}$, see also [Equation B.92](#). Therefore, the following is a reasonable approximation

$$\text{for } z < z_i : \quad H(L_{nf}, z, t) = \overline{H}^{mf} \quad (\text{B.97})$$

because to $x = L_{nf}$ the heat content from the mid field is transported passively. [Equation B.97](#) is the first model assumption about the near field action. With [Equation B.97](#) and introducing a representative far-field vertical velocity \overline{w}^{mf} , [Equation B.95](#) can be written as

$$\text{for } z < z_i : \quad \frac{\partial \overline{H}^{mf}}{\partial t} + \frac{\partial \overline{w}^{mf} \overline{H}^{mf}}{\partial z} = \overline{H}^{mf} \frac{\partial \overline{w}^{mf}}{\partial z} \quad (\text{B.98})$$

Note that [Equation B.98](#) is a non-conservative form of the advection equation

$$\text{for } z < z_i : \quad \frac{\partial \bar{H}^{mf}}{\partial t} + \bar{w}^{mf} \frac{\partial \bar{H}^{mf}}{\partial z} = 0 \quad (\text{B.99})$$

For discretisation on a vertically staggered grid, [Equation B.98](#) is preferred above [Equation B.99](#).

Using the conservation requirement [Equation B.96](#), the heat content H_b yields

$$H_b = \frac{1}{\bar{w}^{mf}(z_i)} \int_{z_b}^{z_i} H(L_{nf}, z, t) \frac{\partial \bar{w}^{mf}}{\partial z} dz \quad (\text{B.100})$$

In [Equation B.100](#), note that $\bar{w}^{mf}(z_i)$ is the maximum vertical velocity and negative but also the vertical gradient of the vertical velocity in the integral is negative below z_i .

Substitution of the constraint [Equation B.100](#) into [Equation B.95](#) yields for the level above z_i :

$$\text{for } z \geq z_i : \quad \frac{\partial \bar{H}^{mf}}{\partial t} + \frac{\partial \bar{w}^{mf} \bar{H}^{mf}}{\partial z} = H_b \frac{\partial \bar{w}^{mf}}{\partial z} \quad (\text{B.101})$$

Summarizing, the 1DV version of the equation for heat content including recirculation by a bubble plume reads

$$\frac{\partial \bar{H}^{mf}}{\partial t} + \frac{\partial \bar{w}^{mf} \bar{H}^{mf}}{\partial z} = Q \quad (\text{B.102})$$

in which the distributed source/sink Q is defined by:

$$Q = \begin{cases} \bar{H}^{mf} \frac{\partial \bar{w}^{mf}}{\partial z} & z_b < z < z_i \\ H_b \frac{\partial \bar{w}^{mf}}{\partial z} & z_i \leq z < \zeta \end{cases} \quad (\text{B.103})$$

B.22.3 Numerical implementation

In this section some implication with respect to the numerical implementation are described. [Equation B.103](#) is applied to a single horizontal grid box containing the air injection or a segment of a line injection of air. Vertically, the grid box is divided into computational layers, see [Figure B.32](#). In this figure only the six top layers are shown. In this example only these six layers are influenced by a bubble screen and not the layers that are deeper in the water column.

For a consistent and mass-conserving implementation we start by defining a vertical distribution of water-volume sources q_k with k representing the layer numbering from the air injection point at layer l to the free surface with layer number k_{surf} . The physical dimensions of q_k are in [1/s] because of the introduction of volume flux [m^3/s] per volume [m^3] of water. By definition of the circulation of the water dragged along with the rising bubble, this vertical source distribution must be zero. Otherwise, extra mass is generated, which is not physical. This condition for mass conservation can be written as

$$\sum_{k=1}^{k_{surf}} q_k \Delta z_k = 0 \quad (\text{B.104})$$

in which the discharges q_k are defined according to

$$q_k = \frac{\partial \bar{w}^{mf}}{\partial z} \quad (\text{B.105})$$

The vertical velocity then follows from vertically integrating the continuity equation and in discretised form this integral reads:

$$w_k = \sum_{k=1}^k \left\{ q_k \Delta z_k - \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \Delta z_k \right\} \quad (\text{B.106})$$

Although [Equation B.104](#) and [Equation B.106](#) suggest a reverse procedure, in the implementation in Delft3D-FLOW q_k is determined by a given vertical profile of the vertical velocity averaged over the mid-field ([Figure B.29](#)). The vertical velocity follows a profile as shown in [Figure B.32](#) and also depends on the amount of injected air. Given this q_k -distribution with the constraint [Equation B.104](#), [Equation B.106](#) yields the vertical velocity that is used in the heat transport equation.

The discrete version of constraint [Equation B.104](#), indicating that the vertically integrated heat sources are zero, reads:

$$H_b \sum_{k_i}^{k_{surf}} q_k \Delta z_k + \sum_{k=1}^{k_i-1} H_k q_k \Delta z_k = 0 \quad (\text{B.107})$$

for which in the layers with a sink holds that

$$\text{for } z < z_i : \left(\frac{\partial H}{\partial t} + \frac{\partial Hu}{\partial x} + \frac{\partial Hv}{\partial y} + \frac{\partial Hw}{\partial z} \right)_k \Delta z_k = H_k q_k \Delta z_k \quad (\text{B.108})$$

The z -independent heat content H_b , defined by [Equation B.107](#), is used in the heat transport equation for computational levels above the inversion of the vertical velocity. In other words, in the layer with a source term due to a bubble screen (with $q_k > 0$) we have

$$\text{for } z \geq z_i : \left(\frac{\partial H}{\partial t} + \frac{\partial Hu}{\partial x} + \frac{\partial Hv}{\partial y} + \frac{\partial Hw}{\partial z} \right)_k \Delta z_k = H_b q_k \Delta z_k \quad (\text{B.109})$$

Summarizing, the application of bubble screens leads to a redistribution of water masses in a water column. This is illustrated in [Figure B.32](#). From the layer at which the bubble injection starts till some distance from the free water surface, water and heat is rejected, which is modelled via a sink term. In the upper layers of the water column this water and heat is discharged into the water column via a source term. This is done in such a way that *conservation of mass* and *conservation of heat* is guaranteed.

B.22.4 Input description

This section describes how the Delft3D-FLOW input should be extended in order to simulate bubble screens.

It is emphasized that the User Interface of Delft3D-FLOW is not able to generate input data for bubble screens. Therefore, this has to be done by hand. The input format, however, is rather similar to other formats. This consists of three steps, which are now described one by one:

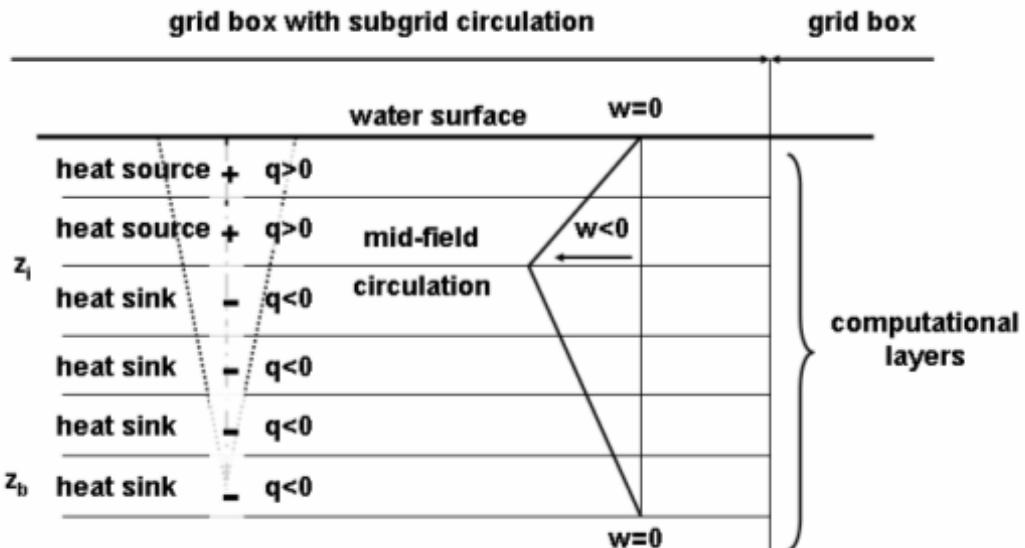


Figure B.32: The mid field averaged heat equation with vertical distribution of sources/sinks and vertical (downward) velocity profile and the model equation applied to a single grid box.

B.22.4.1 Generating a file with bubble screen locations

At first the location of the bubble screens has to be defined. This is done via an attribute file. The user is advised to use the file extension <bub>.

Keyword	Value	Description	Default
Filbub	#name.bub#	Filename for bubble screen locations	none

File contents Definition of location of bubble screens blocking the flow in M- or N-direction and the vertical position of the injection point (see below)

File format Free formatted.
Generated Manually offline.

Remarks:

- ◊ The keyword is case sensitive!
- ◊ Value MUST be enclosed between two number signs (#).



Record description:

Record	Record description
each record	Name of the bubble screen (20 characters) Begin and end indices of the bubble screen, i.e. m1, n1, m2, n2 (4 integers) Vertical position of the injection point, relative to the reference level with the orientation positive upwards (1 real).

Restriction:

- ◊ A multiple bubble screen can be defined along a line segment. The angle of the line segment and the M-direction must be an integer multiple of 45 degrees. Of course,



bubble screens along M- or N-coordinate lines are also possible.

Example:

screen north	3	8	8	3	-8.0
screen south	5	2	5	2	-12.0

B.22.4.2 Extending the discharge locations file with bubble screens

The discharge locations file (`<*.src>`) should be extended with the bubble screens. We remark that the (M, N, K) grid co-ordinates in the `<*.src>` file are not used in case of bubble screens. The grid information from the `<*.bub>` file is used instead.



Restriction:

- ◊ For the time-series only a block interpolation is allowed for bubble screens. This means that for the type of interpolation in the `<*.src>` file a “N” is compulsory.

Example:

power station P1	Y	3	3	2	p	8	8	8
screen north	N	3	8	0	b			
screen south	N	5	4	0	b			

Note that an identifier “b” is applied for bubble screens, whereas a “p” is used for inlet-outlets.

B.22.4.3 Extending the time-series file (`<*.dis>`) with amount of entrained water

The amount of entrained water due to a bubble screen has to be specified by the user. This is a time dependent quantity, which can be specified via the User Interface of Delft3D-FLOW. We remark that steps 1 and 2 (generating `<*.bub>` file and extending the `<*.src>` file) should be done manually. However, extension of the `<*.dis>` file can be done via the User Interface, because this step is comparable with defining discharge rates for “normal” discharges. We remark that in [subsubsection B.22.1.2](#) and [subsubsection B.22.1.3](#) engineering rules have been given how to estimate the amount of entrained water flux given the amount of injected air.



Remarks:

- ◊ A bubble screen can be more than one computation cell wide (see step 1). If so, the associated water flux is subdivided proportionally over all involved computational cells.
- ◊ Bubble screens are implemented using automatically generated discharges of a special type. A discharge will be generated for each computational cell containing a bubble screen, for each vertical layer above the layer containing the bubble screen. The total number of discharges is limited to 5000, including these automatically generated discharges.

In Delft3D-FLOW it is not possible to withdraw more water than available in a computation cell, but here are no limitations on the discharge in computational cells. However, large amounts of associated water flux can lead to unrealistic current profiles.

B.22.5 Coupling with other models

Deltires' hydrodynamic and water quality modules Delft3D-FLOW and D-Water Quality are able to accurately compute the effect of bubble screens on the water circulation. Bubble

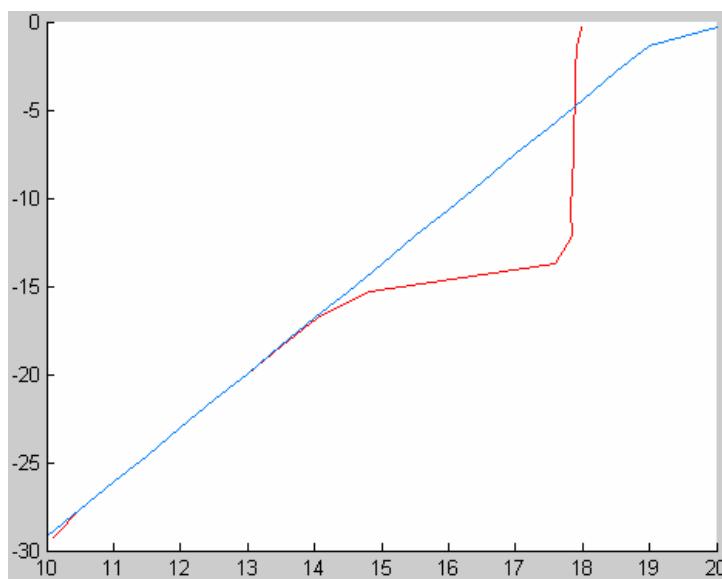


Figure B.33: Initial temperature profile (blue) and after application of a bubble screen at $z = -16 \text{ m}$ (red line). On horizontal axis the temperature (in $^{\circ}\text{C}$) and on the vertical axis the vertical position in the water column (in m).

screens are available in the operational (Main Stream Development) version of Delft3D-FLOW. In D-Water Quality this functionality is available in a special version. In particular, a special dynamic link library (dll) for waste loads is required. So, please contact the Delft3D helpdesk in case of a combined FLOW-WAQ application with bubble screens.

B.22.6 Model results

In this section the application of bubble screens in Delft3D-FLOW is illustrated via a simple test basin. A small basin of 10 by 10 grid points in horizontal direction is used without open boundaries and with a uniform depth of 30 m. Initially, we have a linearly increasing temperature profile with 20 $^{\circ}\text{C}$ at the surface and 10 $^{\circ}\text{C}$ near the bed (see blue line in Figure B.33). Without bubble screen the temperature hardly changes. In case of application of a bubble screen, with the injection point at $x = -16 \text{ m}$, from this point towards the free surface a well-mixed temperature occurs (see red line in Figure B.33). Below the injection point, the temperature does not change due to the bubble screen.

B.23 1D–3D Coupling

Online Delft3D-SOBEK coupling is a technique in which Delft3D (either 2D or 3D) models are coupled with 1D SOBEK models. This is done *online*, implying that at each time step model state information is exchanged between the models. Thus in essence, the coupled SOBEK and Delft3D models constitute to one overall model system.

By doing so, specific qualities of both the Delft3D and SOBEK models can be utilized in the overall model system. For example, using SOBEK to resolve the upstream catchment of a river system (where an adequate representation of the river network geometry is essential), while using Delft3D to model the downstream estuary in 3D (resolving stratification and complex horizontal circulation).

At present, 1D–3D coupling is possible between the hydrodynamic modules of Delft3D and SOBEK and between the water quality modules of these packages. For the hydrodynamic modules, exchanged state information consists of water levels (Delft3D-FLOW) and discharges

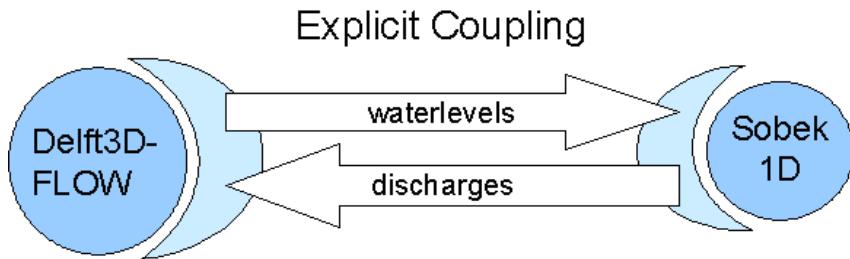


Figure B.34: Explicit exchange of water levels and discharges between Delft3D-FLOW and SOBEK.

(SOBEK).



Restriction:

- ◊ In the current implementation no constituents are exchanged (like temperature and salinity).

For the water quality modules, constituents are exchanged between the models (using the hydrodynamics from the coupled hydrodynamic model as input). The present section of the manual will focus on the online Delft3D-SOBEK coupling for the hydrodynamic modules. *Coupling of the D-Water Quality and SOBEK-WQ modules is described elsewhere.*

In the remainder of this section Delft3D-SOBEK coupling implies the *online* coupling between the hydrodynamic modules of these packages.

B.23.1 Motivation for online Delft3D-SOBEK coupling

As described above, in case of Delft3D-SOBEK coupling an overall model system is setup which consists of a 1D part in SOBEK and a 2D or 3D part in Delft3D-FLOW. Exchange of relevant state information occurs at each model time step (i.e. *online*). For specific applications, like river/estuary modelling, this method can offer a number of advantages with regard to stand-alone 1D (SOBEK) or 2D/3D (Delft3D-FLOW) modelling or *offline* coupling between these models (implying stand-alone SOBEK and Delft3D-FLOW models running sequentially).

In this case SOBEK can be used to model the upstream catchment, enabling an accurate representation of the river network geometry and the storage (important for high water waves) and also to represent hydraulic structures like flood control gates. Delft3D can then be used to model complex horizontal and vertical variability in the downstream estuary, like circulation and stratification (also taking density effects by temperature and salinity into account).

By using *online* instead of *offline* coupling, two-way exchange between these models is enabled in a mass conserving way, which is relevant for subsequent water quality applications.

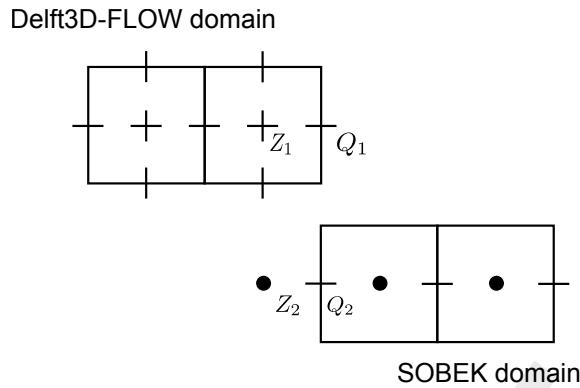


Figure B.35: Coupling of the Delft3D-FLOW and SOBEK grids.

B.23.2 Implementation of Delft3D-SOBEK coupling

The online Delft3D-FLOW and SOBEK coupling is implemented as an explicit coupling using the DelftIO library. Relevant state parameters are exchanged at time step level; SOBEK sends discharges to Delft3D-FLOW, while Delft3D-FLOW sends water levels to SOBEK (see [Figure B.34](#)). In SOBEK, these water levels are subsequently prescribed at specified boundary nodes. In Delft3D-FLOW, these discharges are prescribed at specified total discharge boundaries.

At a more fundamental level this is shown by [Figure B.35](#); both SOBEK and Delft3D-FLOW apply a so-called Arakawa-C grid, which among others implies that velocities/discharges are computed at a different location from water levels.

In [Figure B.35](#), the “round dots” in the SOBEK domain (lower part) represent nodes. In the Delft3D-FLOW domain (upper part) the “+” signs indicate water level points, whereas the “|” and “—” signs indicate velocity/discharge points on the Arakawa-C grid. Water levels and discharges are indicated by Z and Q , respectively.

Assuming that the relevant water level (SOBEK) and discharge (Delft3D-FLOW) boundaries are defined, SOBEK discharge Q_2 will be passed to Delft3D-FLOW to be imposed as boundary conditions at Q_1 . Vice versa, Delft3D-FLOW water level Z_1 will be passed to SOBEK to be imposed as boundary conditions at Z_2 .

The implementation is such that one SOBEK grid cell interacts with one boundary segment in Delft3D-FLOW. If this boundary segment consists of multiple grid cells, the SOBEK discharge is divided evenly over these grid cells. In this case the water level from the centre most grid cell along the Delft3D-FLOW boundary segment is passed to SOBEK.

Remark:

- ◊ Note that since the default discharge and water level boundaries are used, *offline* discharge (Delft3D-FLOW) and water level (SOBEK) boundary forcing has to be prescribed at these nodes/boundaries initially. These are dummy values, however, which are overruled by *online* values obtained from the other module during runtime.



The explicit implementation of the coupling poses more severe restrictions on the maximum allowable Courant-(Friedrichs-Lowy) number (CFL in [Equation B.110](#)) than would otherwise be the case for the stand-alone models:

$$CFL = \frac{\Delta t}{\Delta x} \sqrt{gH}. \quad (\text{B.110})$$

In this case, the maximum allowable Courant number differs based on the directionality of the flow; if water flows from Delft3D to SOBEK a maximum Courant number of 1 is allowed. If, however, water flows from SOBEK to Delft3D-FLOW the maximum Courant number may be somewhat larger.

If these criteria are not sufficiently met, problems will arise with regard to the accuracy of the coupling. Subsequent oscillations at the model boundaries may cause the models to crash.

With regard to the model time step, SOBEK and Delft3D-FLOW use different time integration methods, whereas SOBEK applies a 1-stage method, Delft3D-FLOW applies a 2-stage (ADI) method. Because of this difference, for a coupled Delft3D-SOBEK model the SOBEK time step should be half that of the Delft3D-FLOW model (as such coupling occurs at both stages of the Delft3D-FLOW ADI scheme).



Remark:

- ◊ Note that in SOBEK the model time step automatically can be modified during runtime based on model stability properties. If this occurs there is no communication at the intermediate time steps. So there is only communication on the user defined time steps.

Summarizing some of the important aspects of the implementation:

- ◊ The coupling occurs at **total discharge boundaries** in Delft3D-FLOW, where discharges are obtained from SOBEK reach segments. In SOBEK, the coupling occurs at **boundary nodes** where water levels are obtained from the Delft3D-FLOW discharge boundaries.
- ◊ Coupling occurs at time step level. Due to the different time integration methods for SOBEK (1-stage) and Delft3D (2-stage) **the SOBEK time step should be half that of Delft3D-FLOW** for a coupled model.
- ◊ Due to the explicit implementation of the coupling the *maximum allowable Courant number* at the coupling boundaries is 1, so $CFL \leq 1$. If this criteria is not fulfilled, problems with accuracy and stability may occur.

B.23.3 Model setup and input (including best practise)

In the present section the setup and running of a coupled Delft3D-SOBEK simulation will be outlined. This topic is subdivided into the following sub-sections:

- 1 Preparation of the individual Delft3D-FLOW and SOBEK models.
- 2 Setup of the communication file used by the coupling.
- 3 Running of the coupled model system.
- 4 Best practice with regard to running coupled Delft3D-SOBEK simulations.

B.23.3.1 Preparation of the Delft3D-FLOW and SOBEK models

Requirements posed to the Delft3D-FLOW and SOBEK models used for the coupled system follow primarily from the technical implementation of the coupling as described in [subsubsection B.23.3.3](#):

- ◊ The simulation start and end time of the SOBEK and (all) Delft3D models must be the same.
- ◊ The SOBEK *time step* has to be half that of Delft3D.
- ◊ At the Delft3D-FLOW model boundaries where online coupling with SOBEK is applied, the *Courant number* should be smaller than 1.
- ◊ The bed levels in the SOBEK and Delft3D models should coincide on both sides of the coupling boundaries. The same holds for Domain Decomposition (DD) boundaries.

- ◊ The initial water levels and velocities should be equal on both sides of the coupling boundaries.
- ◊ In Delft3D-FLOW **total discharge boundaries** have to be used to prescribe discharges obtained from SOBEK. **Dummy** forcing data has to be prescribed at these boundaries initially, to be overwritten by the coupling. This dummy forcing must have the value 0.0 during the whole computation, $Q_{t_{start}} = Q_{t_{end}} = 0.0$.
- ◊ Optionally; apply *upwind* for "Depth at grid cell faces" in Delft3D-FLOW for consistency with SOBEK (this can have a small positive effect on stability).
- ◊ Enable online coupling with SOBEK in the Delft3D-FLOW MDF-file by means of the additional keyword `Sbk01 = #delft3d_SOBEK _conf.ini#` where `<delft3d_sobek_conf.ini>` is the file describing the coupling communication (see [subsubsection B.23.3.2](#)).
- ◊ In SOBEK **boundary nodes** have to be used to prescribe water levels obtained from Delft3D-FLOW. **Dummy** forcing data has to be prescribed at these boundaries initially, to be overwritten by the coupling. This dummy forcing must have the value 0.0 during the whole computation, $Q_{t_{start}} = Q_{t_{end}} = 0.0$.
- ◊ Enable Delft3D-FLOW in the *SOBEK settings* and refer to the correct MDF and coupling communication file (see [Figure B.36](#)).
- ◊ In the SOBEK installation directory, e.g. `<C:\Sobek212\>`, in the directory `<Programs>`, one can modify the `<simulate.ini>` file. In this file there is a reference to the executable line for Delft3D-FLOW under the header:

[D3DFLOW Simulation]

The user must make sure the correct arguments are provided for running the Delftflow executable for Delft3D-FLOW. Delftflow requires three arguments:

- 1 An input file name (default: `delft3d-flow_args.txt`)
- 2 An output file name (functionality not supported yet; provide a dummy argument, default `dummy`)
- 3 The computational engine; in this case `delft3d`

With this the executable line for Delft3D-FLOW will look as follows, e.g.:

```
Task1=%D3D_HOME%\w32\flow\bin\delftflow.exe delft3d-flow_args.txt dummy delft3d
```

The `%D3D_HOME%` environment variable is set to the Delft3D installation directory, e.g. `<C:\Delft3D\>`, if you have a Delft3D installation. The Delftflow executable in the directory `<%D3D_HOME%\w32\flow\bin\>` is used for the simulation.

Remark:



- ◊ Note that only discharge and water level information is exchanged between Delft3D-FLOW and SOBEK. If either *temperature* or *salinity* are active transport processes in Delft3D, the temperature and salinity of water entering the Delft3D model from SOBEK will have to be specified in the default *offline* way in Delft3D-FLOW. As such there is no direct relation between discharge water entering the Delft3D-FLOW model (obtained online from SOBEK) and the temperature or salinity of this water (specified offline in Delft3D-FLOW). The user should keep this limitation in mind when setting up a model, for example by placing the Delft3D-SOBEK model interface upstream where no salinity intrudes and by assuming that all water which enters the system here is as such fresh.

B.23.3.2 Setup of the communication file used by coupling

To enable the coupling, the relevant Delft3D-FLOW boundaries and SOBEK nodes and reach segments are prescribed in an ASCII file containing the coupling administration, called `<delft3d_sobek_conf.ini>` (located in the Delft3D working folder). Example input from this file is shown below:

```
[General]
FileType=D3dSOBEKMapping
```

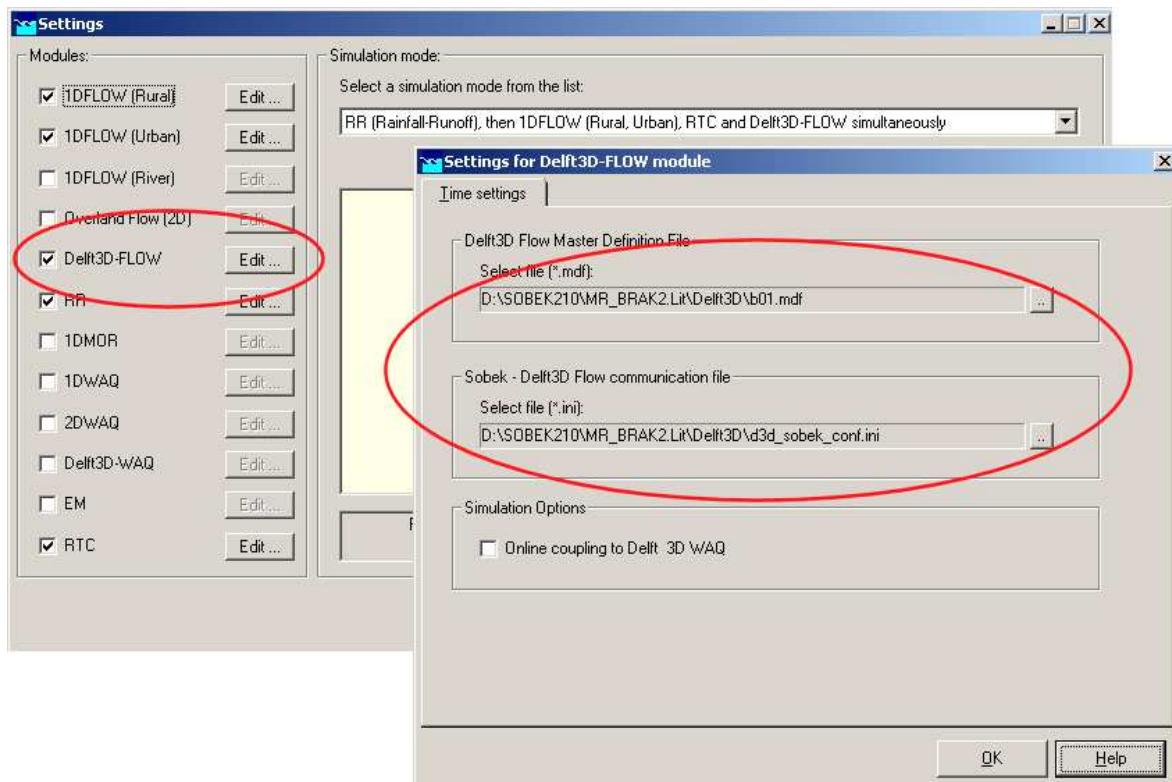


Figure B.36: Enable Delt3D-FLOW in SOBEK settings and select MDF-file and communication-file.

```
[D3dSOBEK coupling]
D3dToSOBEKWaterlevelShift=0
FakeCommunication=0
DumpMapping=-1
DumpCoordinates=0

NumberOfConnections=2

[Connection1]
SOBEKBoundaryID=SOBEKBoundary_1
SOBEKSegmentID=SOBEKReach_1
D3dOpenBoundaryID=Delft3DBoundary_1
SOBEKSegmentDirectionOutgoing=-1

[Connection2]
SOBEKBoundaryID=SOBEKBoundary_2
SOBEKSegmentID=SOBEKReach_2
D3dOpenBoundaryID=Delft3DBoundary_2
SOBEKSegmentDirectionOutgoing=1
```

Where the keywords under [General] and [D3dSOBEK coupling] indicate:

Keyword	Status	Description
[General]		
FileType	Required	
[D3dSOBEK Coupling]		

D3dToSOBEKWaterlevelShift	Optional	Offset added to water levels passed to SOBEK (allows working with different datums in SOBEK and Delft3D). Default value is 0.
FakeCommunication	Optional	If this keyword has a value other than 0 no communication will occur between Delft3D-FLOW and SOBEK.
DumpMapping	Optional	Debug option, writes file containing Delft3D and SOBEK boundary mapping applied. Default filename is <d3s-mapping.txt>.
DumpCoordinates	Optional	Debug option, if enabled the files <D3DDumpFileName> and <SOBEKDumpFileName> containing mapped data will be made.
NumberOfConnections	Required	Number of coupling boundaries.

The keywords under [Connection#] indicate:

Keyword	Status	Description
[Connection#]		
SOBEKBoundaryID	Required	Name of SOBEK boundary node used for coupling.
SOBEKSegmentID	Required	Name of (active) SOBEK segment used for coupling.
D3dOpenBoundaryID	Required	Name of Delft3D boundary used for coupling.
SOBEKSegmentDirectionOutgoing	Optional	Direction of SOBEK schematisation at coupling boundary (-1 if directionality is into Delft3D grid, 1 if directionality is into the SOBEK grid). Default value of 1 if keyword is omitted.

A [Connection#] field has to be specified for each of the coupled boundaries (two boundaries included in example).

B.23.3.3 Running of the coupled model system

The coupled Delft3D-SOBEK model system is run from the SOBEK Case Manager. This implies that the model runs only on a MS Windows platform. If Delft3D-FLOW is enabled as an option in the SOBEK settings, SOBEK will launch both the SOBEK and the Delft3D-FLOW executables.

The path of the Delft3D-FLOW executable used is specified in the SOBEK <simulate.ini> file and by default is set to the %D3D_HOME% path as specified in the Windows environment variables.



Remark:

- ◊ Note that simulations involving Delft3D-FLOW may take considerably longer than stand-alone SOBEK simulations. As such it is practical to run these simulations on dedicated PC's.

B.23.3.4 Best practice with regard to running coupled Delft3D-SOBEK simulations

Running a coupled Delft3D-SOBEK model is no trivial exercise, given the fact that experience with both systems (SOBEK and Delft3D) is required. Moreover, since the systems run in parallel and are interdependent this experience is preferably combined in the person running the simulations. In addition, by using multiple models the overall model is significantly more complex than the individual building blocks. In practise, the following steps have proven to be helpful when running the model:

- ◊ Prior to developing the model, discuss with both the Delft3D and SOBEK modelling parties how and where the coupling will be made and possible consequences this may have on the model schematisations.
- ◊ When developing the Delft3D model, take into account from start that the maximum Courant number at the coupled boundaries should be smaller than 1. This will have consequences on the model time step and grid size.
- ◊ Give logical names to the SOBEK reach segments and nodes and the Delft3D-FLOW boundaries used for coupling (for example SOBEKNode_1, SOBEKSegment_1 and Delft3DBoundary_1). If the number of boundaries increases this will make administration more manageable.
- ◊ Be aware that if the SOBEK network is changed at a later stage reach ID's might change, which could effect the coupling administration.
- ◊ In Delft3D-FLOW, add cross-sections on your boundaries and observation points next to them in order to validate results between Delft3D-FLOW and SOBEK and to ensure that the coupling is working satisfactory.
- ◊ Run the coupled simulations on dedicated PC's, given the fact that simulations involving Delft3D-FLOW may take considerably longer than stand-alone SOBEK simulations.

B.23.4 Versions and limitations

The Delft3D-SOBEK coupling was originally implemented for the coupled Pearl River Estuary (PRE) model (project in period 2005–2008). Developments during that project were consolidated in Delft3D-FLOW version 3.60.01.8345 and SOBEK version 2.12.

The method was subsequently used for the Marina Reservoir project (project in period 2006–...) during which some modifications were made to the implementation.

At the moment the Delft3D-SOBEK coupling has been tested on Windows XP (32-bit and 64-bit) in the following setups:

- ◊ Delft3D-FLOW + SOBEK + SOBEK-RR + SOBEK-RTC
- ◊ Delft3D-FLOW + SOBEK + D-Water Quality + SOBEK-WQ

When using the coupling with other Delft3D or SOBEK modules or on other platforms, the user is advised to be cautious that these might not work properly or are at least untested.

B.24 Output of Courant number messages

After every computation of the advective terms in the momentum equations, Delft3D-FLOW computes the largest local Courant number that has occurred. If this Courant number is larger than 1.0, a warning will be written to the <tridiag> file. A suggestion for the time step to avoid too large Courant numbers is included in the warning. To make sure that for long simulations, the <tridiag> file does not get too large, the number of warnings is limited to 100. Using the keyword `CflMsg`, the user can enable the writing of these warnings also after the 100th message. This can be of use to find the cause of possible instabilities. In the *Additional parameters* group, you should then add the line:

```
CflMsg=#Y#
```

The default setting is `CflMsg=#N#`, resulting the writing of CFL-messages to the <tridiag> file to be truncated after 100 warnings.

B.25 Initialisation of water depth in dry points

To avoid possible difficulties in the flooding of dry areas in the Z-model of Delft3D-FLOW all dry cells are initiated by default with a water depth of 10 % of the wetting and drying threshold `Dryf1c`. This is also equal to the minimum layer thickness in the Z-model. Using the keyword `DepIni`, the user can modify this initial water depth for initially 'dry' cells. E.g. for simulations where the flooding of completely dry areas is to be simulated (*dry dambreak problem*), the initial depth in dry cells can be set to zero by specifying in the *Additional parameters* group:

```
DepIni=0.0
```

Any other value for `DepIni` is allowed, but one should take care not to introduce any unwanted effects in a simulation, such as downhill flow from higher-lying 'dry' areas. This will occur when `DepIni` is chosen larger than the flooding threshold `Dryf1c`.

The value for `DepIni` is in meters. It concerns a total water depth, not a water level.

B.26 Remapping of near-bottom layers for accurate and smooth bottom shear stress in Z-layers

In the Delft3D-FLOW Z-layer model, the partial cell approach is applied to approximate the bottom and free-surface positions with greater accuracy than with a full cell approach (see e.g. [Adcroft et al. \(1997\)](#); [Pacanowski and Gnanadesikan \(1998\)](#)). Due to the partial cell approach, thin cells can exist near the bottom and free-surface. These thin cells are known to introduce inaccuracies and discontinuities in bottom shear stress, velocity profiles and even water levels. [Platzek et al. \(2012\)](#) showed that these inaccuracies and discontinuities can be reduced by performing a near-bottom layer remapping as shown in [Figure B.37](#) for a schematic 2D vertical layer distribution. The remapping is done in both x - and y -direction.

One should note that this remapping, causes horizontally-neighbouring cells to be vertically shifted with respect to each other. This can have some effect on the horizontal transport through advection and diffusion, and may introduce some spurious mixing. This mixing will however commonly be less than the mixing due to the discontinuities that would exist without the remapping.

When the transport of constituents (salinity, temperature, etc.) is involved, the concentrations of the constituents in the cells that are remapped are modified to maintain mass conservation.

This remapping to obtain more accurate and smooth bottom shear stress and velocity profiles can be switched on by specifying in the *Additional parameters* group:

```
Ztbml=#Y#
```

The keyword `Ztbml` stands for Z-model Taubot (τ_b) Modified Layering.

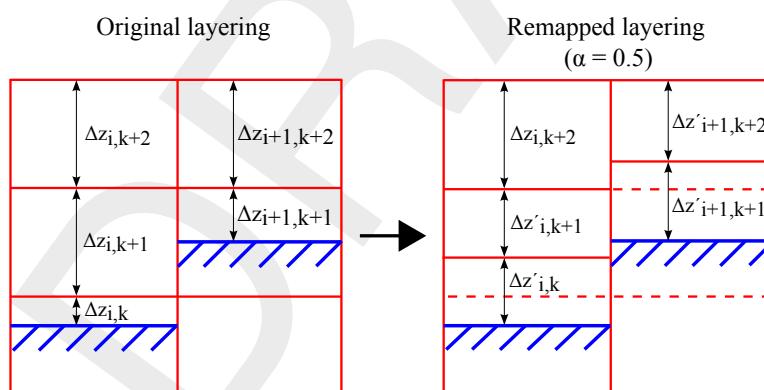


Figure B.37: Remapping of two near-bed layers to an equidistant layering. Figure from [Platzek et al. \(2012\)](#).

B.27 Slope Limiter

A slope limiter was implemented in Delft3D-FLOW to avoid the computation of high velocities along very steep bottom slopes. This e.g. occurs in the case of precipitation on steep banks, i.e. dry cells that lie much higher than the adjacent wet cells. Once the dry cells become wet (after sufficient precipitation), the cells will discharge into the neighbouring (lower-lying) active cells. It was found that very large velocities could occur in such situations, if the water level gradient is not limited for such situations. Using the additional parameter:

```
SlpLim=#Y#,
```

this functionality can be switched on. In the momentum equations the water level gradient is then explicitly corrected using a pressure gradient involving the bottom level of the high-lying cell. This is realised using the following algorithm (in pseudo-code):

```
if (slplim)
    do nm = 1, nmmax
        dpsmax = max (-dps (nm) , -dps (nm+1) )
        if (s0 (nm) < dpsmax) then
            do k = 1, kmax
                dpdx (nm, k) = dpdx (nm, k) - g* (s0 (nm) -dpsmax) /dxu (nm)
            enddo
        elseif (s0 (nmu) < dpsmax) then
            do k = 1, kmax
                dpdx (nm, k) = dpdx (nm, k) + g* (s0 (nmu) -dpsmax) /dxu (nm)
            enddo
        endif
    enddo
endif
```

where nm is the cell index counter, k is the layer index counter, $dpdx$ is the pressure-gradient correction, dps is the bottom depth, $s0$ is the water level, g is the gravitational acceleration and dxu is the horizontal grid size in a u -velocity point. In the y -direction, a similar correction is performed.

B.28 Real-time control

The real-time control (abbreviated to RTC) module allows you to vary the bottom level of a barrier gate during a simulation. This feature should be activated in the Delft3D-FLOW input file using the keyword `Rtcmod`. The RTC module can be used to prescribe a time function for a quantity (e.g. overrule a constant barrier height by a time varying barrier height following a user defined time function) or to change a quantity by means control rules based on simulation results. Switch on the RTC mode using `Rtcmod` and use the `Filbcb` keyword in case you want to merely overrule a quantity using a prescribed time function. To control a quantity based on simulation results use the `Filrtc` keyword and specify the control rules using the `<D3D_Loc.Rtc>`, `<DECISPAR.RTC>`, and `<D3D_MEAS.RTC>` input files of the RTC module.

Keyword	Value	Description	Default
Rtcmod	#Y# or #N#	Switch indicating whether Delft3D-FLOW needs to wait for data from RTC module	#N#
Filbcb	#filename#	Name of the forcings file containing time functions for overruled quantities. See Section B.28.2 for a description of the file format.	none
Filrtc	#filename#	Name of the file containing locations for which simulation results will be used to control the quantities. The format of this file is identical to that of the observation points file (see A.2.26).	none

The contents of the <D3D_Loc.Rtc>, <DECISPAR.RTC>, and <D3D_MEAS.RTC> files are described in the Sections [B.28.3](#), [B.28.4](#) and [B.28.5](#) respectively. Please note that on case-sensitive platforms (such as Linux) the names of these files need to use the same capitalization as listed here.

B.28.1 Run procedure

A Delft3D-FLOW that includes coupling to RTC cannot be started by means of the Delt3D menu. For these simulations a batch file or script must be used. The following instructions assume that you are working from a standard Delft3D installation on Windows. The environment variable %D3D_HOME% points to the root folder of the Delft3D installation. The environment variable %ARCH% indicates the platform configuration, e.g. win32 or win64.

To start the simulation first start Delft3D-FLOW as

```
%D3D_HOME%\%ARCH%\flow2d3d\bin\d_hydro.exe config_flow2d3d.xml
```

where the <config_flow2d3d.xml> is as created by the Delft3D-MENU. Subsequently, you have to wait a few moments for the <TMP_SYNC.RUN> to appear. Then you can start the RTC program as

```
%D3D_HOME%\%ARCH%\rtc\bin\rtc.exe %D3D_HOME%\%ARCH%\rtc\RTC.FNM rtc rtn
```

Please make sure that the run folder contains a <dioconfig.ini> file as copied from the %D3D_HOME%\%ARCH%\flow2d3d\default folder.

B.28.2 Time-series forcing of controlled structures

File contents	Time-series for forcing conditions of RTC controlled structures.
Filetype	ASCII
File format	Fix format for header information; free format for time-series data.
Filename	<name.bcb>
Generated	manually offline

Record description:

If the forcing of an RTC controlled hydraulic structure are known in advance, this file can be used to specify the forcing time series. RTC can currently be used to control

- ◊ the elevation of the bottom of a barrier gate
- ◊ the flow rate of a discharge

Since the flow rate of a discharge can already be forced via the flow rate and concentrations at discharges file (see Section A.2.22), the current file will typically be used for forcing the elevation of the barrier gate. In the following, please read this quantity when we write about the RTC controlled structure parameter.

The format of this file is identical to that of the time-series flow boundary conditions file, see Section A.2.15. The table-name and contents lines are not used. The location line needs to refer to the name of the hydraulic structure (barrier or discharge) to be controlled.

For each RTC controlled structure parameter the data is given in two related blocks:

- 1 A header block containing a number of compulsory and optional keywords accompanied by their values.
- 2 A data block containing the time dependent data.

Description header block:

Record	Text	Format	Value	Format
1	location	a20, 1x	'name hydraulic structure' (see <name.bar> and <name.src> files as described in Sections B.3.2.1 and A.2.21 respectively)	a
2	time-function	a20, 1x	'non-equidistant'	a
3	reference-time	a20, 1x	yyyymmdd (must be equal to it-date)	i8
4	time-unit	a20, 1x	'minutes'	a
5	interpolation	a20, 1x	'linear'	a
6	parameter	a20, 1x	'time', unit '[min]'	a, 'unit', a
7	parameter	a20, 1x	'parameter name', unit '[par unit]'	a, 'unit', a
8	records-in-table	a20, 1x	number of records in the data block	integer

Description data block:

Record	Record description
each record	Time in minutes after the Reference Date and value representing the forcing parameter for which a time varying condition is specified (both reals).

**Remarks:**

- ◊ The parameter name in record 7 of the header block is not checked but is assumed to be equal to 'barrier height' with unit 'm', or 'flow rate' with unit 'm*3/s' for barrier gate height and discharge flow rate respectively.
- ◊ The sequence of header/data block pairs is independent of the order of specification of barriers and discharges in the <name.bar> and <name.src> files.
- ◊ If any barrier or discharge is not specified, the constant barrier height specified in the <name.bar> file or the flow rate specified in the <name.dis> file will be used, unless an RTC control rule is specified.

**Restrictions:**

- ◊ Maximum record length is 5000.
- ◊ Position, format of the keywords and the format of keyword-values in the header are fixed (see example).
- ◊ All keywords have a length of 20 characters.
- ◊ Header in each block must be ended with the (compulsory) keyword: 'records-in-table' accompanied by the number of data records to follow.
- ◊ Times must be multiples of the integration time step; the times specified will be checked on their domain (i.e. data must be specified for the whole simulation period).

Example:

Model with 2 barriers named 'Barrier 1' and 'Barrier 2' and 3 discharges named 'River 1', 'River 2', and 'Intake-Outfall'. The <name.bcb> is used to overrule the constant barrier height of 'Barrier 2' and the flow rates of discharges 'River 1' and 'Intake-Outfall'. The gate of barrier 2 rises from 5 m above the reference plane at the beginning of the simulation to 7.5 m after 8000 minutes. The inflow flow rate at the discharge point called 'River 1' rises linearly from 0 to 150 m³/s over the same period. In the meantime the flow rate across the Intake-Outfall link will increase from 0 to 50 m³/s during the first half of the period, and remain constant afterwards. The final block for the location 'Unused' will not be used in the simulation since there is neither a barrier nor a discharge with the name 'Unused', but the time points associated with this data set will be checked in the same manner as the time points of the data sets actually used. Note that the order of specification isn't important.

```

location          ' Intake-Outfall      '
time-function    'non-equidistant   '
reference-time   19941001
time-unit        'minutes'
interpolation    'linear'
parameter        'time'      unit '[min]'
parameter        'flow rate' unit '[m*3/s]'
records in table 3
      0.0000  0.000
     4000.0000 50.000
     8000.0000 50.000
location          'Unused           '
time-function    'non-equidistant   '
reference-time   19941001
time-unit        'minutes'
interpolation    'linear'

```

```

parameter          'time'           unit '[min]'
parameter          'barrier height' unit '[m]'
records in table 2
    0.0000   -1.000
    8000.0000 -2.000
location          'Barrier 2'      '
time-function     'non-equidistant'
reference-time    19941001
time-unit         'minutes'
interpolation     'linear'
parameter          'time'           unit '[min]'
parameter          'barrier height' unit '[m]'
records in table 2
    0.0000   5.0
    8000.0000 7.5
location          'River 1'        '
time-function     'non-equidistant'
reference-time    19941001
time-unit         'minutes'
interpolation     'linear'
parameter          'time'           unit '[min]'
parameter          'flow rate'      unit '[m^3/s]'
records in table 2
    0.0000   0.00
    8000.0000 1500.00

```

B.28.3 Data Locations layer

The <D3D_Loc.Rtc> file lists the Delft3D-FLOW locations which can be used in the control rules to be specified in the decision layer (Section B.28.4). These names should match one of the following strings:

- ◊ name of a location listed in the file referred to by the Filrtc followed by an underscore and the layer number (always 1 in a 2D model). For these locations and layers Delft3D-FLOW passes the concentrations of the constituents (including 'Salinity' and 'Temperature' as appropriate), the layer 'Elevation', and the 'Water level'.
- ◊ name of a structure (barrier, discharge) to be controlled by RTC For these locations Delft3D-FLOW passes the 'Elevation' (in case of barrier) or 'Flow Rate' (in case of discharge).

Delft3D-FLOW may pass more information to the RTC module, but only data of the selected locations will be used in the decision rules.

Example:

This file indicates to RTC that it can expect from Delft3D-FLOW data for the locations 'ObsPoint1' and 'ObsPoint2' where it will use data layer 1 in both cases, as well as structure state information for 'Barrier 2' and 'River 1'.

```

D3DO id 'ObsPoint1_1' nm '' d3do
D3DO id 'ObsPoint2_1' nm '' d3do
D3DO id 'Barrier 1' nm '' d3do
D3DO id 'River 1' nm '' d3do

```

B.28.4 Decision layer

The <DECISION.RTC> file specifies the control logic which in the end determines the new state (height or flow rate) of structure (barrier or discharge). For the general Delft3D-FLOW locations listed in the <D3D_Loc.Rtc> file (see previous section), the control logic rules

can use the following variables: the concentrations of the constituents (indicated by their Delft3D-FLOW names: 'Salinity', 'Temperature', ...), the elevation of layer centre (indicated by 'Elevation'), and the local water level (indicated by 'Water level'). For the barriers listed in the <D3D_Loc.Rtc> file, the control rules can use the variable 'Elevation' of the gate. For the discharges listed in the file, the rules can use the variable 'Flow Rate'.

Example:

A data line in the decision rules file which uses the salinity as computed by Delft3D-FLOW at the location ObsPoint1 in layer 1 reads:

```
DATA ty 'D3DLoc' lo 'ObsPoint1_1' va 'Salinity' ca 1 cb 0 cn 0 data
```

where the meaning of the keywords `ca`, `cb` and `cn` are given in the general description of the decision file below.

The decision parameter file contains the definition of the decision parameters. There are four possible ways of defining decision parameters:

- ◊ A first possible definition of a decision parameter is a linear combination of Water Flow results, Rainfall-Runoff results, predicted rainfall or other external (wind) data, on-line water quality data, and other decision parameters.
- ◊ A second option is to define decision parameters as a non-linear function of other decision parameters.
- ◊ A third option is to define decision parameters as SOBEK tables.
- ◊ The fourth option is to define SOBEK reservoirs.

Furthermore, there are some standard predefined decision parameters which can be used to define other decision parameters or measures. These predefined decision parameters are:

- ◊ Year
- ◊ Month
- ◊ Day
- ◊ Hour (an integer between 0 and 23)
- ◊ Minute (an integer between 0 and 59)
- ◊ Second (an integer between 0 and 59)
- ◊ Date (a number defined as $10000 * \text{year} + 100 * \text{month} + \text{day}$)
- ◊ Time (a number defined as $10000 * \text{hour} + 100 * \text{minute} + \text{second}$)
- ◊ Date_Time (the number defined as date + time/10000)
- ◊ Day of Week
0=Sunday;
1=Monday;
..
6=Saturday;

DECISPAR.RTC-file (Decision layer)

Function Decision Parameters

The value of decision parameters is based on data locations and also may be dependent of other decision parameters. These decision parameters are specified in the PAR2 records. In the User Interface these records are presented under Decision Parameters as Function Parameters.

These data records may look like:

PAR2 id 'beslispar_1' nm 'para_1' iv 1 do 'None'

DATA ty 'ExtLoc' lo 'Extern_id' va 1 ca 0.1 cb -1 cn 0 data par2

or

PAR2 id 'beslispar_2' nm 'para_2' do 'Interpolate'

DATA ty '2DFlowLoc' lo '2D_Weurt' va 1 ca 1 cb 0 cn 0 data

DATA ty 'InterpolationTable' lo 'InpTable_1' va 1 ca 0 cb 0 cn 0 data par2

or

PAR2 id 'beslispar_3' nm 'para_3' do 'multiply'

DATA ty 'FlowLoc' lo 'Sobek-id1' va 1 ca 1. cb 0. cn 0 data

DATA ty 'RRLoc' lo '3b-stuw' va 3 ca 1. cb 0. cn 0 data

DATA ty 'PrecipLoc' lo 'Neerslag-id' va 1 ca 1. cb 0. cn 0 data

DATA ty 'RRLoc' lo '3B-ow' va 1 ca 0.5 cb 0. cn 0 data

DATA ty 'ParLoc' lo 'beslispar_1' va 1 ca 0 cb -1 cn -1 data

DATA ty 'ExtLoc' lo 'Extern_id' va 1 ca 0.1 cb -1 cn 0 data

DATA ty 'FlowLoc' lo 'Sobek-id2' va 2 ca. 2. cb. 1. cn -1 data par2 where:

- id id of decision parameter
- nm name of decision parameter
- iv initial value to be used at beginning of simulation when parameter is used in computations with negative time shift; the default initial value is 0 for all parameters
- do mathematical actions or functions on location value[s]

The following functions (or mathematical actions) can be defined in PAR2 records:

Functions that can be applied using one argument only:

- ◊ Arccosine (result in degrees)
- ◊ Arccosine (result in radians)
- ◊ Arcsine (result in degrees)
- ◊ Arcsine (argument in radians)
- ◊ Arctangent (result in degrees)
- ◊ Arctangent (result in radians)
- ◊ Ceiling (smallest integer $\geq x$)
- ◊ Cosine (input argument in degrees)
- ◊ Cosine (argument in radians)
- ◊ Exponent
- ◊ Floor (largest integer $\leq x$)
- ◊ Hyperbolic cosine

- ◊ Hyperbolic sine
- ◊ Hyperbolic tangens
- ◊ Logarithm with base 10
- ◊ Natural logarithm
- ◊ Nearest integer
- ◊ None (no function is applied)
- ◊ Sine (input argument in degrees)
- ◊ Sine (argument in radians)
- ◊ Square
- ◊ Square root
- ◊ Tangent (input argument in degrees)
- ◊ Tangent (argument in radians)

Functions that can be applied for two arguments only:

- ◊ Interpolate (determine values using an Interpolation table)

Functions that can be applied for two or more arguments:

- ◊ Add
- ◊ Average
- ◊ Divide
- ◊ Max
- ◊ Min
- ◊ Multiply
- ◊ Power
- ◊ Subtract

The PAR2 record contains DATA sub-records with data for the specified locations. The data items will appear in the same order as specified in the PAR2-record, so the order may influence the results, depending on the specified action function.

The DATA sub-record contains the following keywords:

ty	Type of location, which can be: 'FlowLoc' = Water Flow Location in 1D schematization 'RRLoc' = Rainfall-Runoff Location 'ExtLoc' = External data from file 'PrecipLoc' = Precipitation Prediction Location 'DateTimLoc' = Date Time 'ParLoc' = (Other) Decision Parameter 'WQLoc' = Sobek-WQ Monitoring Location '2DFlowLoc' = Water Flow Location in 2D schematization 'InterpolationTable' = Interpolation Table
lo	Location-id (defined in RTC on the "Data Location" Tab)
va	Selected variable, which has different values for the different data types: Water Flow Location in 1D schematization: Node: 1 = Water Level 4 = Water Depth Reach Segment: 2 = Discharge Structure: 5 = Crest Level 6 = Crest Width 7 = Gate Lower Level 8 = Opening Height 9 = Structure Flow Area

10 = Discharge Structure
11 = Structure Velocity
12 = Water Level Up
13 = Water Level Down
14 = Head
15 = Pressure Difference
16 = Pump Capacity
Measurement Location:
1D Flow Measurement Station:
1 = Water level
2 = Discharge
3 = Storage Surface Area
4 = Water Depth
Manhole Measurement Station:
1 = Water level
3 = Storage Surface Area
4 = Water Depth
Pipe Measurement Station:
2 = Discharge
Reach Location:
1 = Water level
2 = Discharge
3 = Storage Surface Area
4 = Water Depth
Water Flow Location in 2D schematization:
2D History Station & 2D Breaking - Dam:
1 = Water level
4 = Water depth
17 = Bed level
18 = U-velocity
19 = V-velocity
20 = C-Abs. flow velocity [= $\sqrt{U^2 + V^2}$]
Rainfall-Runoff Location
1 = Open Water Level
2 = Groundwater Level
External data
From HIS-file
1 = Value
In case of Public Wind
1 = Wind Direction
2 = Wind Velocity
Precipitation Prediction Location
1 . . . n = Prediction period in number of time steps
Decision Parameter
1 = Parameter Value
Date Time
1 = Actual Value, that can be
Integer value of:
◊ Year(yyyy),
◊ Month(mm),
◊ Day(dd),
◊ Hour(hh),
◊ Minute(mm),

- ◊ Second(ss),
- ◊ Date(yyyymmdd),
- ◊ Time(hmmss),
- ◊ Day_of_week(day no)

Real value of

- ◊ Date+Time(yyyymmdd.hhmmss)
- ◊ CompTimeStep (Timestep as defined in RTC settings)

Sobek-WQ Monitoring Location

SOBEK WQ variable (series id as character string)

ca coefficient for multiplication

cb constant for addition

cn time shift: integer values

1 represents: t+1

0 represents: t

-1 represents: t-1

-2 represents: t-2

etc.

When using negative time shifts, the value used for the first time steps will be equal to the initial value of that parameter (see the iv keyword).

The value of the decision parameter is determined by executing the specified action over the calculated values of all locations.

The time indexes are t , $t - 1$, $t - 2$. At the moment the time shift is limited to at most 720 time steps (based on 5 days memory of a computation time step of 10 minutes). This can be specified in the INI file (keyword DecisionHorizon).

By using different time indices, a moving average or a trend can be determined. For instance, by defining as a decision parameter: the water level at time step t minus the water level in time step $t - 1$, in fact the increase in water level will be determined. If the water level rises quickly, you may want to take some measure. Real-time Control will remember the data values of all data locations of the previous time steps; the communication with the other modules is such that it only receives the values for the current time steps.

When you are using SOBEK-RTC in combination with WQ, you could define the decrease in O₂ concentration, or the (increase) in fraction of WWTP water in the system as decision parameters, and define measures to set structure set points in the SOBEK Water Flow modules, based on these (water quality based) decision parameters.

The coefficients for multiplication and addition can be positive, negative or zero. This gives many possibilities in the definition of decision parameters. Even the definition of a constant decision parameter is possible (specify zero for the multiplication value, and set the proper addition value)

The time shifts can be zero or negative, meaning that the decision parameter is depending on current or previous values of variables at data locations. If a decision parameter is depending on other decision parameters, it should depend on values of previous time steps (since the values for the current time step may not yet be defined). In that case only negative values for the time shift are allowed.

PAR2 records may refer to decision parameter values for the current timesteps or for previous timesteps. When defining decision parameters depending on values of other decision

parameters in the same timesteps, dependency loops should be avoided.

The first set of functions can accommodate multiple arguments; in case of 1 argument (only 1 input decision parameter), the result is equal to that decision parameter. In case of 2 input decision parameters a and b, the result is obvious. In case of 3 input decision parameters a,b, and c, the result for the maximum operation will be: $\max(\max(a,b),c)$ which is the same as $\max(a,b,c)$; the result of the power operation will be $((a^{**}b)^{**}c)$.

The result of the computations is as the name suggests; using the short notation d_1 for decision parameter 1, being equal to $di_1 * dc_1 + da_1$, the results are:

```

multiply: d1*d2
divide : d1 / d2
add: d1 + d2
subtract: d1 - d2
max: max (d1, d2)
min: min (d1, d2)
average: average (d1, d2) = add (d1, d2) / 2
power: power(d1,d2) = d1 ** d2

```

The generalisation to more than 2 arguments is obvious, e.g.

```

subtract (d1, d2, d3, d4) = d1- d2 - d3 - d4
divide (d1, d2, d3, d4) = ((( d1 / d2 ) / d3 ) / d4 )
average (d1, d2, d3) = add (d1, d2, d3) / 3 = (d1+d2+d3) / 3

```

Some example records:

```

PAR2 id 'PreviousLvlShimen' nm " do 'none'
DATA ty 'FlowLoc' lo 'Lvl Lower Shimen' va 1 ca 1 cb 0 cn -1 data par2
PAR2 id 'ExpectedRain402001_7' nm " do 'add'
DATA ty 'PrecipLoc' lo '402001_7' va 1 ca 0.166 cb 0 cn 0 data
DATA ty 'PrecipLoc' lo '402001_7' va 2 ca 0.166 cb 0 cn 0 data
DATA ty 'PrecipLoc' lo '402001_7' va 3 ca 0.166 cb 0 cn 0 data
DATA ty 'PrecipLoc' lo '402001_7' va 4 ca 0.166 cb 0 cn 0 data
DATA ty 'PrecipLoc' lo '402001_7' va 5 ca 0.166 cb 0 cn 0 data
DATA ty 'PrecipLoc' lo '402001_7' va 6 ca 0.166 cb 0 cn 0 data par2
PAR2 id 'ExpectedInflowShimen' nm " do 'max'
DATA ty 'ParLoc' lo 'PreviousInflowShimen' va 1 ca 1 cb 0 cn 0 data
DATA ty 'ParLoc' lo 'ExpectedRunoffShimen' va 1 ca 0.5 cb 0 cn 0 data par2
PAR2 id 'Downstream demands Shimen' nm " do 'add'
DATA ty 'ParLoc' lo 'DemandTaoYuan' va 1 ca 1 cb 0 cn 0 data
DATA ty 'ParLoc' lo 'PWS Shimen' va 1 ca 1 cb 0 cn 0 data
DATA ty 'ParLoc' lo 'Irr Demand Shimen' va 1 ca 1 cb 0 cn 0 data par2

```

As the example indicates, records may be more than one line long. The beginning of a record is indicated by PAR2 in capitals (upper case), and the end of a record is indicated by par2 in lower case.

Time Decision Parameters

A third way for defining decision parameters is by defining SOBEK time-tables. In this way, a decision parameter is simply defined as a standard SOBEK time-table. SOBEK-RTC allows tables for one decision parameter at a time only. The table can contain interpolation and

periodicity switches.

The tables are defined using the PAR3 record.

PAR3 id 'MaxFlowOutlet2TimeTable' PDIN 1 1 365;00:00:00 pdin

TBLE

```
'1995/01/01;00:00:00' 9999. <
'1995/02/15;00:00:00' 500. <
'1995/03/01;00:00:00' 9999. <
'1995/06/01;00:00:00' 750. <
'1995/12/31;23:59:00' 750. <
```

table par3

PAR3 id 'Cons.DemandOutlet1TimeTable' PDIN 1 1 365;00:00:00 pdin

TBLE

```
'1995/01/01;00:00:00' 0. <
'1995/03/01;00:00:00' 60. <
'1995/12/31;23:59:00' 60. <
```

table par3

Where:

id id of decision parameter

PDIN .. pdin= option for interpolation and periodicity

= 1 1 365;00:00:00 means block functions, periodicity one year

= 0 0 means linear interpolation, no periodicity

= 0 1 365;00:00:00 means linear interpolation, period one year

= 1 0 means block function, no periodicity

TBLE .. tble = the table, containing date; time string, value, and

Interpolation Tables

Interpolation tables can be applied by an “Interpolate” function, that determines the values of a particular “Interpolate Function Decision Parameter”. Values for the independent variable are contained in the first column of an interpolation table, while values for the dependent variable are contained in the second column. For the interpolation table given below yields that an input (independent) value of 0.15 results in an output (dependent) value of 0.025. Independent values may be contained in any data series defined in RTC. The dependent values will be assigned to the concerning “Interpolate Function Decision Parameter”.

Note that:

- 1 Independent values in the Interpolation Table (i.e. the values in the first Table column) should be given in ascending order,
- 2 If the independent value is less than the independent value on first Table row, than the dependent value will be equal to the dependent value on the first Table row,
- 3 If the independent value is larger than the independent value on the last Table row, than the dependent value will be equal to the dependent value on the last Table row.

```
INTP id 'InpTable1' nm 'InterpolationTable1' v1 'Heading_v1' v2 'Heading_v2'
TBLE
0.0 0.0 <
0.1 0.0 <
0.2 0.05 <
0.3 0.15 <
0.4 0.25 <
0.5 0.25 <
tbl intp
```

where:

id	id of interpolation table
nm	name of interpolation table
v1	heading of column 1 (Independent input variable)
v2	heading of column 2 (Dependent variable)

SOBEK reservoir

SOBEK offers functionality of modelling reservoirs. This is done by defining a RSVP record in the decision parameter file, and detailed reservoir information in the reservoir input file. An example of the RSVP record is described in this subsection and the reservoir file (Reservoir.Rtc) is described in the next subsection.

An example of the RSVP record:

```
RSVP id 'Res1' nm '' nb 2 nq 5 nt 2 ns 1 hav 'HeadVolumeReservoir1'
rule 'RuleCurveRsv1' hedg 'HedgingRule'
bg 'Bottom gate Definition' tg 'Turbine gate Definition' sg 'Spillway gate Definition'
dp 'RsvRes1BotGate1' 'RsvRes1BotGate2' 'RsvRes1TurbGate1'
'RsvRes1TurbGate2' 'RsvRes1SpillGate1'
no 2 gb 1 2 gt 1 2 gs 2 dm 'Cons.DemandOutlet1TimeTable'
'Cons.DemandOutlet2'
mf '9999' 'MaxFlowOutlet2TimeTable' il 'Initial RsvLvl1' ei 'Expected Inflow1' rsvp
```

Where:

id	reservoir id
nm	name of the reservoir
nq	total number of gates
nb	number of bottom gates
nt	number of turbine gates
ns	number of spillway gates SOBEK will check that nq=nb+nt+ns
dp	id's of decision parameters (flows over individual gates)
hav	reference to head-area-volume curve definition
rule	reference to rule curve definition
hedg	reference to hedging rule definition

il	reference to decision parameter representing initial reservoir level in m (initial= at begin of time step)
ei	reference to decision parameter representing expected inflow (in m ³ /s)
no	number of outlet links (check: no <= ng)
gb	assignment of bottom gates to outlet links
gt	assignment of turbine gates to outlet links
gs	assignment of spillway gates to outlet links
mf	reference to maximum flow definition for each outlet link
dm	reference to demand definition for each outlet link

Combination of different type of decision parameters

In the decision parameter file, all type of decision parameters (PARA, PAR2, PAR3, and RSVP) records can be used. There is no specific order of specification required. It is possible to use decision parameters in RSVP records (i.e. the discharges over the different gates) as 'input' decision parameters in PAR2 records.

Also the RSVP decision parameters may depend on other parameters (initial reservoir level, expected inflow, consumptive demands on each downstream outlet) which itself may be defined as decision parameters in PARA, PAR2, PAR3, or even other RSVP records.

Default RTC will compute the decision parameters in the order in which they are specified in the input file. However, SOBEK-RTC can also determine the order of computations itself (use the option SetSequenceDecisionParameters=-1 in the Ini file RTC.DAT). RTC will check whether cyclic definitions of decision parameters occur. It is not allowed to define decision parameter A depending on decision parameter B, decision parameter B depending on C, and C on A for the same time step. Such cyclic chains of dependencies are only allowed if somewhere a time delay is introduced (like the value of decision parameter A at time t is depending on the value of decision parameter B at time $t - 1$).

reservoi rtc-file (Decision layer)

This file contains the detailed information of the SOBEK-reservoirs. The information is related to the RSVP records in the decision parameter file (DECISPAR.RTC). The information in the reservoir input file consists of:

- ◊ level-area-volume curve of the reservoir (HAVC record)
- ◊ reservoir rule curves (RULE record)
- ◊ bottom gate data (BOTG record)
- ◊ turbine data (TURB record)
- ◊ spillway data (SPIL record)
- ◊ Q-h relation data for individual gates (QHRE record)
- ◊ maximum flow data for individual gates (MAXF record)
- ◊ flow demands (for energy generation) for individual turbine gates (ENGD record)
- ◊ hedging rule (HEDG record)

Each of these records is described below.

An example of each record with a description:

```
HAVC id 'HeadVolumeReservoir1'  
TBLE  
0100000000  
1010000000100000000
```

```
2010000000200000000
3010000000300000000
4010000000400000000
5010000000500000000
tble havc
```

Where:

id id of the head-area-volume curve relation

TBLE .. tble contains the relation;

the first column is the level (in m with respect to reference level),

the second column is the area [m^2]

the third column is the volume (m³)

```
RULE id 'RuleCurveRsv1' PDIN 0 1 365;00:00:00 pdin
TBLE
```

```
'1995/01/01;00:00:00' 50. 45. 40.
```

```
'1995/02/01;00:00:00' 50. 45. 40.
```

```
'1995/03/01;00:00:00' 50. 45. 40.
```

```
'1995/04/01;00:00:00' 50. 45. 35.
```

```
'1995/05/01;00:00:00' 50. 40. 30.
```

```
'1995/06/01;00:00:00' 45. 35. 25.
```

```
'1995/07/01;00:00:00' 40. 30. 20.
```

```
'1995/08/01;00:00:00' 35. 30. 20.
```

```
'1995/09/01;00:00:00' 35. 30. 20.
```

```
'1995/10/01;00:00:00' 40. 35. 25.
```

```
'1995/11/01;00:00:00' 45. 40. 30.
```

```
'1995/12/01;00:00:00' 50. 45. 35.
```

```
'1995/12/30;00:00:00' 50. 45. 40.
```

tble rule

Where:

id id of the rule curve

TBLE .. tble contains the rule curve time table;

the first column is the flood control level;

the second column is the target level;

the third column is the firm level;

(all levels in m with respect to reference level)

the flood control level \geq the target level \geq the firm level

```
BOTG id 'Bottom gate Definition' lv 0 10 qh 'Q-NetHead bottom gate' 'Q-NetHead bottom
gate'
```

```
mf 'Max.Flow 999' 'Max.Flow 999' botg
```

Where:

id	id of the bottom gate definition
lv	intake levels for the number of bottom gates (this number is defined in the RSVP record in DECISPAR.RTC)
qh	reference to Q-h relations for all bottom gates
mf	maximum flow definition for each individual bottom gate

TURB id 'Turbine gate Definition' lv 20 15 qh 'Q-NetHead turbine gate' 'Q-NetHead turbine gate'

ed 'Energy demands' 'Energy demands 2nd turbine' mf 'Max.Flow turbine gate' 'Max.Flow turbine gate' turb

Where:

id	id of the turbine gate definition
lv	intake levels for the number of turbines (this number is defined in the RSVP record in DECISPAR.RTC)
qh	reference to Q-h relations for all turbine gates
mf	maximum flow definition for each individual turbine gate
ed	energy demand definition for each individual turbine gate (energy demands can of course only be defined for the turbine gates, not for bottom gates or spillway gates)

SPIL id 'Spillway gate Definition' lv 50 qh 'Q-NetHead spillway gate' mf 'Max.Flow 999' spil

Where:

id	id of the spillway gate definition
lv	intake levels for the number of spillways (this number is defined in the RSVP record in DECISPAR.RTC)
qh	reference to Q-h relations for all spillways
mf	maximum flow definition for each individual spillway gate

MAXF id 'Max.Flow turbine gate' PDIN 1 1 365;00:00:00 pdin

TBLE

'1995/01/01;00:00:00' 31.25

'1995/01/15;00:00:00' 27.

'1995/02/01;00:00:00' 24.5

'1995/02/15;00:00:00' 19.

'1995/03/01;00:00:00' 50.

'1995/05/01;00:00:00' 50.5

'1995/06/01;00:00:00' 59.5

'1995/06/15;00:00:00' 59.5

'1995/07/01;00:00:00' 50.

'1995/07/15;00:00:00' 50.

'1995/08/01;00:00:00' 50.

'1995/08/15;00:00:00' 50.

'1995/09/01;00:00:00' 50.

'1995/10/01;00:00:00' 50.

'1995/11/01;00:00:00' 50.

'1995/12/01;00:00:00' 50.

'1995/12/15;00:00:00' 50.

'1995/12/31;23:59:00' 50.

table maxf

Where:

id id of the maximum flow definition

PDIN .. pdin = option for interpolation and periodicity

= 1 1 365;00:00:00 means block functions, periodicity one year

= 0 0 means linear interpolation, no periodicity

= 0 1 365;00:00:00 means linear interpolation, period one year

= 1 0 means block function, no periodicity

TBLE .. tble = the table, containing date;time string, value, and

QHRE id 'Q-NetHead bottom gate'

```
TBLE
0 999
10 999
20 999
30 999
40 999
45 999
50 999 tble qhre
```

Where:

id id of Q-h relation

TBLE .. tble contains the relation;

the first column is the flow (m³/s)

the second column is the water level h (m)

ENGD id 'Energy demands' PDIN 1 1 365;00:00:00 pdin

```
TBLE
'1995/01/01;00:00:00' 31.25
'1995/01/15;00:00:00' 27.
'1995/02/01;00:00:00' 24.5
'1995/02/15;00:00:00' 19.
'1995/03/01;00:00:00' 12.5
'1995/04/01;00:00:00' 12.5
'1995/05/01;00:00:00' 12.5
'1995/06/01;00:00:00' 12.5
'1995/10/01;00:00:00' 12.5
'1995/11/01;00:00:00' 12.5
'1995/12/31;23:59:00' 12.5
tble engd
```

Where:

id id of the energy demand definition

PDIN .. pdin = option for interpolation and periodicity

= 1 1 365;00:00:00 means block functions, periodicity one year

= 0 0 means linear interpolation, no periodicity

= 0 1 365;00:00:00 means linear interpolation, period one year

= 1 0 means block function, no periodicity

TBLE .. tble = the table, containing date;time string, value, and <

The values specified in the table are flows in m³/s.

HEDG id 'HedgingRule'

```
TBLE
100 100
90 100
70 75
50 50
30 25
10 0 tble hedg
```

Where:

id id of hedging rule relation

TBLE .. tble contains the relation;

the first column is the level percentage (between 0 and 100)

(0 corresponds with the level at dead storage (=the level at the lowest outlet);

100 corresponds with the firm level).

the second column is the release percentage (between 0 and 100)

B.28.5 Measures layer

The <D3D_MEAS.RTC> file links the structures to the decision values which contain the values to be used for controlling the structure (the measures). The file starts with the keyword FLM2.1 on the first line. The measures are specified in an FLCM record for each structure which is controlled by RTC. Each FLCM record contains one or more SBMS sub-records in which the measures for the structure are specified.

The general syntax reads:

```
FLCM id 'Delft3D structure name' nm '' na 0
      SBMS pr 1 ty 10 psp 'Decision variable name' iv 0.0 sbms
f1cm
```

where in the FLCM record:

id Delft3D structure name on which the measure is working
 nm not used for Delft3D, leave empty
 na option not active, 0 = active, <> 0 means not active. This option can be used
 to switch off RTC for this structure while keeping the definition

and in the SBMS record:

pr priority
 ty type of measure; the format of the record is depending on the type
 2 = table with n check values, n set points
 5 = record with n decision parameters, n check values and 1 set point (number);
 6 = record with n decision parameters, n check values and 1 set point as a
 decision variable
 7 = record with n decision parameters, n check parameters, 1 set point (number)
 8 = record with n decision parameters, n check parameters and a set point as a
 decision parameter
 9 = MATLAB measure
 10 = record with 1 setpoint decision parameter only
 bp id of decision parameter
 nv number of values or decision parameters (only for type ty 2, 5, 6, 7, 8)
 cv check value (only for type 2, 5, 6; nv values for type 2, 5, 6)
 ch check to be carried on on the check value (not for type 2)
 cp check decision parameter (only for measure type 7, 8)
 sp set point of the controller (PID or interval controller) or the controlled parameter
 (hydraulic controller or time controller) controlling the Flow-structure.
 psp parameter set point (only for ty 6, 8, 10)
 mi MATLAB communication id, the id which is used in MATLAB for this measure
 iv initial set point value;

The following table gives a concise overview of the measure types:

Measure Description	Decision parameters	Check value/parameters	Setpoint value/parameter	Measure Type No.
Type A	1	none	Interpolation Table	2
Type B1	n	n values	1 value	5
Type B2	n	n values	1 parameter	6
Type B3	n	n parameters	1 value	7
Type B4	n	n parameters	1 parameter	8
Type A	none	none	from MATLAB	9
Type B5	1	none	1 parameter	10

Remarks:

- ◊ For measure type 5-8, the value of n may also be equal to 1.



- ◊ It is possible to define multiple measures at one Flow controller. These measures can be linked to the same decision parameter as well, but that is not required. If one of the defined measures is active, the set point of local controller controlling the Flow structure will be adjusted to that measure.
- ◊ In defining measures of type 5 to type 8, the multiple condition check is performed using 'AND' logic: only if all conditions hold, the set point will be set according to the decision rule.
- ◊ With the exception of the MATLAB measure of type 9, all measure types are related to a decision parameter ('bp' field in the data record). For the MATLAB measure type, everything has to be defined in the MATLAB m-file.

Example:

The following lines define that the new flow rate of discharge point 'River 2' is given by the decision parameter 'River2_NewQ' and the new gate height of barrier 'Barrier 1' is given by 'Barrier1_NewH'.

```
FLM2.1
FLCM id 'River 2' nm '' na 0
    SBMS pr 1 ty 10 psp 'River2_NewQ' iv 0.0 sbms
flcm
FLCM id 'Barrier 1' nm '' na 0
    SBMS pr 1 ty 10 psp 'Barrier1_NewH' iv 0.0 sbms
flcm
```

C Astronomical constituents

C.1 Introduction

In this Appendix we list the astronomical constituents that you can use when prescribing astronomical boundary conditions for the flow. In total you can select from 234 tidal constituents. See [subsection 4.5.6](#) for details on how to use them in boundary conditions.

C.2 List of astronomical constituents

The set of constituents can be divided into the primary constituents, which appear in the equilibrium tide (no land masses; only one deep ocean), and compound constituents. The latter are linear combinations of the primary constituents.

The names and frequencies of all 234 internally available constituents of Deltares tidal analysis program Delft3D-TIDE are given below. For the primary constituents the relative magnitude in the equilibrium tide is given as well.

If you have only a short time-series for a tidal analysis you can use astronomical coupling between nearby constituents. To this end also the *equilibrium amplitude relation* with their main constituents is given. The *equilibrium phase relation* is equal to zero.

Remark:

- ◊ If you want to apply astronomical coupling, first try to find the amplitude and phase relations based on a long period analysis of a neighbouring station. Only if such information is not available, you may resort to the equilibrium tide relations.



Table C.1: Astronomical constituents

Constituent name	Angular frequency [degrees/hour]	Amplitude in equilibrium tide [m]	Amplitude coupling relation
SA	0.0410686	0.01156	
SSA	0.0821373	0.07281	
MSM	0.4715211	0.01579	
MM	0.5443747	0.08254	
MSF	1.0158958	0.01369	
MS0	1.0158958		
MF	1.0980331	0.15647	
KO0	1.0980331		
MK0	1.0980331		
SNU	1.4874169		
SN	1.5602705		
MSTM	1.5695542	0.00569	
MFM	1.6424078	0.02996	
2SM	2.0317916		
MSQM	2.1139289	0.00478	
MQM	2.1867825	0.00396	
2SMN	2.5761663		
2OK1	12.8450025		
2Q1	12.8542862	0.00955	0.025 × O1
NJ1	12.8542862		
SIGMA1	12.9271398	0.01152	
MUK1	12.9271398		
NUJ1	12.9271398		
Q1	13.3986609	0.07343	0.191 × O1

continued on next page

Table C.1 – continued from previous page

Constituent name	Angular frequency [degrees/hour]	Amplitude in equilibrium tide [m]	Amplitude coupling relation
NK1	13.3986609		
RO1	13.4715145	0.01395	$0.036 \times O1$
NUK1	13.4715145		
O1	13.9430356	0.38358	
TAU1	14.0251728	0.00504	
MP1	14.0251728		
M1B	14.4874103	0.01065	$0.350 \times M1A$
M1C	14.4920521		
M1A	14.4966939	0.02964	
M1	14.4966939	0.03150	$0.082 \times O1$
NO1	14.4966939		
CHI1	14.5695476	0.00580	
LP1	14.5695476		
PI1	14.9178647	0.01028	
TK1	14.9178647		
P1	14.9589314	0.17543	$0.328 \times K1$
SK1	14.9589314		
S1	15.0000000	0.00416	
K1	15.0410686	0.53496	
MO1	15.0410686		
SP1	15.0410686		
PSI1	15.0821353	0.00109	
RP1	15.0821353		
FI1	15.1232059	0.00755	
KP1	15.1232059		
THETA1	15.5125897	0.00578	
LABDAO1	15.5125897		
J1	15.5854433	0.03022	$0.079 \times O1$
MQ1	15.5854433		
2PO1	15.9748272		
SO1	16.0569644		
OO1	16.1391017	0.01939	$0.051 \times O1$
2KO1	16.1391017		
UPSILON1	16.6834764	0.00372	
KQ1	16.6834764		
2MN2S2	26.4079379		
3MKS2	26.8701753		
2NS2	26.8794590		
3MS2	26.9523126		
OQ2	27.3416964		
MNK2	27.3416964		
EPSILON2	27.4238337	0.00671	
MNS2	27.4238337		
2ML2S2	27.4966873		
MNUS2	27.4966873		
MNK2S2	27.5059710		
2MS2K2	27.8039339		
O2	27.8860711		
NLK2	27.8860711		
2MK2	27.8860711		
2N2	27.8953548	0.02303	$0.132 \times N2$
MU2	27.9682084	0.02776	$0.031 \times M2$
2MS2	27.9682084		
SNK2	28.3575922		

continued on next page

Table C.1 – continued from previous page

Constituent name	Angular frequency [degrees/hour]	Amplitude in equilibrium tide [m]	Amplitude coupling relation
NA2	28.3986628		
N2	28.4397295	0.17398	0.191 × M2
KQ2	28.4397295		
NB2	28.4807962		
NU2	28.5125831	0.03304	0.194 × N2
3MSN2	28.6040041		
2KN2S2	28.6040041		
OP2	28.9019669		
MSK2	28.9019669		
GAMMA2	28.9112506	0.00273	
ALFA2	28.9430356	0.00313	
MPS2	28.9430356		
MA2	28.9430356		
M2	28.9841042	0.90872	
KO2	28.9841042		
MSP2	29.0251728		
MB2	29.0251728		
DELTA2	29.0662415		
MKS2	29.0662415		
M2(KS)2	29.1483788		
2SN(MK)2	29.3734880		
LABDA2	29.4556253	0.00466	0.005 × M2
SNM2	29.4556253		
2MN2	29.5284789		
L2	29.5284789	0.02663	0.029 × M2
L2A	29.5284789	0.02569	
L2B	29.5377626	0.00704	0.274 × L2A
2SK2	29.9178627		
T2	29.9589333	0.02476	0.059 × S2
S2	30.0000000	0.42248	
KP2	30.0000000		
R2	30.0410667	0.00366	0.009 × S2
K2	30.0821373	0.12004	0.284 × S2
MSNU2	30.4715211		
MSN2	30.5443747		
ZETA2	30.5536584	0.00134	
ETA2	30.6265120	0.00702	
KJ2	30.6265120		
MKN2	30.6265120		
2KM(SN)2	30.7086493		
2SM2	31.0158958		
SKM2	31.0980331		
2MS2N2	31.0887494		
2SNU2	31.4874169		
2SN2	31.5602705		
SKN2	31.6424078		
MQ3	42.3827651		
NO3	42.3827651		
MO3	42.9271398		
2MK3	42.9271398		
2MP3	43.0092771		
M3	43.4761563	0.01780	
NK3	43.4807981		
SO3	43.9430356		

continued on next page

Table C.1 – continued from previous page

Constituent name	Angular frequency [degrees/hour]	Amplitude in equilibrium tide [m]	Amplitude coupling relation
MP3	43.9430356		
MK3	44.0251728		
SP3	44.9589314		
2MQ3	44.5695476		
SK3	45.0410686		
2SO3	46.0569644		
K3	45.1232059		
4MS4	55.9364168		
2MNS4	56.4079379		
3MK4	56.8701753		
MNLK4	56.8701753		
3MS4	56.9523126		
MSN4	57.3416964		
MN4	57.4238337		
MNU4	57.4966873		
2MLS4	57.4966873		
2MSK4	57.8860711		
M4	57.9682084		
2MKS4	58.0503457		
SN4	58.4397295		
3MN4	58.5125831		
2SMK4	58.9019669		
MS4	58.9841042		
MK4	59.0662415		
2SNM4	59.4556253		
2MSN4	59.5284789		
SL4	59.5284789		
S4	60.0000000		
SK4	60.0821373		
2SMN4	60.5443747		
3SM4	61.0158958		
2SKM4	61.0980331		
MNO5	71.3668693		
3MK5	71.9112440		
3MP5	71.9933813		
M5	72.4649024		
MNK5	72.4649024		
2MP5	72.9271398		
MSO5	72.9271398		
3MO5	73.0092771		
MSK5	74.0251728		
3KM5	74.1073101		
2(MN)S6	84.8476674		
3MNS6	85.3920421		
4MK6	85.8542796		
2NM6	85.8635632		
4MS6	85.9364168		
2MSNK6	86.3258006		
2MN6	86.4079379		
2MNU6	86.4807915		
3MSK6	86.8701753		
M6	86.9523126		
MSN6	87.4238337		
MNK6	87.5059710		

continued on next page

Table C.1 – continued from previous page

Constituent name	Angular frequency [degrees/hour]	Amplitude in equilibrium tide [m]	Amplitude coupling relation
4MN6	87.4966873		
MKNU6	87.5788246		
2(MS)K6	87.8860711		
2MS6	87.9682084		
2MK6	88.0503457		
2SN6	88.4397295		
3MSN6	88.5125831		
MKL6	88.5947204		
2SM6	88.9841042		
MSK6	89.0662415		
S6	90.0000000		
2MNO7	100.3509735		
2NMK7	100.9046319		
M7	101.4490066		
2MSO7	101.9112440		
MSK07	103.0092771		
2(MN)8	114.8476674		
3MN8	115.3920421		
3MNKS8	115.4741794		
M8	115.9364168		
2MSN8	116.4079379		
2MNK8	116.4900752		
3MS8	116.9523126		
3MK8	117.0344499		
2SNM8	117.4238337		
MSNK8	117.5059710		
2(MS)8	117.9682084		
2MSK8	118.0503457		
3SM8	118.9841042		
2SMK8	119.0662415		
S8	120.0000000		
2(MN)K9	129.8887361		
3MNK9	130.4331108		
4MK9	130.9774855		
3MSK9	131.9933813		
4MN10	144.3761463		
M10	144.9205210		
3MSN10	145.3920421		
4MS10	145.9364168		
2(MS)N10	146.4079379		
2MNSK10	146.4900752		
3M2S10	146.9523126		
4MSK11	160.9774855		
M12	173.9046253		
4MSN12	174.3761463		
5MS12	174.9205210		
3MNKS12	175.4741794		
4M2S12	175.9364168		

DRAFT

D Some modelling guidelines

D.1 Introduction

When setting up a model application you have to make many decisions and choices, such as do I need to resolve the full three dimensional hydrodynamics and transport processes, or are the main characteristics of the model area represented in a depth averaged model. Other choices concern coefficients and parameters to select, such as how large can I choose the time step and still represent the time dependent processes accurately, or in case of a 3D computation do I need to apply the advanced $k-\varepsilon$ model or can I use a simpler formulation for the vertical eddy viscosity and diffusivity.

Answering these questions in general terms is almost impossible as they are strongly related to the problem area at hand and the questions posed and answering them appeals to your modelling experience and even to some extent to your personal preferences. Having said this, you still have to make these choices and you are not helped by just saying ‘it is not a simple matter’. Therefore we discuss some of the criteria you can and in our opinion should apply to make a decision.

In the next section we present and discuss some guidelines to determine if you may need a 3D computation, what kind of model formulation you may need for the vertical eddy viscosity and diffusivity and on the time step you can apply.

D.2 Depth-averaged or 3D model

In this section we give some guidelines to determine if you need to resolve the vertical profile of the flow and the transport quantities or that you can use a depth averaged simulation.

In [Figure D.1](#) we present a decision tree to support your decision:

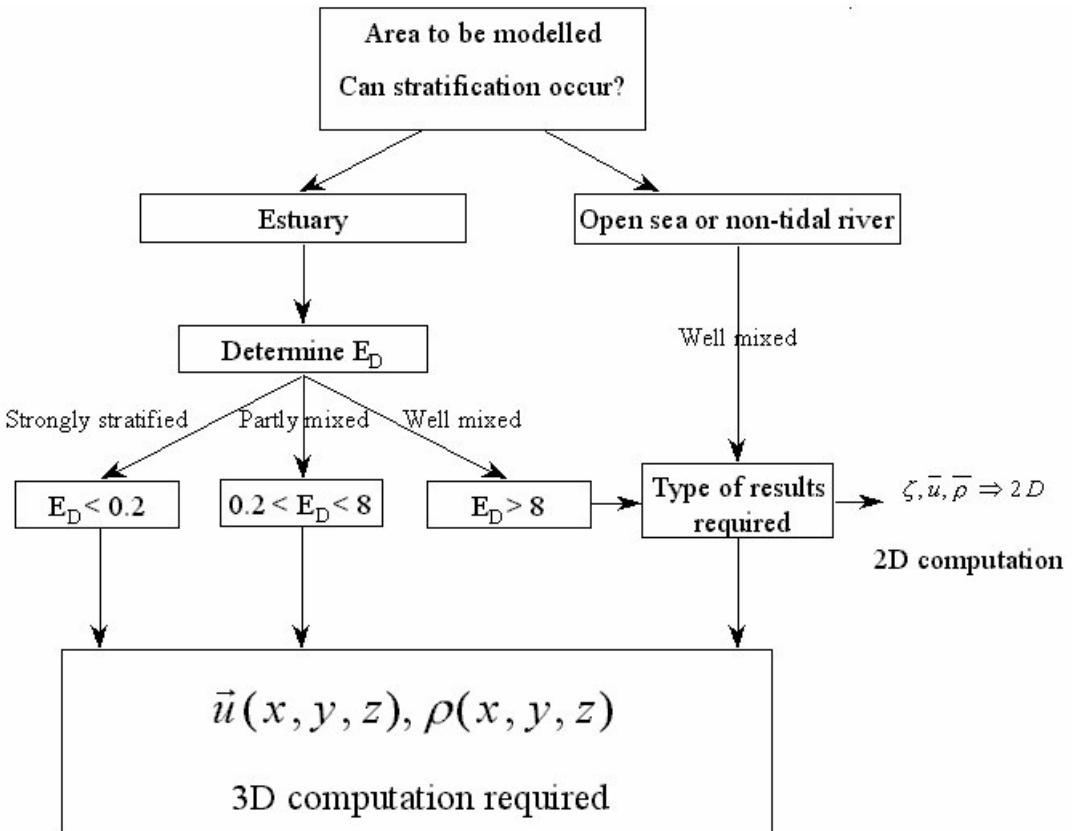
- ◊ What kind of area is to be modelled and can stratification occur?
- ◊ Non-tidal rivers and open seas are generally well-mixed and you can rely on a 2D (depth averaged) computation, unless the project requires the vertical profile of some quantities.
- ◊ Estuaries can range from well-mixed to strongly stratified.
- ◊ To determine the possibility of stratification you should determine the Estuary Number given by:

$$E_D = \frac{\rho u_t^3}{\pi \Delta \rho g h_0 u_r}, \quad (\text{D.1})$$

where:

g	acceleration of gravity [m/s^2],
h_0	depth at the mouth of the estuary [m],
u_r	river velocity, i.e. the river flow rate divided by the cross-sectional area at the mouth of the estuary [m/s],
u_t	amplitude of the profile averaged tidal velocity at the mouth of the estuary [m/s],
ρ	density of sea water [kg/m^3],
$\Delta \rho$	density difference between sea water and river water [kg/m^3].

Depending E_D ([Thatcher, 1981](#)), the estuary is strongly stratified, partly stratified or well-mixed:

**Figure D.1:** Steps to determine if a 3D model is required

$E_D < 0.2$ strongly stratified,

$0.2 < E_D < 8$ partly mixed,

$8 < E_D$ well-mixed.

- ◊ Only in case of a well-mixed estuary you can apply a 2D or depth averaged computation, unless of course the vertical profile of some quantity is required.
- ◊ In both the partly mixed and strongly stratified case you should in principal always use a 3D computation. If only depth averaged quantities are required you might apply a 2D computation in the partly mixed estuary, but in case of the strongly stratified estuary you should obtain depth averaged quantities from depth integration of the full 3D quantities.

In the next section we will discuss the choice of the vertical turbulence model in case of a 3D computation.

D.3 Selection of the vertical turbulence model

In this section we discuss the selection of the vertical turbulence model to be used, so it is assumed a full 3D computation will be executed.

We discuss the selection in relation to the vertical stratification.

D.3.1 Well-mixed

The resolution of the vertical profile was required for some specific reason, but not because some quantities were vertically stratified.

We distinguish two situations depending on the existence of (important) vertical circulations or not. These vertical circulations might be driven by the wind or by discharges into the area of concern:

- 1 No (important) vertical circulations.

You can apply a simple constant or spatially varying eddy viscosity or the $k-L$ turbulence model.

- 2 Yes, (wind driven) vertical circulations are important.

Depending on the complexity of the circulations you are suggested to apply the $k-L$ (simple cases) or the $k-\varepsilon$ turbulence model.

D.3.2 Partly mixed

You are suggested to apply the $k-L$ or the $k-\varepsilon$ turbulence model. The latter is the best alternative since it is user independent and its coefficients are fixed and well investigated ([Uittenboogaard et al., 1992](#)). The former is simpler and requires less computing time. In case of doubt: select the $k-\varepsilon$ turbulence model.

D.3.3 Strongly stratified

It is recommended to use the $k-\varepsilon$ turbulence model.

DRAFT

E Computational grid

In this Appendix we discuss several details related to the use of the staggered grid, the grid enclosure and the numbering and location of open and closed boundaries. For the general use of Delft3D this information is not needed, for all practical applications the use of these items is completely transparent to you and you do not need to be aware of the definition and implementation details.

However, if you want to inspect certain files, such as the grid file or the bathymetry file and want to check all details in relation to values and locations, you must be aware of certain details related to definition and implementation. These details are discussed and illustrated with examples in this Appendix. You can find details related to the numerical scheme in [chapter 10](#) and a detailed description of the structure and the contents of the related files in [section A.2](#).

Essential to understanding the use and implementation of the staggered grid system is the grid numbering and the definition of a computational control volume (sometimes called a computational grid cell, or a computational tile). In [Figure E.1](#) (left) the staggered grid and the numbering is given, indicating which water level, velocity components and depth have the same (array) number in x - and y -direction in the computational code.

The numerical grid developed with the grid generation program RGFGRID is the grid defining a computational control volume. With the depth defined at the corners of the computational control volume, the numerical grid is drawn through the depth points. This relation between the numerical grid and the computational control volume also controls the location of the open and closed boundaries.

From [Figure E.1](#) right it is clear that left and below the first computational control volume there must exist a grid (array) point to hold the boundary condition, be it a water level or a velocity controlled boundary. This is even more clear from [Figure E.2](#) left where the lower left computational control volume is displayed.

To the right and above the most right and upper computational control volume no additional grid point is required if the boundary condition would always be a velocity or discharge boundary condition. The boundary condition would simply be described at the velocity point in the respective edge. However, for water level boundaries the situation is less simple. An additional grid (array) point must exist outside the last computational control volume to prescribe the boundary condition, [Figure E.2](#) right.

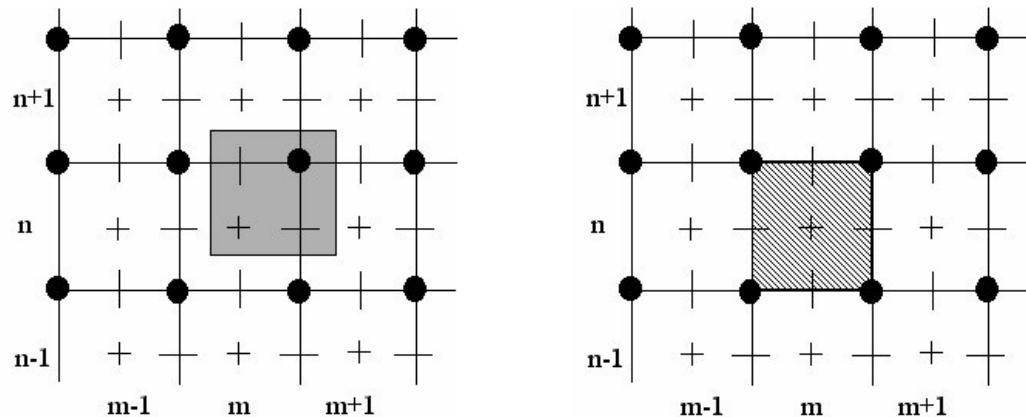
To discuss the relation between the computational grid and the location of boundary conditions we use the simple case of a rectilinear area without any islands, [Figure E.3](#).

Remark:

- ◊ From [Figure E.3](#) alone it can not be decided whether the area is an open sea area, an area fully enclosed by land or some mixture.

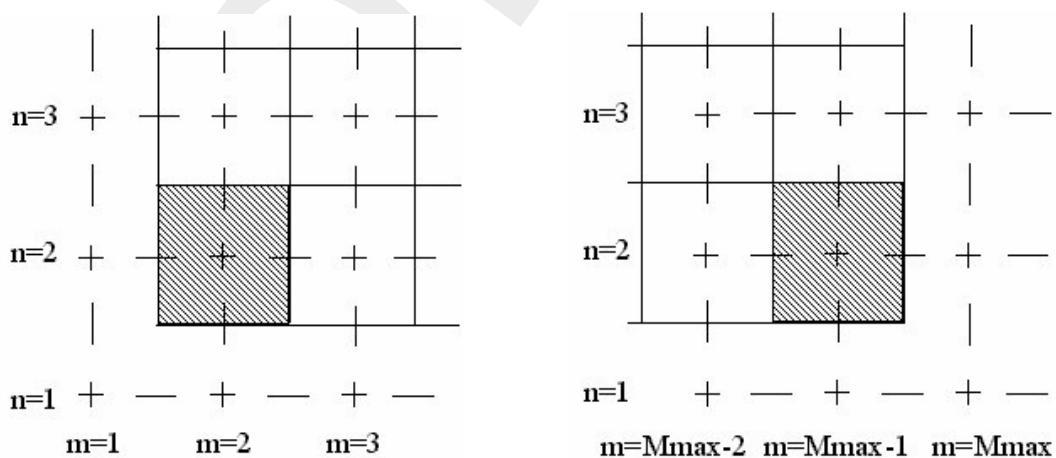


At closed boundaries the normal velocity is zero. From [Figure E.2](#) it is obvious that closed boundaries can be defined at the velocity point of the first or last grid point. The same holds for an open boundary with velocity or discharge boundary conditions. For the left and lower boundary this velocity point is located at the first numerical grid line and its logical grid number is 1. For the right and upper boundary the velocity at the edge of the last computational control volume is located at grid (array) point Mmax-1 or Nmax-1. However, for implementation reasons the velocity boundary at the right and upper side of the area is virtually also defined at grid point Mmax or Nmax; in the implementation of the computational core the boundary



Legend:

full lines	the numerical grid.
grey area	left figure: items with the same (array) number (m, n). right figure: computational control volume.
+	water level, concentration of constituents, salinity, temperature.
-	horizontal velocity component x -, ξ -, u - or m -direction.
	horizontal velocity component in y -, η -, v - or n -direction.
•	depth.

Figure E.1: Left: items with the same (array) number. Right: a computational control volume**Figure E.2:** Lower-left (left) and lower right (right) computational grid cell

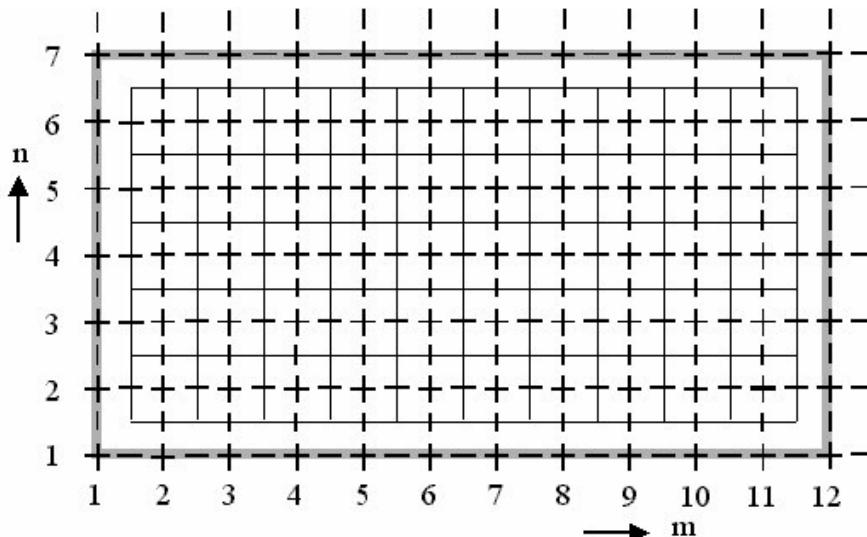


Figure E.3: Definition sketch of a $(12 * 7)$ staggered grid with grid enclosure (thick line) and numerical grid

condition is shifted to the grid (array) points $M_{max}-1$ and $N_{max}-1$. For open boundaries with water level control the water level point is located just outside the first or last computational control volume used.

In most practical applications both open and closed boundaries are used and both water level and velocity controlled open boundaries are used in the same application.

In Figure E.4 we use the more general case of Figure E.3 with the location and combination of water level and velocity controlled open boundaries at the four edges and corners. The open boundary points are indicated by encircled water level or velocity points.

Remark:

- ◊ The shift in the location of water level and velocity controlled open boundary has consequences for the overall dimensions of the area being modelled. For all practical applications this relation is transparent, but for simple cases where you want to check certain properties, such as mass conservation, this is important, see the example.



Water level controlled open boundaries:

Water level controlled open boundaries are always located just outside the first or last computational control volume. The line through these water level points is called the (external) grid enclosure.

For Figure E.4 water level controlled open boundaries can be located at:

- 1 lower: ($m=2$ to $M_{max}-1$, $n=1$),
- 2 left: ($m=1$, $n=2$ to $N_{max}-1$),
- 3 upper: ($m=2$ to $M_{max}-1$, $n=N_{max}$),
- 4 right: ($m=M_{max}$, $n=2$ to $N_{max}-1$).

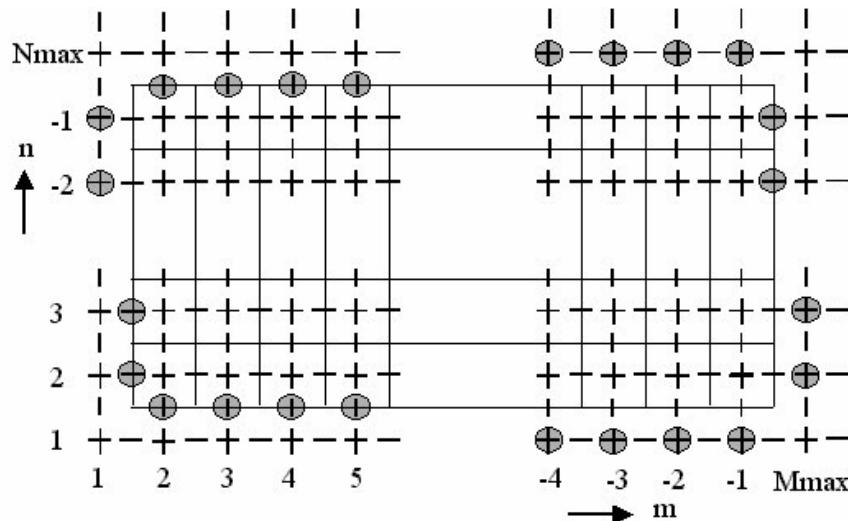


Figure E.4: Location and combination of water level and velocity controlled open boundaries

Velocity controlled open boundaries:

Velocity controlled open boundaries are either located at the (external) grid enclosure (in terms of the grid point (array) number) for the left and lower boundaries, or just inside the grid enclosure (in terms of the grid point (array) number) for the right and upper boundaries.

However, to make the location of open boundaries irrespective of the open boundary type, *also the right and upper boundaries are indicated as being located at the grid enclosure* (at implementation level the grid number is decreased by one).

For [Figure E.4](#) the velocity controlled open boundaries can be located at:

- 1 lower: ($m=2$ to $M_{max}-1$, $n=1$),
- 2 left: ($m=1$, $n=2$ to $N_{max}-1$),
- 3 upper: ($m=2$ to $M_{max}-1$, $n=N_{max}$), internally ($m=2$ to $M_{max}-1$, $N_{max}-1$),
- 4 right: ($m=M_{max}$, $n=2$ to $N_{max}-1$), internally ($m=M_{max}-1$, $n=2$ to $N_{max}-1$).



Remark:

- ◊ The water level points at the corner points (1,1), (1, M_{max}), (M_{max} , N_{max}) and (1, N_{max}) are not used in the numerical approximation of the hydrodynamic equations.

For internal grid enclosures (defining an internal area with either closed or open boundaries) the same rules apply.

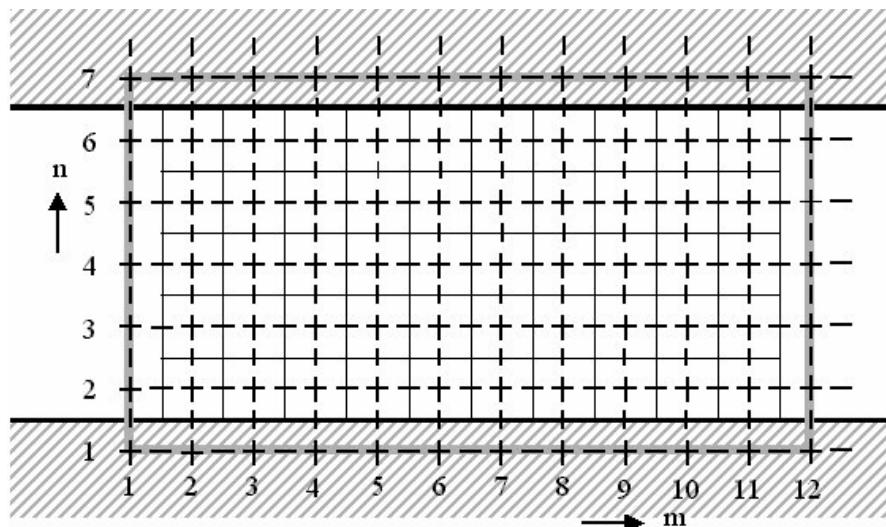


Figure E.5: Straight channel with $10 * 5$ computational grid cells

On summarising:

- 1 The grid generator displays the computational control volumes, i.e. the grid cells that are (potentially) used in the computation.
- 2 For the definition of open boundaries the number of (possibly) active grid points in both (horizontal) directions is extended by one grid point.
- 3 These additional grid (array) points are *included* in the number of grid points as displayed in the FLOW-GUI; Mmax and Nmax.
- 4 The grid file of the grid generator only includes the indices of the computational control volumes, i.e. does not include indices of the additional grid (array) points.
- 5 The grid enclosure generated by RGFGRID takes these additional (array) points into account.
- 6 QUICKIN takes the additional grid (array) points into account and *automatically* assigns them a value of -999.
- 7 In the **Visualisation Area** of the FLOW-GUI the numerical grid is displayed as created with RGFGRID . The additional points nor the grid enclosure are displayed.

Example:

Suppose you want to define a rectilinear grid in a channel of dimensions $10\,000 * 5\,000 \text{ m}^2$ with a grid size of 1000 m. This will require a grid of $10 * 5$ grid *cells* and thus $11 * 6$ grid *points*. When you apply RGFGRID to generate this grid you must set the grid dimensions to $11 * 6$ (RGFGRID handles grid points, not grid cells).

The grid file will contain 2 blocks of 6 rows with 11 grid point indices, one block for the *x*-co-ordinates and the second block for the *y*-co-ordinates.

The grid enclosure will include the extra grid (array) points and the default grid enclosure will go through the points (1, 1), (12, 1), (12, 7) and (1, 7).

In the **Visualisation Area** a grid will be displayed of $11 * 6$ grid points (with $10 * 5$ grid cells). In the FLOW-GUI Mmax = 12 and Nmax = 7.

[Figure E.5](#) shows the grid and the grid enclosure for this example.

The closed (thick) outer line is the grid enclosure, the thinner inner line represents the compu-

tational grid. The number of computational control volumes inside the area enclosed by thin lines is $10 * 5$, as it was meant to be.

**Remarks:**

- ◊ The channel *only* has the required area of $5 \cdot 10^7 \text{ m}^2$ if the left and right open boundaries are *velocity* controlled. The channel area will be $5.5 \cdot 10^7 \text{ m}^2$ when both open boundaries are water level controlled and $5.25 \cdot 10^7 \text{ m}^2$ with a combination of water level and velocity controlled boundaries.
- ◊ In (almost) all practical applications this artefact is unimportant and transparent, only in these simple cases you must be aware of these peculiarities.
- ◊ You must take into account the type of boundary conditions you are going to apply when you define the computational grid.

F Delft3D-NESTHD

F.1 Introduction

Delft3D-FLOW models require hydrodynamic and transport boundary conditions. In case the boundary conditions of a model are generated by a larger (overall) model we speak of a nested model. Nesting in Delft3D-FLOW is executed in three steps, using two separate utilities and the Delft3D-FLOW program. This manual describes the steps to generate time-series boundary conditions for a nested Delft3D-FLOW model.

In principle, the nested boundary conditions are generated by bi-linear interpolation of computational results at monitoring stations of the overall model.

The procedure to generate nested time-series boundary conditions consists of 3 steps:

- 1 Using the tool NESTHD1 a list of monitoring stations in the overall model, needed for the interpolation, can be generated. In addition to this, the program generates the nesting administration, i.e. the link between the boundary support points in the nested model and the monitoring stations in the overall model.
- 2 Run the overall model with the list of monitoring stations generated by NESTHD1.
- 3 The actual time-series boundary conditions for the nested model are generated by NESTHD2 using the history file of the overall model and the nest administration.

Restrictions:

At present, the nesting programs have the following limitations:

- 1 Both model grids must be supplied in separate grid files and must be defined in the same co-ordinate system.
- 2 Hydrodynamic time-series boundary conditions for the nested model are of the water level or (perpendicular) velocity type. Discharge and Riemann boundary conditions are not implemented yet.
- 3 The boundary definition of the nested model has to be supplied as an attribute file.
- 4 Temporary dry points in the overall model are not taken into account in the nesting programs.
- 5 For a nested 3D model the number and the thickness of the layers must be identical to the overall model.
- 6 The generated time-series boundary conditions are written to a time-series file. This implies that the reference date of a simulation with the nested model must be identical to the reference date of the overall model simulation used to generate the boundary conditions.

F.2 How to use NESTHD1

NESTHD1 is part of the Delft3D-FLOW tools. To access NESTHD1, first set the current directory to the directory where you have stored the required input files. Then:

- Select **Tools** in the **Hydrodynamics** selection window, see [Figure F.1](#):

Next [Figure F.2](#) is displayed.

To start NESTHD1:

- Select **Nesting (1)**.

You are asked to supply the following information, see also [Figure F.3](#):

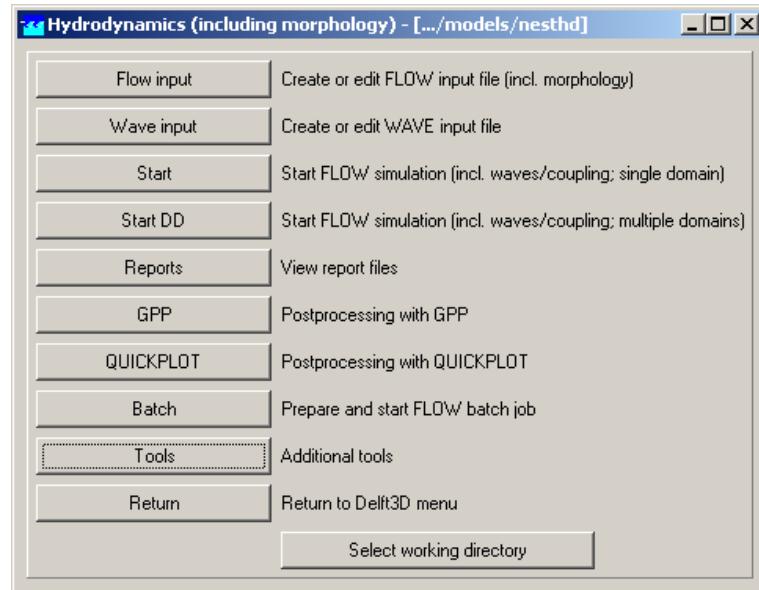


Figure F.1: *Hydrodynamics* selection window with the Tools option

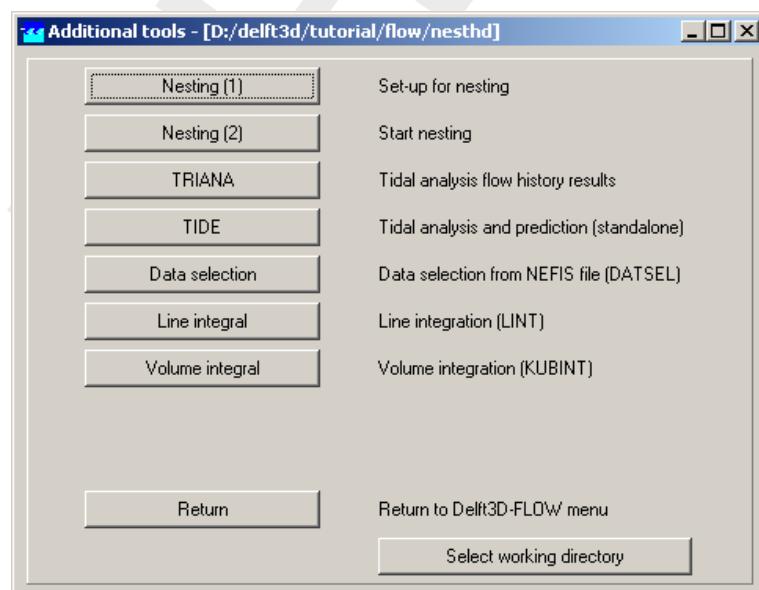


Figure F.2: *Additional tools* window with the NESTHD1 and NESTHD2 options

```

D:\Delft3D\tutorial\flow\nesthd\nesthd1.exe
Deltares, NESTHD1 Version 1.56.01.5067, Oct 30 2008, 09:52:26

Overall model
File name grid file      : overall.grd
File name enclosure file : overall.enc

Detailed model
File name grid file      : nested.grd
File name enclosure file : nested.enc
File name boundary definition file: nested.bnd

File name administration file : nesthd1.adm
File FLOW observation file   : nesthd1.obs

```

Figure F.3: Specification of input and output files for NESTHD1

- 1 The filename of the grid file of the overall model.
- 2 The filename of the enclosure file of the overall model.
- 3 The filename of the grid file of the nested model.
- 4 The filename of the enclosure file of the nested model.
- 5 The filename of the boundary definition file of the nested model. This file must fulfil the conventions as specified in the Delft3D-FLOW manual.
- 6 The name of the file which will contain the nest administration. An example of this file as used for nesting of the Stonecutters Island model in the Pearl Estuary model is given in [section F.4](#).
- 7 The name of the file which will contain the position of the monitoring stations in the overall model. This file can be used directly in a Delft3D-FLOW simulation with the overall model.

After running NESTHD1 a simulation with the overall model should be executed using the monitoring stations as generated by NESTHD1.

F.3 How to use NESTHD2

NESTHD2 is part of the Delft3D-FLOW tools. To access NESTHD2, first set the current directory to the directory where you have stored the required input files. Then enter the *Tools* menu, see [Figure F.1](#).

To start NESTHD2, see [Figure F.2](#):

➤ Select *Nesting* (2).

You are asked to supply the following information, see also [Figure F.4](#):

- 1 The boundary definition file of the detailed model.
- 2 The name of the administration file as generated by NESTHD1.
- 3 The extension (runid) of the history file of the overall model.
- 4 The name of the file which will contain the hydrodynamic boundary conditions. This file can be used directly in a simulation with the nested model.
- 5 The name of the file which will contain the transport boundary conditions. This file can be used directly in a simulation with the nested model.
- 6 The filename of the diagnostic file. You are advised to inspect this file.
- 7 An optional (constant) adjustment of the boundary values at water level boundaries. This can be used if both models have a different vertical reference level.
- 8 In case of a 3-dimensional overall model and a nested model with velocity boundaries you are asked if you want to generate depth-averaged or 3-dimensional velocity boundary conditions.

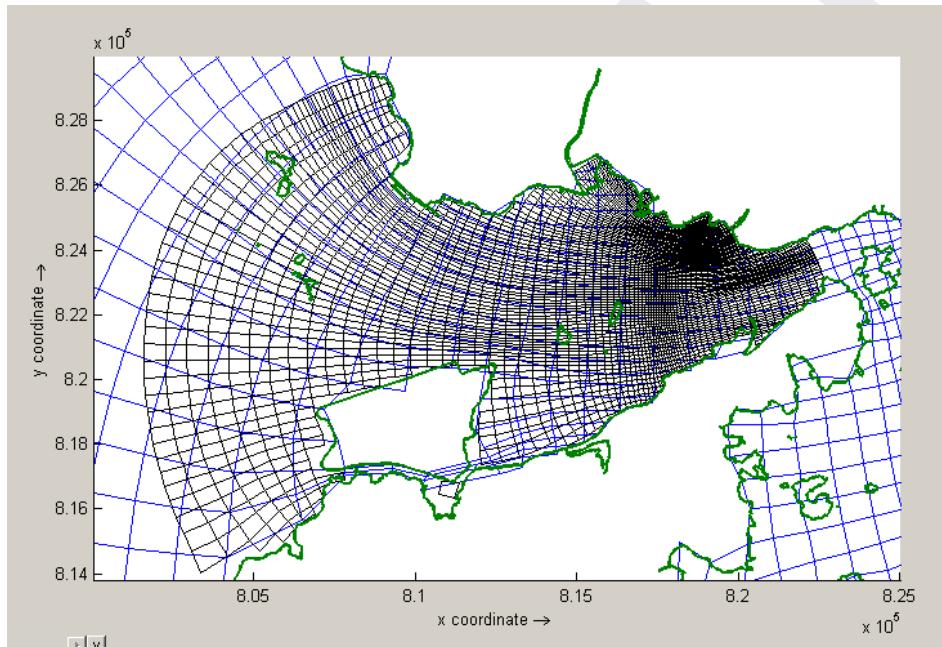
```

c:\D:\Delft3D\tutorial\flow\nesthd\nesthd2.exe
Deltares, NESTHD2 Version 1.57.00.5067, Oct 30 2008, 09:52:45
Detail model
File name boundary definition file : nested.bnd
File name administration file : nesthd1.adm
Overall model
Run-id NEFIS files : o2d
Output for detail model
File name hydrodynamic bc. : siulam2d.bct
File name transport bc. : siulam2d.bcc
File name diagnostic file : nesthd2.dia

A0 correction water level boundaries : 0
Generate boundary conditions for salinity <y/n> ? y
Add to salinity : 0
Maximum salinity : 40
Minimum salinity : 0

Type of profile:
1) Uniform <2Dh>
2) Linear
3) 3D :1_

```

Figure F.4: Specification of input and output files for NESTHD2**Figure F.5:** Overview grids overall and nested models

- 9 For each of the constituents in the overall model you are asked:
 - 9.1 If you want to generate boundary conditions for this constituent.
 - 9.2 How much you want to add to the computed overall model value for this constituent.
 - 9.3 The maximum value for this constituent.
 - 9.4 The minimum value for this constituent.
- 10 The type of transport profile. You can chose between a uniform profile for depth-averaged computations with the nested model, a linear profile using only the computed boundary values at the surface and the bed and a 3D-profile resulting in boundary values for all computational layers of the nested model.

F.4 Example

This example describes the nesting of the Siu Lam model in the Pearl Estuary model. The grid layout of both models is given in [Figure F.5](#):

The input and output files of this example are stored in the FLOW Tutorial directory: <tutorial\flow\nesthd>.

Using NESTHD1 the monitoring stations file and the administration file are generated. The contents of these files are shown below:

Part of monitoring stations file <nesthd1.obs>:

```
(M,N) = ( 27, 35)      27 35
(M,N) = ( 27, 36)      27 36
(M,N) = ( 27, 41)      27 41
(M,N) = ( 27, 42)      27 42
(M,N) = ( 28, 35)      28 35
(M,N) = ( 28, 36)      28 36
. . . .
. . . .
(M,N) = ( 29, 35)      29 35
(M,N) = ( 29, 36)      29 36
(M,N) = ( 29, 41)      29 41
(M,N) = ( 29, 42)      29 42
(M,N) = ( 31, 30)      31 30
(M,N) = ( 31, 31)      31 31
```

These monitoring stations have been included (added) to the already existing <*.obs> file of the Pearl model. In this tutorial <hkcrssl1.obs>.

Part of the administration file <nesthd1.adm>:

```
*
* Deltares, NESTHD1 Version 1.56.01.5067, Oct 30 2008, 09:52:26
*
* Run date: 2008/10/30 09:56:52
*
* Name grid file overall model          : overall.grd
* Name enclosure file overall model    : overall.enc
*
* Name grid file detailed model        : nested.grd
* Name enclosure file detailed model  : nested.enc
* Name bnd. definition file detailed model : nested.bnd
*
* Name nest administration file       : nesthd1.adm
* Name FLOW observation file         : nesthd1.obs
*
Nest administration for water level support point (M,N) = (   49,      1)
 28      45 0.5107
 29      45 0.1822
 28      46 0.2264
 29      46 0.0808
Nest administration for water level support point (M,N) = (   56,      1)
 28      43 0.1496
 29      43 0.0271
 28      44 0.6972
 29      44 0.1261
. . .
. . .
Nest administration for water level support point (M,N) = (   53,     74)
 47      45 0.8563
```

```

48      45 0.1437
0       0 0.0000
0       0 0.0000
Nest administration for water level support point (M,N) = ( 57, 74)
47      44 0.3268
48      44 0.0964
47      45 0.4454
48      45 0.1314
Nest administration for velocity support point (M,N) = ( 49, 1) Angle = 20.409
28      45 0.3888
29      45 0.3216
28      46 0.1585
29      46 0.1311
Nest administration for velocity support point (M,N) = ( 56, 1) Angle = 20.409
28      43 0.1368
29      43 0.0773
28      44 0.5022
29      44 0.2838
. . .
. . .
Nest administration for velocity support point (M,N) = ( 53, 74) Angle = 122.632
47      45 0.9973
48      45 0.0027
0       0 0.0000
0       0 0.0000
Nest administration for velocity support point (M,N) = ( 57, 74) Angle = 122.632
47      44 0.4247
48      44 0.0355
47      45 0.4982
48      45 0.0416

```

**Remarks:**

- ◊ When you run the NESTHD1 tool, the location of the nested boundary should be final.
- ◊ The type of data (water level or velocity) need not be final. That's why NESTHD1 generates the administration for both.
- ◊ The type of data should also be final. NESTHD1 only generates the administration for *time-series*.

The first lines, starting with an asterisk, are comment lines. These are followed by several blocks of 5 lines containing the information on nesting of water level and transport boundaries. The first of these lines gives the indices of the boundary support point. This line is followed by 4 lines containing the indices of the surrounding overall model monitoring stations and the relative weights of these stations. The information on nesting of water levels and transport is followed by information on nesting of velocities.

To adjust the nesting, for instance to avoid the use of a monitoring station that dries during the computation, the administration file should be adjusted by hand.

Before running NESTHD2, first the overall model should be run, using the <o2d.mdf> file.

**Remark:**

- ◊ It is a good modelling practice to re-generate the bathymetry in the overall model using the detailed bathymetry of the nested model.

If you Verify this scenario the following warning will be generated:

```
*** WARNING Station (M,N) = ( 28, 46) lies outside the computational domain
```

This means that this required nesting stations lies on a dry point. If you check the adminis-

tration file, you will see that every boundary support point has at least 2 overall nest stations. Ignoring the nest station which lies on a dry point would result in the worst case that the nesting information only comes from 1 overall station. (If you check the administration file carefully, you will find out that the mentioned station is always part of a set of 4.)

Using the administration file <nesthd1.adm> and the history file <trih-o2d.*> of the Pearl Estuary model, the boundary conditions for the Siu Lam model have been generated by running NESTHD2.

The result files are the hydrodynamic flow boundary conditions <siulam2d.bct> and the transport boundary conditions <siulam2d.bcc>.

The diagnostic file contains the following warning:

```
*** Warning: Nest station ( 28, 46) not on history file. Weights reset
```

This station, as noted before, lies on a dry point. NESTHD2 then resets weights.

DRAFT

DRAFT



Deltares systems

PO Box 177
2600 MH Delft
Rotterdamseweg 185
2629 HD Delft
The Netherlands

+31 (0)88 335 81 88
sales@deltaressystems.nl
www.deltaressystems.nl