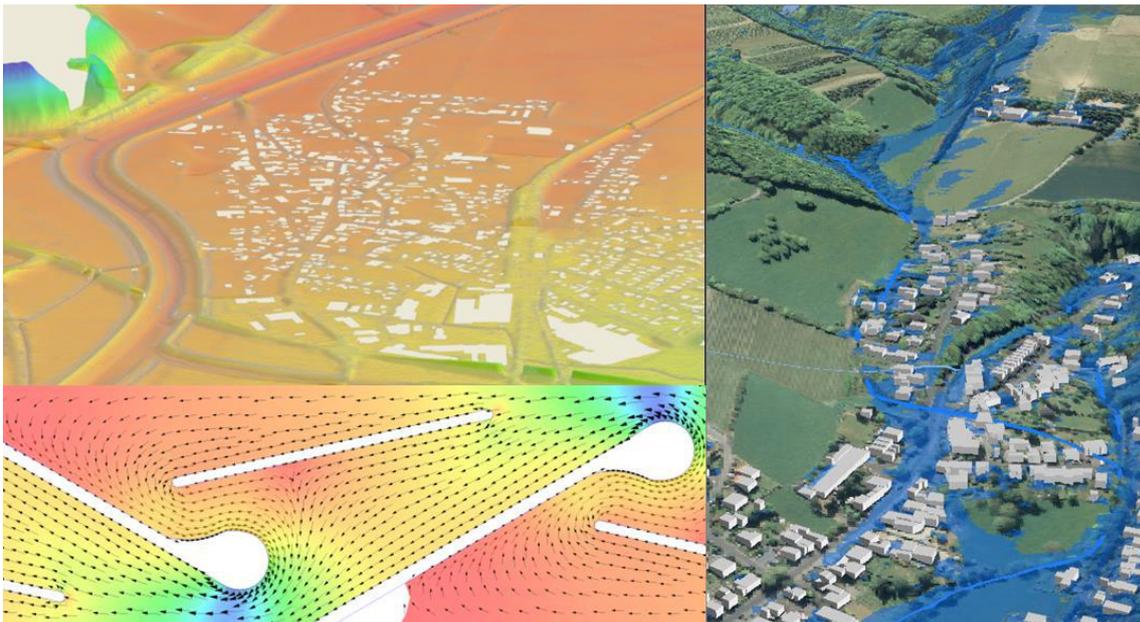




Benutzerhandbuch

HYDRO_AS-2D

2D-Strömungsmodell für die wasserwirtschaftliche Praxis



Version 5.3.1
Aachen, Oktober 2021

Hydrotec Ingenieurgesellschaft für Wasser und Umwelt mbH

HYDRO_AS-2D Benutzerhandbuch

Version 5.3.1

Stand: Oktober 2021

Hydrotec
©Alle Rechte vorbehalten

Jede Art der Vervielfältigung – auch auszugsweise – ist nur mit Genehmigung gestattet.

Hydrotec Ingenieurgesellschaft für Wasser und Umwelt mbH

Bachstraße 62-64
52066 Aachen
Tel: 0241 / 946890
Fax: 0241 / 506889
E-Mail: vertrieb@hydrotec.de
www.hydrotec.de

Inhaltsverzeichnis

Abbildungsverzeichnis	IV
Tabellenverzeichnis	V
1 Einführung	1
1.1 Über das Benutzerhandbuch	1
1.2 HYDRO_AS-2D: Anwendungsbereiche, Möglichkeiten und Grenzen	1
2 Hydromechanische Gleichungen	2
2.1 2D-Strömungsgleichungen	2
2.1.1 Rauheit	3
2.1.2 Viskosität	4
2.1.3 Anfangs- und Randbedingungen	4
2.2 1D - Bauwerke	6
2.2.1 Überströmte Bauwerke	7
2.2.2 Durchströmte Bauwerke	8
2.2.3 Abflusskoeffizient	9
2.2.4 Anwendungskriterien der 1D-Bauwerke	10
3 Numerisches Lösungsverfahren	11
3.1 Räumliche und zeitliche Diskretisierung	11
3.2 Eigenschaften der in HYDRO_AS-2D verwendeten Methoden	12
3.3 1step und 2step	13
4 Modellerstellung	14
4.1 Einführung	14
4.2 SMS	14
4.3 Übernahme von topographischen Geländedaten und Netzerstellung	16
4.4 Rauheit und Viskosität	17
4.5 Randbedingungen an Nodestings	20
4.5.1 Zulauf	22
4.5.2 Zulauf gebunden an Auslauf	23
4.5.3 Auslauf / le	25
4.5.4 Auslauf W/Q-Strickler	25
4.5.5 Auslauf W/Q-Wehr	26
4.5.6 Auslauf H-Wehr (Zeit)	27
4.5.7 Auslauf H-Wehr (Steuerung)	30
4.5.8 Auslauf Abflussganglinie	32
4.5.9 H-Randbedingung	32
4.5.10 Durchlässe	33
4.5.11 Wehrüberfall	34
4.5.12 Kontrollquerschnitt	35
4.5.13 Namen für Nodestings	36
4.6 Randbedingungen an Netzknoten	38
4.6.1 Konstruktionsunterkante (KUK/Druckabfluss)	38
4.6.2 Pegelpunkte	40
4.6.3 Quellterme an Netzknoten: Niederschläge als Zeitreihen in SMS	40
4.6.4 Quellterme an Netzknoten: Konstante Zuflüsse/Abflüsse	42
4.6.5 Quellterme an Netzknoten: Zufluss-/Abflusszeitreihen (Schächte)	45
4.6.6 Quellterme an Netzknoten: zeitabhängige Zuflüsse (sources-in)	47
4.7 Anfangswasserstand	47

4.8	Anfangsgeschwindigkeit	50
4.9	Dateiformate der Eingangsdateien	50
4.10	Netzqualität	52
4.11	Räumliche Auflösung	53
4.12	Speichern von Modelldaten	54
4.13	Berechnungskontrolle und Sensitivitätsuntersuchung	54
5	Durchführung der Berechnung	55
5.1	Upgrade von Modelldaten auf aktuelle Version	55
5.2	Konvertieren von Ein- und Ausgabedateien	56
5.3	Multiprozessorumgebung	57
5.4	Auswahl von Lizenzen	57
5.5	Eingabedateien für den Präprozessor HYDRO_2DM	58
5.6	Ausgabedateien des Präprozessors/Eingabedateien für die Simulationsprogramme	58
5.7	Ausgabedateien der Simulationsprogramme	59
5.7.1	Ausgabe der Wasserspiegellagen	63
5.7.2	Berechnung von Maximalwerten und Summenwerten	64
5.7.3	Ausgabe der Volumenbilanz	64
5.7.4	Ausgabedateien im Binärformat	65
5.8	Fortsetzung eines Rechenlaufs	65
5.9	Kontrolliertes Beenden eines Rechenlaufs und Aktivierung zusätzlicher Ausgaben	66
5.10	Optimieren von Rechenzeiten	67
5.10.1	1step vs. 2step	67
5.10.2	Analyse der Zeitschrittweiten	67
5.10.3	Konstruktionsunterkanten	68
5.10.4	Wassertiefe	68
5.10.5	Amin	68
5.10.6	Schreiben der Ergebnisse	69
6	Graphische Benutzeroberfläche	70
6.1	Allgemeines zur Benutzeroberfläche	71
6.1.1	Gruppierung von Bedienelementen	71
6.1.2	Optionen	71
6.2	Öffnen einer 2dm-Datei	71
6.3	Bedienelemente der Benutzeroberfläche	72
6.4	Importprotokoll	72
6.5	Prüfungen	74
6.5.1	Anzeigen der Prüfergebnisse in SMS	77
6.6	Neu laden der 2dm-Datei	78
6.7	Simulation durchführen	78
6.8	Exportieren und Konvertieren in aktuelle Programmversion	81
6.9	MapView-Datensatz erzeugen	81
6.9.1	MapView Benutzeroberfläche	81
6.9.2	MapView Kommandozeilen Tool	85
7	Ergebnisdarstellung	89
7.1	Ergebnisdarstellung mit SMS	89
7.2	Ergebnisdarstellung mit JabPlot	89
7.3	Ergebnisdarstellung mit MapView	92

8	Steuerung mit Scripting	94
8.1	Einführung: Möglichkeiten der Scripting-Steuerung	94
8.2	Technische Einführung	95
8.2.1	Grundprinzip des Scripting	96
8.2.2	Beispiel: Wehrsteuerung	97
8.2.3	Statistische Auswertung	101
8.3	Verfügbare Callbacks	104
8.4	Allgemeine Hinweise zur Lua-Programmierung	105
8.4.1	Methodentypen	105
8.4.2	Ganze Zahlen vs. Fließkommazahlen	106
8.4.3	Einheiten	106
8.4.4	Anlegen von Objekten	106
8.5	Überblick über alle Scripting-Klassen	108
8.6	Referenz Scripting	109
8.6.1	Klasse hydroas.Culvert	109
8.6.1.1	Statische Klassenfunktionen	109
8.6.1.2	Objektmethoden	111
8.6.2	Klasse hydroas.CulvertSet	114
8.6.2.1	Statische Klassenfunktionen	114
8.6.2.2	Objektmethoden	115
8.6.3	Klasse hydroas.Element	118
8.6.3.1	Statische Klassenfunktionen	118
8.6.3.2	Objektmethoden	119
8.6.4	Klasse hydroas.Global	119
8.6.4.1	Statische Klassenfunktionen	119
8.6.5	Klasse hydroas.GrainFraction	124
8.6.5.1	Statische Klassenfunktionen	124
8.6.5.2	Objektmethoden	124
8.6.6	Klasse hydroas.InflowBC	125
8.6.6.1	Statische Klassenfunktionen	125
8.6.6.2	Objektmethoden	126
8.6.7	Klasse hydroas.Node	130
8.6.7.1	Statische Klassenfunktionen	130
8.6.7.2	Objektmethoden	132
8.6.8	Klasse hydroas.NodeSet	140
8.6.8.1	Statische Klassenfunktionen	141
8.6.8.2	Objektmethoden	142
8.6.9	Klasse hydroas.NodeString	147
8.6.9.1	Statische Klassenfunktionen	148
8.6.9.2	Objektmethoden	148
8.6.10	Klasse hydroas.OutflowBC	149
8.6.10.1	Statische Klassenfunktionen	149
8.6.10.2	Objektmethoden	150
8.6.11	Klasse hydroas.Types	155
8.6.11.1	Statische Klassenfunktionen	156
8.6.12	Klasse hydroas.Weir1D	157
8.6.12.1	Statische Klassenfunktionen	157
8.6.12.2	Objektmethoden	157
8.6.13	Klasse hydroas.WeirSet1D	159
8.6.13.1	Statische Klassenfunktionen	160
8.6.13.2	Objektmethoden	164

8.7	Lua	164
8.7.1	Zusatzmodul Lua-Filesystem	164
8.7.2	Zusatzmodul Lua-Date	167
8.7.3	Literatur und Referenzen	167
8.7.4	Lua-Programm-Verzeichnisse	168
9	Validierung	170
9.1	Strömung in einer trapezförmigen Rinne	170
9.2	Dammbruch 1D	172
9.3	Dammbruch 2D	173
9.4	Trapezkanal	173
10	Release-Notes	175
10.1	Version 5.3.1 (12.10.2021)	175
10.2	Version 5.3.0 (10.09.2021)	175
10.3	Version 5.2.5 (03.03.2021)	177
10.4	Version 5.2.4 (01.12.2020)	177
10.5	Version 5.2.3 (01.10.2020)	177
10.6	Version 5.2.2 (24.08.2020)	178
10.7	Version 5.2.1 (10.06.2020)	178
10.8	Version 5.2.0 (25.05.2020)	178
10.9	Version 5.1.9 (25.03.2020)	180
10.10	Version 5.1.8 (06.02.2020)	180
10.11	Version 5.1.7 (19.12.2019)	180
10.12	Version 5.1.6 (28.10.2019)	181
10.13	Version 5.1.5 (06.09.2019)	181
10.14	Version 5.1.4 (08.08.2019)	181
10.15	Version 5.1.3 (19.07.2019)	181
10.16	Version 5.1.2 (12.07.2019)	181
10.17	Version 5.1.1 (24.06.2019)	182
10.18	Version 5.1.0 (08.05.2019)	182
10.19	Version 5.0.2 (14.12.2018)	182
10.20	Version 5.0.1 (08.11.2018)	183
10.21	Version 5.0.0 (24.10.2018)	183
10.22	Version 4.4.7 (25.06.2018)	184
10.23	Version 4.4.6 (13.06.2018)	184
10.24	Version 4.4.5 (17.05.2018)	184
10.25	Version 4.4.4 (03.05.2018)	184
10.26	Version 4.4.3 (16.03.2018)	184
10.27	Version 4.4.2 (24.01.2018)	184
10.28	Version 4.4.1 (19.01.2018)	184
10.29	Version 4.4.0 (06.12.2017)	185
10.30	Version 4.3.4 (05.09.2017)	185
10.31	Version 4.3.3 (28.08.2017)	185
10.32	Version 4.3.2 (24.08.2017)	185
10.33	Version 4.3.1 (18.08.2017)	186
10.34	Version 4.3.0 (08.08.2017)	186
10.35	Version 4.2.7 (26.06.2017)	186
10.36	Version 4.2.6 (15.05.2017)	187
10.37	Version 4.2.5 (04.04.2017)	187
10.38	Version 4.2.4 (08.12.2016)	187
10.39	Version 4.2.3 (02.11.2016)	187

10.40	Version 4.2.2 (11.10.2016)	187
10.41	Version 4.2.1 (01.09.2016)	187
10.42	Version 4.2.0 (22.08.2016)	188
10.43	Version 4.1.6 (23.04.2016)	188
10.44	Version 4.1.5 (13.04.2016)	188
10.45	Version 4.1.4 (11.02.2016)	188
10.46	Version 4.1.3 (26.01.2016)	188
10.47	Version 4.1.2 (07.01.2016)	189
10.48	Version 4.1.1 (15.10.2015)	189
10.49	Version 4.1.0 (15.10.2015)	189
10.50	Version 4.0.4 (30.07.2015)	190
10.51	Version 4.0.3 (23.07.2015)	190
10.52	Version 4.0.2 (18.05.2015)	190
10.53	Version 4.0.1 (23.04.2015)	190
10.54	Version 4.0.0 (10.02.2015)	190
Literatur		192

Abbildungsverzeichnis

2.1	Systemskizze	3
2.2	Skizze: 1D-Wehr	8
2.3	Skizze: 1D-Durchlass	8
3.1	Kontrollvolumen (KV) für Knoten ij	12
4.1	lineares Netz	14
4.2	quadratisches Netz	14
4.3	Global Parameters Dialog	15
4.4	Dialog zur Konfiguration der Materialbereiche	18
4.5	Material Properties Dialog	19
4.6	Vorgabe des Strickler-Werts in SMS als wassertiefenabhängiger Wert	20
4.7	Vorgabe von Randbedingungen über den Dialog Assign BC	21
4.8	Dialog zur Eingabe der Randbedingungen an Nodestrings	22
4.9	Define Curve Dialog mit Zuflussganglinie für eine stationäre Berechnung	23
4.10	Zulauf gebunden an Auslauf - Skizze zur Modellierung	23
4.11	Zulauf gebunden an Auslauf - Vorgabe des Namens in SMS	24
4.12	Zulauf gebunden an Auslauf - Vorgabe der Koordinaten in SMS	24
4.13	Zulauf gebunden an Auslauf - prinzipielle Darstellung mit Materialtypen	25
4.14	Auslauf/le	25
4.15	Definition einer W-Q-Beziehung	26
4.16	Auslauf W/Q-Wehr: Pegelpunkt per Name angeben	26
4.17	Auslauf W/Q - Wehr: Pegelpunkt per Koordinaten angeben	27
4.18	Eingabe der H-Wehr (Zeit) Randbedingung in SMS	28
4.19	Wehrtypen für den unvollkommenen Überfall	29
4.20	Angabe des Unterwasser-Pegelpunktes per Name	29
4.21	Angabe des Unterwasser-Pegelpunktes per Koordinaten und zusätzliche Angabe der Wehrhöhe für ein Standardprofil	30
4.22	Eingabe der H-Wehr (Steuerung) Randbedingung in SMS	31
4.23	Beispiel einer Wehrsteuerung bei einem Hochwasser (inkl. Vorabsenkung)	32
4.24	Durchlass / Rechteck	33
4.25	Durchlass / kreisförmig	33
4.26	Parameter für die Randbedingung Wehrüberfall	35
4.27	Anordnung mehrerer Kontrollquerschnitte nebeneinander	36
4.28	Definition einer KUK	38
4.29	Nodel Bounday Conditions in den Disply Options	39
4.30	Markierung der Knoten mit KUK im Brückenbereich	39
4.31	Eingabe der Niederschlagsreihen	40
4.32	TreeView in HDFView	44
4.33	Datei in HDFView neu laden	45
4.34	Einem Knoten wird die Kennung SchachtA zugeordnet	46
4.35	Beispiel: See im 2D-Modell	48
4.36	Scatter Set um den See	49
4.37	Dialog zur Interpolation von Scatter Informationen	49
4.38	Nutzung des Data Calculators	50
4.39	Parameter <i>Eingabedateien binär</i>	51
4.40	Dialog Mesh Quality	52
4.41	Menü nach dem Aufruf des Save as Dialogs	54
5.1	Benutzerdefinierte Dateinamen	60
5.2	Einstellung von 6 Nachkommastellen in der Ausgabedatei q_strg.dat	62
5.3	Optionen für das Schreiben der cpr-Dateien in SMS	63

5.4	Auswahl <i>Rechenlauf fortsetzen?</i> in SMS	65
5.5	Fortsetzen eines Rechenlaufs	66
6.1	Startbildschirm der Benutzeroberfläche	70
6.2	Gruppierung von Bedienelementen	71
6.3	Liste zuletzt geöffneter Dateien	72
6.4	Allgemeine Bedienelemente	72
6.5	Seite Importprotokoll	73
6.6	Filtern von Meldungen	73
6.7	Kopieren und Speichern der Meldungsliste	73
6.8	Seite Prüfungen	74
6.9	Einstellungen zu den Prüfungen des Modells	75
6.10	Reiter Prüfeinstellungen in SMS	76
6.11	Meldungen mit Ortsbezug	76
6.12	SMS Werkzeug „Get Attributes“	77
6.13	Zuweisen einer Attributspalte als Name eines Observation Coverages	78
6.14	Seite Simulation	78
6.15	Startparameter der Simulation	79
6.16	Weitere Schaltflächen nach dem Simulationsstart	80
6.17	Rückfrage Simulation abbrechen	80
6.18	Debug-Ausgabe ausschalten	80
6.19	Seite Exportieren	81
6.20	MapView-Datensatz erzeugen	83
7.1	Gewässerquerschnitt in JabPlot	90
7.2	Längsschnitt in JabPlot	90
7.3	Modellansicht mit Wassertiefe in JabPlot	91
7.4	3D-Ansicht in JabPlot	92
7.5	Ergebnisdarstellung in MapView	93
7.6	MapView-Datensatz als Schnittstelle	93
8.1	Ausschnitt des Beispielmodells HAScripting	98
8.2	Plot der Wassertiefe am Polderauslass	101
8.3	Global Parameters: Zeitintervall Q_Strg [s]	105
8.4	1D-Wehrelemente zu Abbildung eines Wehres	159
8.5	Namensschema: CreekWeir_.*	160
9.1	Ermittelte Fließgeschwindigkeit im Trapezquerschnitt [30]	170
9.2	Ermittelte Schubspannung im Trapezquerschnitt [30]	170
9.3	Berechnungsnetz und ermittelte Schubspannungsverteilung	172
9.4	Ermittelte tiefengemittelte Fließgeschwindigkeiten im Trapezquerschnitt	172
9.5	Ermittelte Schubspannungen im Trapezquerschnitt	172
9.6	Validierungsbeispiel Dammbbruch 1D	173
9.7	Stationärer Wasserspiegel im Trapezkanal mit HYDRO_AS-2D berechnet	174
9.8	Vergleich der Wassertiefen im Trapezkanal	174

Tabellenverzeichnis

4.2	Unterschiede der Abflussberechnung an Kontrollquerschnitten in SMS und in HYDRO- _AS-2D	36
5.2	ASCII-Dateien (.dat) /XMDF-Dateien (.h5) zum Einlesen in SMS	60
6.1	Alle von MapView ausgewerteten Ausgabedateien	82

1 Einführung

1.1 Über das Benutzerhandbuch

Das vorliegende Benutzerhandbuch beschreibt das zweidimensionale Strömungsmodell HYDRO_AS-2D. Die Programmbedienung (Oberfläche) erfolgt dabei mit der Software SMS (Surface-water Modeling System). Die Oberfläche von SMS ermöglicht einen leichten Umgang mit verschiedenen Datenformaten, eine schnelle Datenvisualisierung und -prüfung, eine komfortable Netzbearbeitung und eine interaktive Dateneingabe. Die SMS-Programmoberfläche wird von Aquaveo (Provo, Utah, USA) gepflegt und ständig erweitert bzw. dem neuesten Entwicklungsstand angepasst.

1.2 HYDRO_AS-2D: Anwendungsbereiche, Möglichkeiten und Grenzen

Zweidimensionale hydrodynamisch-numerische Simulationsmodelle sind ein unentbehrliches Hilfsmittel für verschiedene wasserwirtschaftliche Untersuchungen im Bereich natürlicher Fließgewässer geworden. Deren Einsatz ist sehr vielfältig und erstreckt sich von der Berechnung der Flut- und Dammbruchwellenausbreitung über die Strömungs- und Abflusssimulationen bei Hoch- aber auch Niedrigwasser bis hin zur Berechnung des Sediment- und des Schadstofftransportes. Die zweidimensionale Simulation soll in Abhängigkeit von der Aufgabenstellung u.a. Aufschluss über folgende Punkte geben:

- Überschwemmungsgrenzen, Überflutungsdauer, Strömungsgeschwindigkeiten, Wassertiefen
- Abflussaufteilung Flussschlauch/ Vorländer,
- Retentionswirkung, Sohlschubspannungen,
- eventuell auch Schwebstoffablagerung, Geschiebetransport oder Wärmetransport (Zusatzmodule).

Das Strömungsmodell HYDRO_AS-2D wurde schwerpunktmäßig für die Berechnung von Damnbrüchen und zur Flutwellenausbreitung entwickelt, kann jedoch genauso erfolgreich für die allgemeine zweidimensionale Strömungssimulationen eingesetzt werden. Außerdem gibt es Module für die Berechnung von Geschiebe und Schwebstoffen (HYDRO_FT-2D) und Wärmetransport (HYDRO_WT-2D).

Eine wichtige Voraussetzung für die erfolgreiche Modellierung mittels eines zweidimensionalen Modells ist, dass das zu analysierende Problem eine zweidimensionale Natur besitzt, d.h., mit den Flachwassergleichungen gut modellierbar ist. So können z.B. die Strömungs- und Stofftransportvorgänge in einem tiefen See mit bedeutenden Temperaturunterschieden über die Wassertiefe, bei einigen Fragestellungen oft nicht mit den zweidimensionalen tiefengemittelten Strömungsgleichungen ausreichend genau beschrieben werden. Dies vor allem deswegen, weil die vertikale Geschwindigkeitskomponente für den zu simulierenden Prozess eine wesentliche Rolle spielt und bei der gegebenen Fragestellung nicht vernachlässigt werden darf. In einem solchen Fall sollte ein Modell einer dreidimensionalen Simulation zum Einsatz kommen.

Die Frage der Anwendbarkeit von HYDRO_AS-2D ist folglich mit der Frage der Anwendbarkeit der Flachwassergleichungen beantwortet. In der Praxis können viele praktische Probleme aus der Wasserwirtschaft erfolgreich mittels 2D-tiefengemittelten Strömungsgleichungen modelliert werden. Dies haben bisherige Erfahrungen beim Einsatz von 2D-Modellen gezeigt. So können z.B. zum Teil äußerst komplexe Abflussverhältnisse in überströmten Flusstälern effektiv mittels dieser Strömungsgleichungen modelliert werden. Es empfiehlt sich in diesem Zusammenhang weitere Referenzen zu studieren [10, 25, 1, 34], um mehr Sicherheit beim praktischen Einsatz des 2D-Modells zu erlangen.

2 Hydromechanische Gleichungen

2.1 2D-Strömungsgleichungen

Ausgangspunkt für die zweidimensionale mathematische Modellierung sowohl von Strömungsvorgängen in natürlichen Fließgewässern als auch für die Wasserspiegellagenberechnung und Flutwellenausbreitung sind die 2D-tiefengemittelten Strömungsgleichungen [1], die als Flachwassergleichungen (FWG) bekannt sind. Diese entstehen durch die Integration der dreidimensionalen Kontinuitätsgleichung und der Reynolds bzw. Navier-Stokes-Gleichungen für inkompressible Fluide über die Wassertiefe und unter Annahme einer hydrostatischen Druckverteilung [29, 35].

In kompakter Vektorform lauten die 2D-Strömungsgleichungen [33, 25]:

$$\frac{\partial \mathbf{w}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} + \mathbf{s} = 0 \quad (1)$$

wobei

$$\mathbf{w} = \begin{bmatrix} H \\ uh \\ vh \end{bmatrix} \quad (2)$$

$$\mathbf{f} = \begin{bmatrix} uh \\ u^2h + 0,5gh^2 - vh \frac{\partial u}{\partial x} \\ uvh - vh \frac{\partial v}{\partial x} \end{bmatrix} \quad (3)$$

$$\mathbf{s} = \begin{bmatrix} 0 \\ gh(I_{Rx} - I_{Sx}) \\ gh(I_{Ry} - I_{Sy}) \end{bmatrix} \quad (4)$$

$$\mathbf{g} = \begin{bmatrix} vh \\ uvh - vh \frac{\partial u}{\partial y} \\ v^2h + 0,5gh^2 - vh \frac{\partial v}{\partial y} \end{bmatrix} \quad (5)$$

Hierbei bezeichnet $H = h + z$ den Wasserspiegel über einem Bezugsniveau, u und v sind die Geschwindigkeitskomponenten in x - und y -Richtung.

Der Quellterm \mathbf{s} beinhaltet Ausdrücke für das Reibungsgefälle \mathbf{I}_R (mit den Komponenten I_{Rx} und I_{Ry}) und für die Sohlenneigung (I_{Sx} , I_{Sy}).

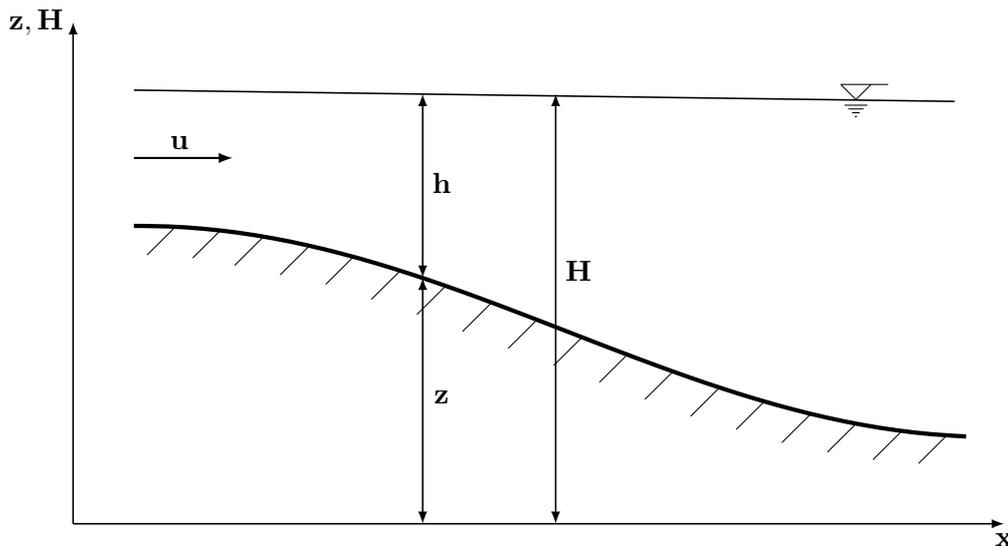


Abbildung 2.1: Systemskizze

Die Sohlennigung in x- und in y-Richtung ist durch den jeweiligen Gradienten des Sohlenniveaus z definiert:

$$I_{Sx} = \frac{-\partial z}{\partial x}, \quad I_{Sy} = \frac{-\partial z}{\partial y} \quad (6)$$

2.1.1 Rauheit

Die Berechnung des Reibungsgefälles erfolgt nach dem Darcy-Weisbach Ansatz:

$$I_R = \frac{\lambda v |v|}{2gD} \quad (7)$$

Die Bestimmung des Widerstandsbeiwertes λ erfolgt nach dem Manning-Strickler-Ansatz:

$$\lambda = 6,34 \frac{2g n^2}{D^{1/3}} \quad (8)$$

Hierbei ist n der Manning-Reibungskoeffizient als Kehrwert des Strickler-Beiwertes, g ist die Erdbeschleunigung und $D = 4 r_{hy}$ ist der hydraulische Durchmesser. Bei den 2D-Flachwassergleichungen wird der hydraulische Radius r_{hy} gleich der Wassertiefe h gesetzt.

Anmerkung: Für die Bestimmung des Widerstandsbeiwertes λ existieren auch andere Formeln wie z.B. die von Prandtl-Colebrook [31]. Im hydraulisch rauen Bereich, wie es häufig der Fall bei der Modellierung von natürlichen Fließgewässern ist, dürfte dennoch der zwischen den nach verschiedenen Ansätzen ermittelten Werten vorhandene Unterschied gering bleiben [25].

2.1.2 Viskosität

Es gibt verschiedene Wege, die Viskosität zu definieren: den Ansatz einer (abschnittsweise) konstanten Viskosität [19], einen algebraischen bzw. empirischen Viskositätsansatz [11] sowie ein (k-e) Turbulenzmodell [2].

Trotz neuer Entwicklungen bei der Turbulenzmodellierung existieren z.B. für die Simulation der Ausbreitung einer Dammbrech- / Flutwelle noch keine zuverlässigen und überprüften Turbulenzmodelle. Deswegen wird immer noch ein großer Teil von tiefengemittelten Strömungen mit einer konstanten Viskosität numerisch simuliert. Außerdem zeigt die Erfahrung, dass bei einer sorgfältigen Optimierung mit dem Ansatz einer (abschnittsweise) konstanten Viskosität vergleichbare Ergebnisse wie mit einem Turbulenzmodell erzielt werden können. Dies hängt u.a. damit zusammen, dass die Viskosität bei den meisten praktischen Anwendungen und insbesondere bei der Dammbrech- / Flutwellenausbreitung eine untergeordnete Rolle spielt. Infolge dessen kann sie meist anhand empirischer Ansätze, in Abhängigkeit davon, welcher Mechanismus für den turbulenten Transport des Impulses im zugeordneten Bereich dominiert, gut abgeschätzt werden. Aus diesen Gründen wurde in HYDRO_AS-2D eine Kombination aus dem empirischen Viskositätsansatz und dem Ansatz einer über das Element konstanten Viskosität gewählt.

Die Viskosität in m^2/s wurde somit anhand folgender Formel definiert:

$$\nu = \nu_0 + c_\mu u_* h \quad (9)$$

wobei ν_0 (in m^2/s) einen abschnittsweise konstanten Wert darstellt (kann in SMS für jedes einzelne Element vorgegeben werden).

Der zweite Summand (auf der rechten Seite) stellt die durch Sohlreibung hervorgerufene Wirbelviskosität in Abhängigkeit von der Schubspannungsgeschwindigkeit u_* und von der Wassertiefe h dar. Für den Koeffizienten c_μ wurde anhand von Versuchen im Bereich natürlicher und naturnaher Fließgewässer ein Wert zwischen 0,3 und 0,9 ermittelt. Im Programm wurde standardmäßig ein mittlerer Wert von 0,6 eingesetzt.

Anmerkung: Bei der Abflussmodellierung spielt die Viskosität meist eine untergeordnete Rolle. Deswegen genügt es oft, $\nu_0 = 0$ einzusetzen, ohne bemerkenswerte Fehler dabei zu verursachen. Werte größer als Null (ν_0 zwischen 0,1 und 2) sind vor allem dort erforderlich, wo es zu einem stärkeren Durchmischen kommt sowie in den Bereichen mit bedeutenden Scherströmungen.

2.1.3 Anfangs- und Randbedingungen

Flachwassergleichungssysteme müssen – aus mathematischer Sicht – zusätzlich noch die Anfangs- und die Randbedingungen spezifiziert werden. In diesem Zusammenhang stellt sich die Frage der Zulässigkeit bzw. der Stabilität von verwendeten Randbedingungen. Zu viele bzw. zu wenig oder falsch

spezifizierte Randbedingungen können zu einem mathematisch schlecht konditionierten Problem führen, das folglich keine realistische Lösung mehr erlaubt. Dies macht sich durch unplausible oder gar stark fluktuierende Größen (Wassertiefen und/oder Geschwindigkeiten) im Bereich von Randbedingungen bemerkbar.

Randbedingungen

Aus praktischer Sicht sollen die definierten Randbedingungen relativ einfach, allerdings ausreichend genau und sehr stabil sein. Um die gegebenen Anforderungen zu erfüllen, ist es vorteilhaft, die Zu- und Auslaufränder an den Modellrand zu verlegen.

Hinweis: Achten Sie bei der Ermittlung bzw. Beschaffung von Geländedaten auf die im Modell zusätzlich erforderliche Ein- und Auslaufstrecken ober- bzw. unterhalb des Untersuchungsraums.

Es ist zwischen folgenden zwei grundsätzlichen Randtypen zu unterscheiden:

- geschlossene Ränder
- offene Ränder

Bei den sog. geschlossenen Rändern kann die Strömung parallel zum Rand erfolgen. D.h., die senkrechte Geschwindigkeitskomponente ist gleich null und diese Ränder werden infolgedessen nicht durchströmt. Dies ist z.B. bei einer hohen Wand der Fall. Alle Ränder, die nicht als Zu- und/ oder als Auslauf spezifiziert sind, werden vom Programm automatisch als geschlossene Ränder definiert.

Hinweis: Die tangentielle Spannung wird in HYDRO_AS-2D an geschlossenen Rändern auf null gesetzt, was der sog. slip-velocity Randbedingung entspricht. Der infolge nicht berücksichtigter Spannungen aufgetretene Modellierungsfehler bleibt bei praktischen Anwendungen i.d.R. meist sehr gering. Der Einfluss von vernachlässigten Spannungen auf die Strömung kann dann, falls erforderlich, über ein erhöhte Sohlrauheit in den Randzellen approximativ im Modell berücksichtigt werden.

Die sog. offenen Ränder werden im Gegensatz zu den geschlossenen Rändern durchströmt. Unter diese Kategorie fallen insbesondere die Zulaufränder, die Auslaufränder und die Ränder für die eine sog. H-Randbedingung spezifiziert ist.

Die Auswahl der im Programm möglichen Randbedingungen ist in [Kapitel 4.5](#) beschrieben.

Anfangsbedingungen

Das Spezifizieren einer Anfangsbedingung ist weniger problematisch und kann über die Vorgabe eines Anfangsströmungszustandes (Wassertiefe, Fließgeschwindigkeit) erfolgen, vgl. [Kapitel 4.7](#) und [Kapitel 4.8](#). Dennoch soll beachtet werden, dass der anfängliche Strömungszustand sinnvoll bzw. realistisch zu definieren ist.

2.2 1D - Bauwerke

Innerhalb des Untersuchungsgebiets befinden sich oft Bauwerke, die bei der Abflussberechnung zu berücksichtigen sind.

In HYDRO_AS-2D können Bauwerke durch 1D-Elemente dargestellt werden. Für diese 1DElemente werden empirische Berechnungsansätze zur Abflussbestimmung eingesetzt.

Für die Abflussmodellierung werden folgende zwei Bauwerkstypen unterschieden:

- überströmte Bauwerke
- durchströmte Bauwerke

2.2.1 Überströmte Bauwerke

Die Berechnung von **überströmten** Bauwerken (Wehre, Überfälle, Schwellen, Sohlenrampen, überströmte Straßendämme und Deiche, etc.) erfolgt über die Überfallformel (BWK, 1999) für den Abfluss Q in m^3/s nach Du Buat:

$$Q = \frac{2}{3} c_s \cdot \mu \cdot b \cdot \sqrt{2g} \cdot \left(h_{ow} + \frac{v^2}{2g} \right)^{3/2} \quad (10)$$

Der Abminderungsfaktor wird dabei wie folgt berechnet:

$$c_s = \sqrt{1 - \left(\frac{h_{uw}}{h_{ow}} \right)^{16}} \quad (11)$$

Es gibt die Möglichkeit in der Berechnung der Geschwindigkeitshöhe die Richtung der Strömung im Oberwasser im Verhältnis zum Wehr zu berücksichtigen, siehe [Kapitel 4.5.11](#). Bei genau senkrechter Anströmung des Wehrkörpers wird die Geschwindigkeit komplett berücksichtigt. Bei nicht senkrechter Anströmung wird der Anteil entsprechend reduziert, bis die Geschwindigkeitshöhe bei einer zum Wehrkörper parallelen Strömung auf null gesetzt wird. Ist die Strömungsrichtung vom Wehr weg gerichtet, wird die Energiehöhe entsprechend verringert.

Variable	Einheit	Beschreibung
h_{ow}	m	Überfallhöhe in Bezug auf die Kronenhöhe
v	m/s	Geschwindigkeit über dem Wehrkörper
μ	-	Überfallbeiwert
c_s	-	Abminderungsfaktor für unvollkommenen Überfall
b	m	Abflusswirksame Breite des Bauwerks
g	m/s^2	Gravitationskonstante (9,81 m/s^2)
h_{uw}	m	Höhe relativ zur Kronenhöhe im Unterwasser

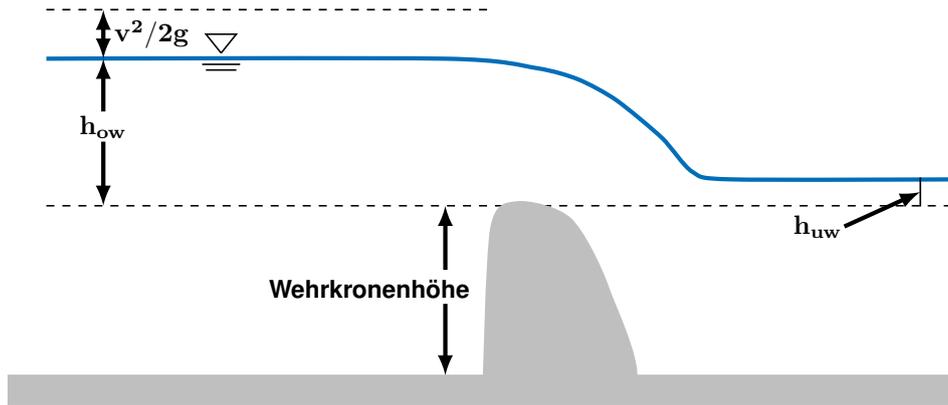


Abbildung 2.2: Skizze: 1D-Wehr

2.2.2 Durchströmte Bauwerke

Für die Abfluss-Berechnung von **durchströmten** Bauwerken (Durchlässe, Brückenöffnungen, Flutöffnungen, etc.) wird folgende Formel für den Abfluss Q in m^3/s angesetzt:

$$Q = c \cdot b \cdot h \cdot \sqrt{2g \cdot \left(\Delta H + \frac{v^2}{2g} \right)} \quad (12)$$

Variable	Einheit	Beschreibung
c	-	Abflussfaktor
b	m	Abflusswirksame Breite des Bauwerks (entsprechende Umrechnung bei kreisförmigen Durchlässen)
h	m	Fließtiefe im Bauwerk, maximal Durchlasshöhe
ΔH	m	Druckhöhe als Differenz der Wasserspiegellagen im Oberwasser und Unterwasser
v	m/s	Geschwindigkeit im Durchlass
g	m/s^2	Gravitationskonstante ($9,81 \text{ m/s}^2$)

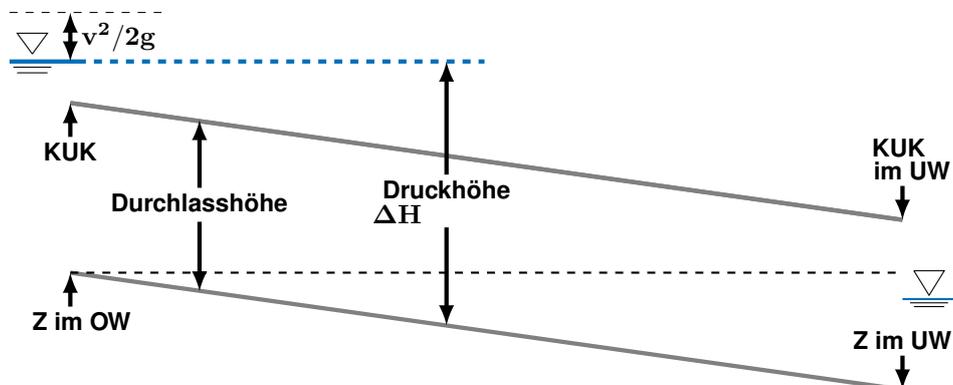


Abbildung 2.3: Skizze: 1D-Durchlass

In den Berechnungen von h und ΔH werden verschiedene Einflüsse auf den Abfluss berücksichtigt:

- Geschiebeanlandung (nur bei GS, FT, ST, Steuerung relevant): Für Einlauf und Auslauf wird jeweils überprüft, ob die Sohle des zugehörigen Netzknotens höher als die angegebene Sohle des Bauwerks liegt. Ist das der Fall, wird die Sohlhöhe des Netzknotens als Sohlhöhe des Bauwerkseinlaufs oder -auslaufs verwendet. Dadurch verringern sich Fließtiefe und ggf. Druckhöhe.
- Rohrneigung: Als Referenz für die Druckhöhe wird das Maximum der Sohlhöhen des Bauwerkseinlaufs und des Bauwerksauslaufs verwendet, wenn der Durchlass nicht von unten eingestaut ist.
- Einstau des Bauwerks im Unterwasser: Bei Einstau wird der Unterwasserstand in der Berechnung der Druckhöhe berücksichtigt.
- Druckabfluss wird durch ΔH berücksichtigt. Zur Bestimmung von ΔH wird der Wasserspiegel im Oberwasser benutzt, der höher sein kann als die KUK des Durchlasses im Oberwasser.
- Die Fließtiefe h kann nicht höher werden als die Durchlasshöhe. Damit entspricht der durchströmte Querschnitt $b \cdot h$ maximal dem Bauwerksquerschnitt.
- Geschiebeanlandung im Unterwasser (nur bei GS, FT, ST, Steuerung relevant): Die maximale Öffnungshöhe im Unterwasser kann durch Geschiebeanlandung reduziert sein. Die Fließtiefe h kann maximal der Öffnungshöhe im Unterwasser entsprechen. Dadurch wird der Abfluss ggf. verringert. Der Durchlass kann auch komplett verlegt sein.

Es gibt die Möglichkeit in der Berechnung der Geschwindigkeitshöhe die Richtung der Strömung im Oberwasser im Verhältnis zum Durchlass zu berücksichtigen, siehe [Kapitel 4.5.10](#). Ist die Strömungsrichtung im Oberwasser parallel zur Ausrichtung des Durchlasses, wird die Geschwindigkeit komplett berücksichtigt. Bei nicht parallel verlaufender Strömung wird der Anteil entsprechend reduziert, bis die Geschwindigkeitshöhe bei einer senkrechten Strömung auf null gesetzt wird. Ist die Strömungsrichtung vom Durchlass weg gerichtet, wird die Energiehöhe entsprechend verringert.

Es ist im Allgemeinen schwierig, bei verschiedenen Bauwerkstypen das gesamte Abflussregime mithilfe empirischer Ansätze richtig abzudecken. Es empfiehlt sich deshalb eine Durchflusskontrolle mit genauerer Parameteranpassung durchzuführen oder Literaturwerte zur Validierung heranzuziehen.

2.2.3 Abflusskoeffizient

Rechteckige und kreisförmige Durchlässe lassen sich über vorher definierte Nodestrings mit der eindimensionalen Durchlassformel berechnen. Als Parameter muss unter anderem der Abflusskoeffizient c angegeben werden (siehe auch [Kapitel 4.5.10](#)). Der Abflusskoeffizient beschreibt die Kombination der Fließwiderstände bzw. Fließverluste der gesamten Verrohrung.

Geeignete Werte liegen im Allgemeinen zwischen 0,5 und 0,8. Eine plausible Näherung liefert auch die folgende Formel:

$$c = \frac{1}{\sqrt{k_e + k_o + \frac{2gL}{(K_{St}^2 \cdot R^{4/3})}}} \quad (13)$$

Variable	Einheit	Beschreibung
k_e	-	Einlaufverlustbeiwert, z. B. 0,3
k_o	-	Auslaufverlustbeiwert, z. B. 1,0
L	m	Durchlass-/Rohrlänge
K_{St}	$m^{1/3}/s$	Rauheitsbeiwert nach Strickler
R	m	hydraulischer Radius (bei Vollfüllung)
g	m/s^2	Gravitationskonstante (9,81 m/s^2)

2.2.4 Anwendungskriterien der 1D-Bauwerke

Folgende Kriterien sind bei der Modellierung mit 1D-Bauwerken zu beachten:

- 1D-Bauwerke werden eingesetzt, wenn die hydraulisch maßgebenden Bauwerksabmessungen deutlich kleiner als das Untersuchungsgebiet sind.
- Die 1D-Bauwerke bestimmen den Abfluss anhand der oben beschriebenen Ansätze. Translation/Fließzeit und Retention einer Welle innerhalb der Bauwerke werden nicht berücksichtigt. Daher sollten die Nodestings für 1D-Bauwerke - vor allem bei instationären Vorgängen - nicht länger als 50 m sein.
- Der Einlaufbereich sollte nicht mit einer markanten Beschleunigungsstrecke zusammenfallen. In diesem Fall sollte das Bauwerk direkt zweidimensional modelliert werden.
- Als Kriterium für die Plausibilisierung der Anwendung der Durchlassformel kann die Froude-Zahl dienen. Diese sollte kleiner als 0,80 bleiben.
- Es kann in Abhängigkeit des Verhältnisses von Elementbreite zur Breite des 1D-Bauwerks sinnvoll sein, mehrere Nodestings nebeneinander anzuordnen. In diesem Fall ist der Durchlassquerschnitt bzw die Breite des Wehres auf die einzelnen Nodestings aufzuteilen.
- Die Flächen der Kontrollvolumen an Einlauf und Auslauf eines Bauwerks sollten in derselben Größenordnung liegen. Vor allem sind wesentlich kleinere Kontrollvolumen am Auslaufknoten zu vermeiden.

3 Numerisches Lösungsverfahren

3.1 Räumliche und zeitliche Diskretisierung

Für die Durchführung der numerischen Simulation ist eine Aufteilung des Gesamtgebiets in eine bestimmte Anzahl von diskreten Elementen erforderlich (Diskretisierungsvorgang). Das in HYDRO_AS-2D integrierte Verfahren basiert auf der numerischen Lösung der 2D-tiefengemittelten Strömungsgleichungen mit der räumlichen Diskretisierung nach der Finite-Volumen-Methode.

Bei der Finite-Volumen-Methode werden die Erhaltungsgleichungen (Flachwassergleichungen) in integraler Form verwendet. Durch Integration über das Kontrollvolumen V und unter Verwendung des Integralsatzes von Gauß erhält man folgende Darstellung:

$$\frac{\partial}{\partial t} \int_V w dV = - \oint_{\partial V} (\vec{f}, \vec{g}) \cdot \vec{n} d\partial V - \int_V s dV \quad (14)$$

Die approximative Lösung des Flachwassergleichungssystems zum Zeitpunkt $n \Delta t$ und für die Zelle (ij) wurde als w_{ij}^n bezeichnet, w_{ij}^{n+1} ist dementsprechend die approximative Lösung zum nächsten Zeitpunkt. Diese approximative Lösung kann in folgender predictor-corrector Form geschrieben werden:

$$w_{ij}^p = w_{ij}^n - \Delta t R(w_{ij}^n) \quad (15)$$

$$w_{ij}^{n+1} = 0,5 \left[w_{ij}^n + w_{ij}^p - \Delta t R(w_{ij}^p) \right] \quad (16)$$

Die gewählte Zeitdiskretisierung entspricht somit dem Heun-Verfahren, welches ein explizites Runge-Kutta-Verfahren zweiter Ordnung ist. Das Residuum R aus der obigen Gleichung ist wie folgt definiert:

$$R(w_{ij}) = \frac{1}{A_{KV}} \sum_e \left[(\vec{f}, \vec{g}) \cdot \vec{n} dl \right]_e + s_{ij} \quad (17)$$

Variable	Beschreibung
A_{KV}	Fläche des Kontrollvolumens
dl	Länge der jeweiligen Zellseite e
\vec{n}	Normalenvektor für die jeweilige Zellseite e

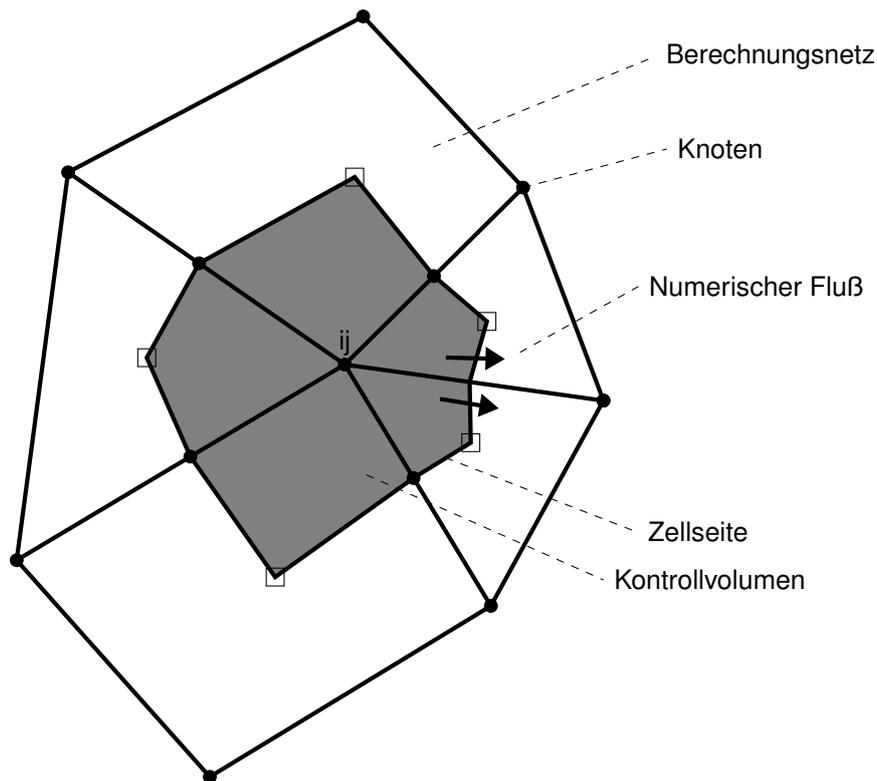


Abbildung 3.1: Kontrollvolumen (KV) für Knoten ij

3.2 Eigenschaften der in HYDRO_AS-2D verwendeten Methoden

Bei der Programmentwicklung wurden besonders hohe Anforderungen definiert und angestrebt, damit das Verfahren möglichst viele der verschiedenen mathematisch-physikalischen Eigenschaften der tiefengemittelten Strömungsgleichungen nahezu exakt beschreiben kann. Dadurch konnte die Genauigkeit der numerischen Lösung wesentlich verbessert werden. Des Weiteren wurde das im HYDRO_AS-2D eingesetzte Verfahren für die praktische Anwendung hinsichtlich der erforderlichen Genauigkeit, Robustheit und der Rechenzeit optimiert. Das in HYDRO_AS-2D implementierte Verfahren zeichnet sich u.a. durch sehr geringe Numerische- / Querdiffusion aus. Es können somit Wirbelbildung, Impulsaustausch zwischen dem Flussschlauch und dem Vorland, Wechselsprünge, Sohlensprünge, örtliche Verluste, Querschnittseinengungen und -aufweitungen, Strömung unter Druckabfluss, steile Geländeneigungen (Wildbäche) etc. modelliert werden. Das Verfahren wurde u.a. durch die Teilnahme an internationalen Workshops wie CADAM neutral und objektiv getestet und validiert [24]. Beispiele finden Sie in [Kapitel 9](#).

HYDRO_AS-2D verwendet ein aus Vierecks- und Dreieckselementen zusammengesetztes Berechnungsnetz. Die Verwendung eines solchen Netzes ermöglicht u.a. eine leichte Anpassung an die topographischen und die hydrodynamischen Gegebenheiten der jeweiligen Aufgabenstellung. Damit können z.B. die Fließ-, Deich- und Wegeverläufe relativ einfach und vor allem genau erfasst werden, was für den zu modellierenden Strömungsprozess eine entscheidende Rolle spielen kann.

Die FV-Methode zeichnet sich durch ihre konservative Eigenschaft aus und ist deswegen für die Berechnung von diskontinuierlichen Übergängen (Wechselsprüngen, Sohlstufen, Querschnittsänderungen, etc.) besonders zu empfehlen. Des Weiteren treten infolgedessen keine Massendefizite, wie bei manchen anderen Verfahren, auf.

Die Strömungs- bzw. Abflussberechnungen werden mit HYDRO_AS-2D grundsätzlich instationär durchgeführt. Das eingesetzte explizite Zeitschrittverfahren sorgt an dieser Stelle für eine zeitgenaue Simula-

tion des Wellenablaufs. Die Zeitschrittlänge muss dabei das folgende Stabilitätskriterium (sog. Courant-Friedrichs-Lewy Stabilitätsbedingung) für jedes (verbundene) Knotenpaar des Berechnungsnetzes erfüllen:

$$\Delta t \leq \min \left(\frac{\Delta L}{|v| + \sqrt{gh}} \right) \quad (18)$$

Dabei ist ΔL die charakteristische Länge und v der Komponentenvektor der tiefengemittelten Strömungsgeschwindigkeit in Richtung der Kante (Knotenpaar-Verbindung). Die charakteristische Länge entspricht i.d.R. der Kantenlänge (= Abstand der Knoten). Nur am Modellrand wird ΔL auf die halbe Kantenlänge gesetzt.

Wie aus der obigen Formel ersichtlich, sollen zu kleine Knotenabstände möglichst vermieden werden, damit die Zeitschrittlänge nicht zu stark verringert wird.

Infolge kleiner Zeitschritte ist es jedoch möglich, die Grenze zwischen dem trockenen und dem unter Wasser stehenden Gelände besser zu ermitteln.

An der Grenze zwischen dem trockenen und dem unter Wasser stehenden Gelände können während der Berechnung häufig Schwierigkeiten bzw. Instabilitäten auftreten. Um dies zu vermeiden wird bei vielen numerischen Verfahren eine sog. minimale Wassertiefe eingeführt, d.h., die Punkte werden im Programm erst ab dieser Wassertiefe in die Berechnung einbezogen. In HYDRO_AS-2D kann die minimale Wassertiefe vom Nutzer gewählt werden. Der Standardwert ist 1 cm.

Anmerkung: Das Programm wurde vor allem für praktische Anwendungen konzipiert und optimiert. D.h., dass z.B. bei der Nachrechnung von Modellversuchen (in kleinem Maßstab) Anpassungen notwendig sind. Als minimale Wassertiefe kann in so einem Fall z.B. ein Wert von 0,1 mm eingesetzt werden.

3.3 1step und 2step

Es gibt zwei Versionen des Berechnungsmoduls:

- HYDRO_AS (hydro-as.exe; hydro_as) stellt die Grundversion des 2D-Modells dar, mit dem u.a. hoch instationäre Vorgänge wie z.B. bei einer Dammbrechwellenausbreitung erfolgreich simuliert werden können. Die gewählte Zeitdiskretisierung entspricht dem expliziten Runge- Kutta-Verfahren zweiter Ordnung (2step, vgl. [Kapitel 3.1](#)).
- HYDRO_AS-1STEP (hydro_as-1step.exe; hydro_as-1step) stellt eine Vereinfachung der Grundversion dar, indem die Zeitdiskretisierung nur die erste Ordnung an Genauigkeit besitzt. Die Berechnung wird infolge dessen etwa doppelt so schnell wie bei der Grundversion durchgeführt. Bedeutende Fehler können erst bei der Modellierung hoch instationärer Vorgänge wie z.B. bei einer Dammbrechwellenausbreitung auftreten. Bei weniger instationären Vorgängen wie z.B. bei einer Flutwellenausbreitung oder einer instationären (und insbesondere stationären) Abflussberechnung, führt der Einsatz von HYDRO_AS-1STEP i.d.R. zum gleichen Ergebnis wie bei der Grundversion.

4 Modellerstellung

4.1 Einführung

Für jedes neue Projekt wird empfohlen, vor der Modellerstellung ein neues Verzeichnis anzulegen. Ferner wird empfohlen, für jeden neuen Rechelauf einen neuen Ordner zu erstellen, damit die durchgeführten Berechnungen und Veränderungen am Modell effektiv zurückverfolgt werden können.

4.2 SMS

Die Programmbedienung (Oberfläche) erfolgt über das Programm SMS (Surface-water Modeling System). Die SMS-Programmoberfläche wird von Aquaveo (Provo, Utah, USA) gepflegt und ständig erweitert bzw. dem neuesten Entwicklungsstand angepasst.

Die Oberfläche von SMS ermöglicht einen leichten Umgang mit verschiedenen Datenformaten, eine schnelle Datenvisualisierung und -prüfung, eine komfortable Netzgenerierung und eine interaktive Dateneingabe.

HYDRO_AS-2D wurde an die Oberfläche von SMS gekoppelt, d.h., die von SMS erzeugten Ausgabedateien dienen HYDRO_AS-2D als Eingangsdaten.

Die Modellerstellung erfolgt demnach in SMS, wobei dazu zusätzliche Programme (z.B. ArcGIS, GeCAD, AutoCAD, etc.) eingesetzt werden können.

Bei der Netzerstellung ist zu beachten, dass der richtige Netztyp verwendet wird.

SMS unterstützt zwei verschiedene Netztypen: ein quadratisches sowie ein lineares Netz. Von HYDRO_AS-2D wird ausschließlich ein lineares Netz verwendet, da für die örtliche Diskretisierung ein Finite Volumen Verfahren zu Grunde liegt.

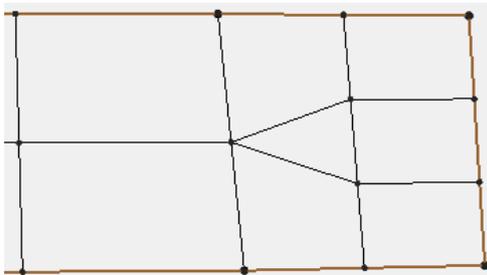


Abbildung 4.1: lineares Netz

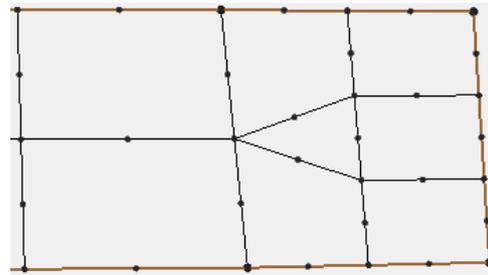


Abbildung 4.2: quadratisches Netz

Die Berechnungsergebnisse werden mit SMS eingelesen und dargestellt. Für SMS existiert ein ausführliches Benutzerhandbuch (in englischer Sprache).

Über die HYDRO_AS-2D-Vorlage.2dm wird ein in SMS erstelltes Modell an HYDRO_AS-2D gekoppelt. Die HYDRO_AS-2D-Vorlage.2dm enthält HYDRO_AS-2D-spezifische Parameter und Randbedingungen, die nach erfolgter Kopplung bereitgestellt werden.

Die HYDRO_AS-2D-Modellparameter werden in SMS im HYDRO_AS-2D Dialog Global Parameters angegeben.

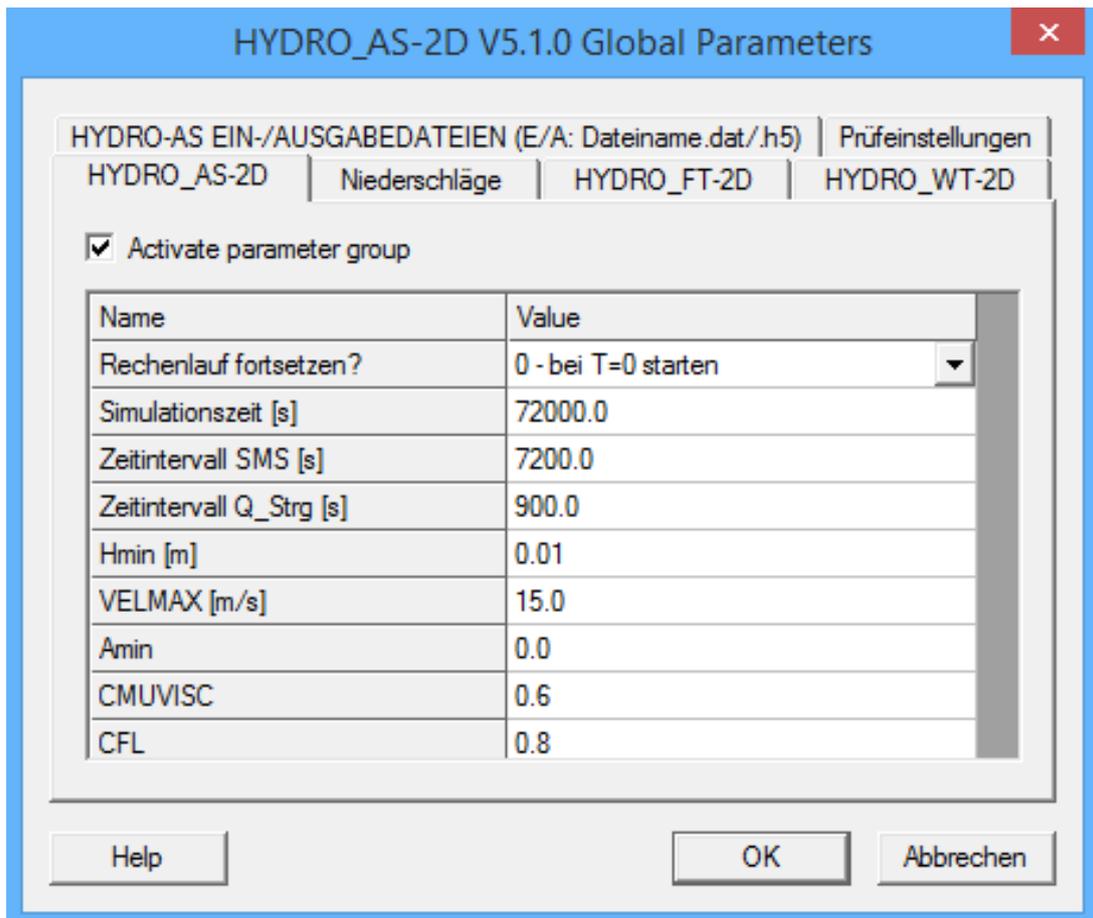


Abbildung 4.3: Global Parameters Dialog

Es folgt eine Erläuterung der einzelnen Parameter:

Parameter	Beschreibung	Default
Rechenlauf fortsetzen? vgl. auch Kapitel 5.8	legt fest, ob ein Rechenlauf fortgesetzt werden soll. Falls ja, werden die Ergebnisdaten aus der Datei hydro_as-2d.cpr in das Programm übernommen und als Anfangsbedingung eingesetzt. Der Rechenlauf kann zu einem beliebigen Zeitschritt oder beim letzten Zeitschritt fortgesetzt werden.	0 - bei T=0 starten
Simulationszeit [s]	definiert die Simulationszeit in Sekunden	72000 s
Zeitintervall SMS [s]	definiert das Zeitintervall (in Sekunden) zum Abspeichern von flächenhaften Ergebnisdaten, die im Anschluss der Berechnungen in SMS visualisiert werden können.	7200 s
Zeitintervall Q_Strg [s]	definiert das Zeitintervall (in Sekunden), für das die Daten der Nodestrings in die q_strg.dat geschrieben werden.	900 s
Hmin [m]	minimale Wassertiefe in m (Beginn der Benetzung)	0,01 m
VELMAX [m/s]	maximale erlaubte Geschwindigkeit in m/s	15 m/s
Amin	minimale zur Berechnung des internen Rechenzeitschritts erlaubte Elementgröße in m ² (vgl. Kapitel 5.10.5)	0 m ²

Parameter	Beschreibung	Default
CMUVISC	Koeffizient c_{μ} in der Formel für die Viskosität, vergl. Gleichung 9	0,6
CFL	Courant-Friedrichs-Lewy-Zahl. Faktor zwischen 0 und 1 für die interne Zeitschrittweite	0,8

Bemerkung: HYDRO_AS-2D ist ausschließlich für Angaben im metrischen System konzipiert.

4.3 Übernahme von topographischen Geländedaten und Netzerstellung

Die Grundlagendaten stellen einen wesentlichen Faktor bei der zweidimensionalen Strömungs- / Abflusssimulation dar und sind entscheidend für die Qualität der gesamten Modellierung. Die für die numerische Simulation notwendigen Geländeinformationen können aus folgenden Daten zusammengesetzt werden:

- einem punktuellen bzw. regelmäßigen Raster DGM (z.B. 1 m x 1 m Laserscan-Daten)
- Bruchkanten für die Uferlinien, Dammverläufe, Querbauwerke, etc.
- Querprofilaufnahmen der Fließgewässer (aquatischer Bereich)
- Lageplänen
- Luftbildaufnahmen
- Bestands- und Detailplänen über die vorhandenen Brücken, Wehre, Sohlenstufen, Durchlässe und weitere strömungsrelevante Bauwerke
- Flächennutzung zur Festlegung der Rauheit
- Gebäudepolygone

HYDRO_AS-2D verwendet ein aus Vierecks- und Dreieckselementen zusammengesetztes Berechnungsnetz. Dies ermöglicht u.a. eine leichtere Anpassung an die topographischen und hydrodynamischen Gegebenheiten der jeweiligen Aufgabenstellung.

Eine voll automatisierte Bildung des Berechnungsnetzes z.B. durch angepasste Delaunay- Triangulierung unter Berücksichtigung der vorhandenen Bruchkanten ist nur in Kombination mit speziellen Netzerstellungsprogrammen wie z.B. LASER_AS-2D [14] zu empfehlen. Sonst kann dadurch einerseits u.U. eine zu hohe Anzahl von Berechnungspunkten und andererseits aus numerischer Sicht für die Berechnung eine inadäquate Konfiguration der Netzelemente entstehen. Daraus können entweder numerische Instabilitäten oder Ungenauigkeiten in der numerischen Simulation resultieren. Es ist im Allgemeinen ein Berechnungsnetz anzustreben, dessen Form sich sowohl dem Strömungsverlauf als auch dem Geländeverlauf weitgehend anpasst. D.h. die Elemente werden so gebildet, dass alle wichtigen Bruchlinien (Ufer, Dämme, Straßen etc.) durch das Netz eingehalten werden.

Alle strömungsrelevanten Bereiche müssen aus numerischer Sicht genügend genau beschrieben werden. Gegebenenfalls ist eine Netzverfeinerung vorzunehmen.

Besonders wichtig für die Modellbildung ist die genaue Übertragung von topographischen Geländedaten aus einem DGM auf die Knotenpunkte des Berechnungsnetzes. Durch ungenaue Übertragung der Geländehöhen auf das Berechnungsgitter können erhebliche Fehler im Modell erzeugt werden. Dies trifft besonders für die sog. Bruchkanten (z.B. Böschungsoberkante eines Flussschlauches) zu, welche die entscheidende Rolle bei der Ausuferung spielen. Eine exakte Übernahme solcher Objekte in das Berechnungsnetz ist deshalb unbedingt anzustreben.

Alternativ zur automatischen Netzerstellung erfolgen die Übernahme von topographischen Daten in SMS und die Netzerzeugung in mehreren Schritten. SMS unterstützt dafür folgende Dateitypen:

- XYZ mit Rechtswert, Hochwert und Geländehöhe
- DXF-AutoCAD mit 3D-Bruchkanten
- GIS-Daten im shapefile-Format
- Hoch aufgelöste Laserscandaten der Geländeoberfläche

Diese Dateien können über File ⇒ Open in SMS geladen werden.

Die genaue Beschreibung des XYZ-Datenformates ist im SMS-Benutzerhandbuch gegeben.

Die Punkte aus der XYZ-Datei können dann im Mesh Modul unmittelbar weiter verwendet bzw. direkt vernetzt werden. Für die Vernetzung gibt es in SMS mehrere Hilfsmittel, von der Erzeugung einzelner drei- bzw. viereckigen Elemente bis hin zur automatischen Triangulierung des gesamten Netzes oder Teilen davon. Die verschiedenen Möglichkeiten dafür und die dazu erforderlichen Schritte sind im SMS-Benutzerhandbuch beschrieben.

Die Übernahme und die Weiterverarbeitung von Daten aus der DXF-Datei erfolgt über das Map Modul. Des Weiteren können diese Daten über das DXF Menü in Feature Objects oder in Scatter Points umgewandelt werden. Die als Feature Objects erhaltenen Daten können zusätzlich im Map Modul verarbeitet werden, bevor diese endgültig in das Mesh Modul übernommen werden. Gleichwohl können die Scatter Points aus dem Scatter Modul über Data ⇒ Scatter → Mesh zur Weiterverwendung in das Mesh Modul übernommen werden.

Hinweis:

Die Punktübernahme (x,y,z-Werte) kann aus Feature Objects nur über das Scatter Modul erfolgen:

1. Schritt (Map Modul): Feature Objects ⇒ Map → Scatter (Arc Nodes and Vertices)
2. Schritt (Scatter Modul): Data ⇒ Scatter → Mesh

Im Zusammenhang mit der Netzerstellung bietet SMS weitere Möglichkeiten:

- für die Netzkontrolle und Optimierung,
- zum Nacharbeiten und Korrigieren des Berechnungsnetzes sowie der Geländedaten und der Rauheit,
- durch das Hinterlegen von Bildern und topographischen Karten (TIFF-Format) zur Kontrolle und zur besseren Anpassung sowie zur Datenübernahme.

Detailliertere Informationen zum Einsatz von SMS bei der Datenübernahme und bei der Netzerstellung können dem SMS-Benutzerhandbuch entnommen werden.

4.4 Rauheit und Viskosität

Für die Durchführung der Berechnung muss die Verteilung der Rauheit sowie der Wirbelviskosität über das Modellgebiet flächendeckend definiert werden. Jedem Element des Berechnungsnetzes kann dabei ein bestimmter Materialtyp zugeordnet werden. Verschiedene definierte Materialtypen sind dabei mit der Oberflächenbeschaffenheit und mit dem zu erwartenden Strömungscharakter (Turbulenz) eng verbunden.

Die flächendeckende Erfassung der Oberflächenbeschaffenheit für die Vorländer erfolgt zumeist anhand der Flächennutzung (photogrammetrische Auswertedaten, Luftbilder). Diese Daten werden i.d.R.

durch Ortsbesichtigung und Fotodokumentation ergänzt. Die verfügbaren Daten können, abhängig vom Format (DXF, TIFF, SHAPE), zur besseren Orientierung, hinterlegt werden.

Die Definition von Materialtypen für die Netzelemente erfolgt in SMS über das Menü Elements ⇒ Assign Material Type. Vorher müssen jedoch die gewünschten Elemente selektiert werden.

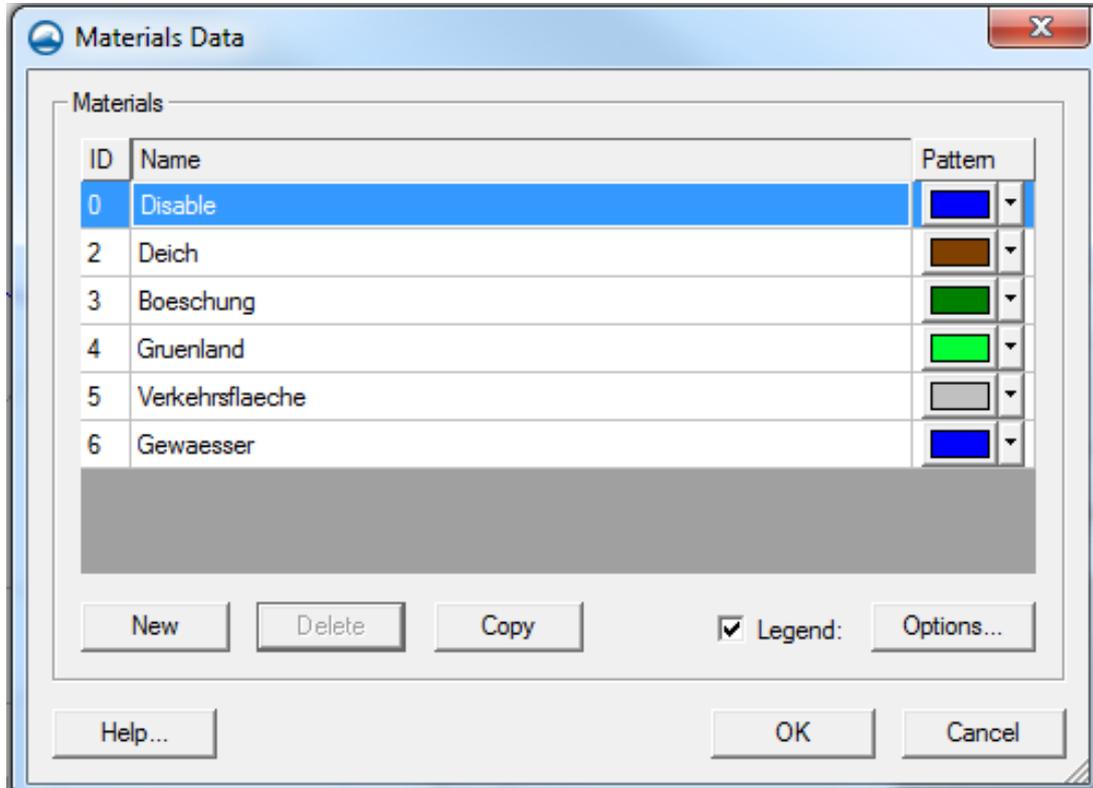


Abbildung 4.4: Dialog zur Konfiguration der Materialbereiche

Die Zuordnung von Rauheitswerten in $m^{1/3}/s$ und von Viskosität in m^2/s für den jeweiligen Materialtyp erfolgt über das HYDRO_AS-2D Menü Material Properties. Der für die Viskosität spezifizierte Wert bezieht sich dabei nur auf den konstanten Viskositätsanteil (vgl. [Kapitel 2.1.2](#)). Strickler-Werte können dabei als konstante Werte oder als wassertiefenabhängige Werte eingegeben werden.

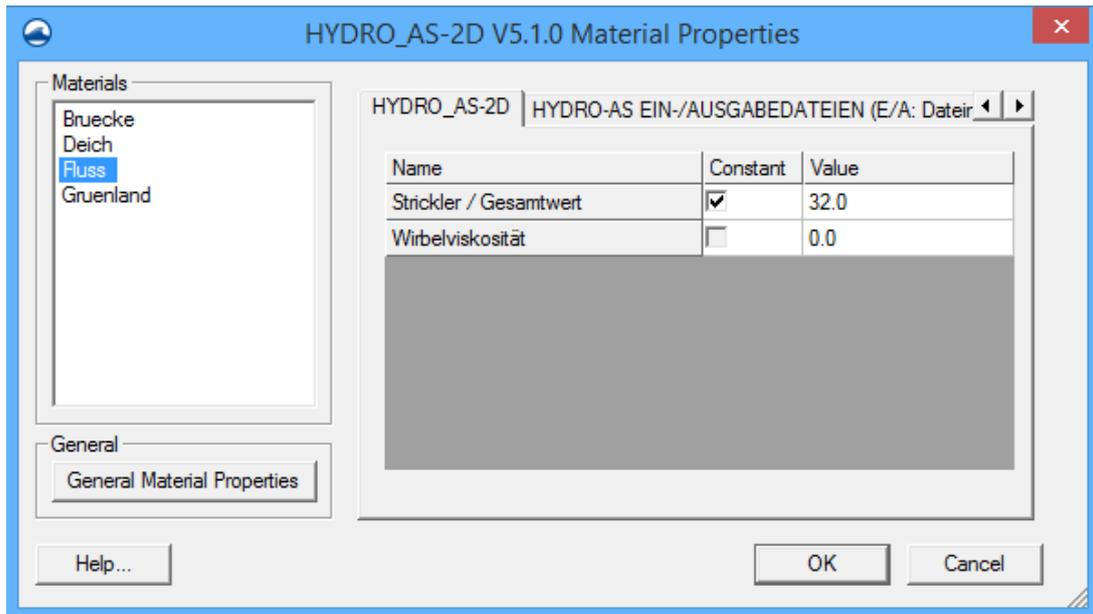


Abbildung 4.5: Material Properties Dialog

Zur Eingabe wassertiefenabhängiger Strickler-Werte entfernt man den Haken bei „Constant“. Die Menü-Ansicht ändert sich und man klickt Define. Es erscheint ein Dialog zur Definition der Beziehung zwischen Strickler-Wert (K_{St}) und Wassertiefe. Die Definition eines wassertiefenabhängigen Strickler-Wertes soll helfen, einige Problemstellungen besser zu modellieren. Der Anwender muss aber in der Lage sein, die Stützstellen der Beziehung $K_{St} = f(h)$ explizit vorgeben zu können (Handrechnung vorab, Rauheit aus Modellversuchen oder Kalibrierungen).

Anwendungsfälle sind u.a.:

- Bewuchs (z.B. rauer Waldboden mit kleinen Büschen, der bis zu einer bestimmten Fließtiefe durchströmt, dann überströmt wird,
- Raugerinneabfluss (z.B. aufgelöste Sohlrampen ohne/mit einzelnen Störsteinen).

Anmerkung: Die Berechnung von Reibungskräften erfolgt im Modell über die Kontrollvolumina. D. h. für jedes Kontrollvolumen wird die zugehörige Reibungsfläche ermittelt. Diese Fläche hängt nur von der Netzgeometrie ab und ist unabhängig von der Wassertiefe.

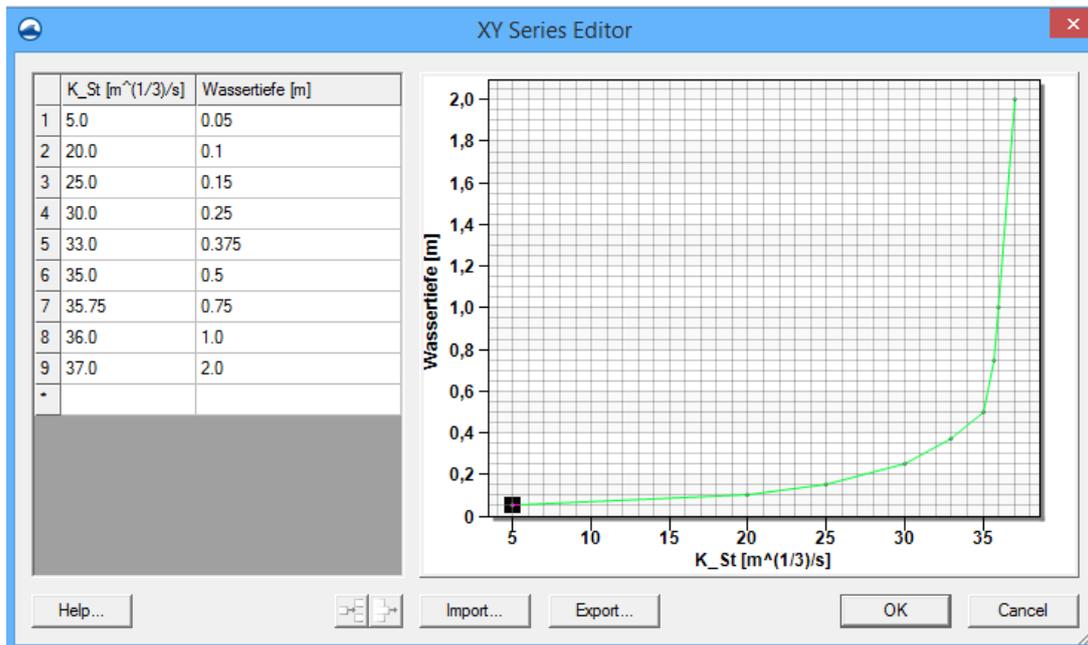


Abbildung 4.6: Vorgabe des Strickler-Werts in SMS als wassertiefenabhängiger Wert

Die Beziehung K_{St} /Wassertiefe wird tabellarisch definiert und als Diagramm dargestellt. Die Beziehung wird durch beliebig viele Punkte definiert. Der Strickler-Wert wird zwischen den vorgegebenen Punkten linear interpoliert und außerhalb der angegebenen Werte konstant extrapoliert.

Ermittelte Strickler-Werte werden zu den SMS-Ausgabe-Zeitpunkten in die Datei `kst_gesamt` im Verzeichnis `Data-out\AS` geschrieben. Diese Datei kann in SMS geladen werden.

4.5 Randbedingungen an Nodestings

Die Spezifikation von Randbedingungen erfolgt über die HYDRO_AS-2D Menüschnittfläche. Die interaktive Dateneingabe bietet eine komfortable Eingabe verschiedener Randbedingungen. Neben verschiedenen Ein- und Auslauftrandbedingungen können weitere sog. interne Randbedingungen, wie z.B. Bauwerke, spezifiziert werden. Beim Anlegen der Nodestings für die Randbedingungen ist Folgendes zu beachten:

- Netzknoten sollten nur zu maximal einer Randbedingung gehören. D.h. verschiedene Nodestings dürfen keine gemeinsamen Knoten haben.
- Ein Zulauf sollte ca. 300 m vom tatsächlichen Untersuchungsgebiet entfernt liegen, damit sich die Strömung bis dorthin richtig entwickeln kann und eventuelle Störungen abklingen können. Die besten Ergebnisse sind dabei auf einer gleichmäßigen geraden Fließstrecke mit wenigen Veränderungen im Strömungsverhalten zu erwarten.
- Auch der Auslauftrand sollte ca. 500 m unterhalb des zu untersuchenden Bereichs liegen, damit die Berechnungsergebnisse dort nicht von der Randbedingung beeinflusst sind. Ggf. ist zu untersuchen, wie weit der Einfluss der Randbedingung nach oben reicht, und die Auslaufstrecke entsprechend anzupassen.
- Die Angaben 300 m und 500 m sind Anhaltswerte. Bei Gewässern mit sehr geringem Gefälle können auch deutlich größere Abstände erforderlich sein.
- Die Vorgabe von Randbedingungen erfolgt über den Dialog Assign BC. Dazu wird ein Nodestring definiert, selektiert und mittels Rechtsklick auf den Nodestring der Dialog geöffnet.

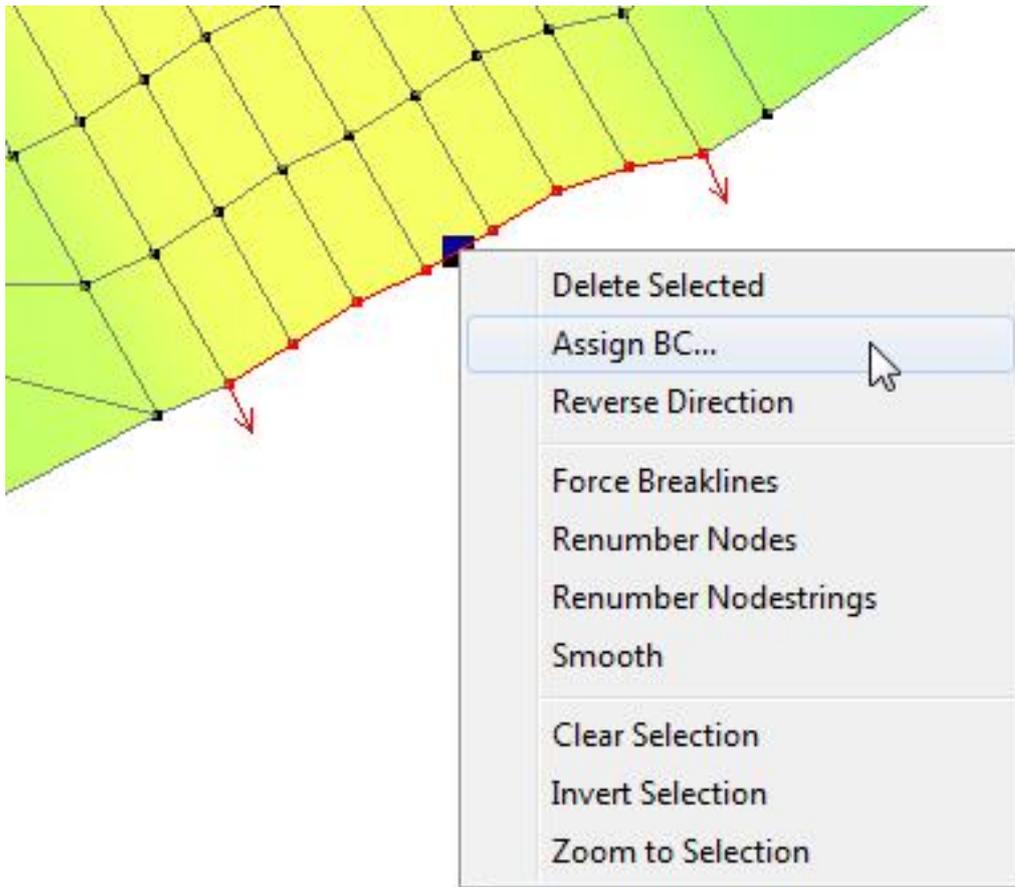


Abbildung 4.7: Vorgabe von Randbedingungen über den Dialog Assign BC

Anmerkung: Falsch oder unplausible gesetzte Randbedingungen können zu Instabilitäten und in Extremfällen zu vollkommen unbrauchbaren Ergebnissen führen. Deswegen ist besondere Sorgfalt beim Eingeben von Randbedingungen erforderlich. Überlappende Nodestrings sind grundsätzlich zu vermeiden. Ein in der Nähe eines Wirbels gelegener Auslauftrand kann zu Ungenauigkeiten in der resultierenden Lösung führen.

Hinweis: Verlegen Sie die Zu- und Auslaufränder nach außen und gestalten Sie das Berechnungsnetz in diesem Bereiche möglichst gleichmäßig. Lassen Sie die Einflüsse der Randbedingungen bis zum tatsächlichen Untersuchungsgebiet abklingen. Überlegen Sie sich hierzu, wie Sie die Zu- und Auslaufbereiche in einem physikalischen Modell gestalten würden. Wie würden Sie die Wasserzugabe und die Wasserentnahme in solch einem Modell ausführen? Es existiert eine Analogie zwischen den beiden Modellarten. Dies wird Ihnen ein besseres Verständnis für die 2D-numerische Modellierung vermitteln.

Im Folgenden werden die Randbedingungen analog zur Anordnung im Eingabedialog einzeln erläutert:

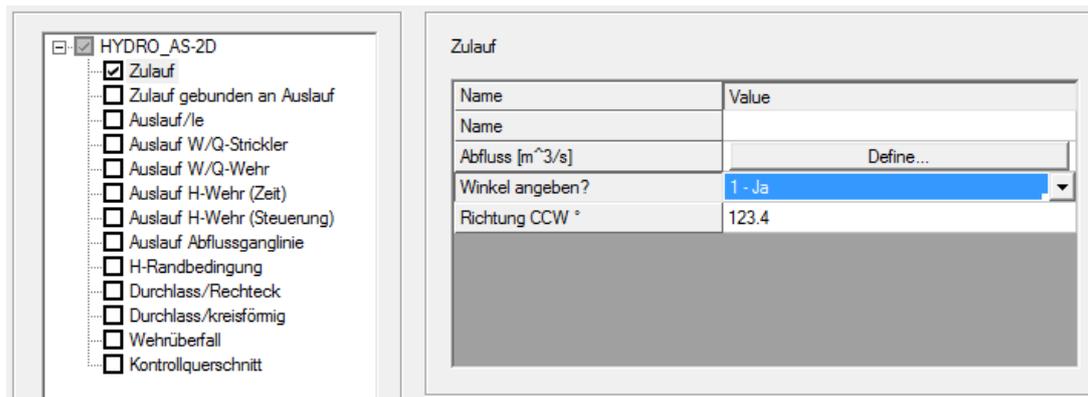


Abbildung 4.8: Dialog zur Eingabe der Randbedingungen an Nodestrings

4.5.1 Zulauf

Am Zulauf wird die Zuflussmenge (Abflusskurve) definiert. Die Strömungsrichtung wird per Default intern ermittelt und ist in diesem Fall senkrecht zum Nodestring. Die Strömungsrichtung kann aber auch als Winkel angegeben werden. Dazu wird im Feld Winkel angeben? 1 - Ja ausgewählt. Dann erscheint ein weiteres Eingabefeld für den Winkel. Dieser wird in Grad (zwischen 0° und 360°) gegen den Uhrzeigersinn (**C**ounter-**C**lock-**W**ise) eingegeben. Der Winkel 0° gibt dabei die Richtung von West nach Ost an.

Für die Vorgabe einer Zuflussganglinie wählen Sie die Randbedingung Zulauf aus. Der vorgegebene Zufluss wird dann im Modell über die im Nodestring enthaltenen Knoten, abhängig von der Wassertiefe verteilt.

Die Zuflussganglinie wird über den Define Dialog entweder interaktiv angegeben oder aus einer entsprechenden aufbereiteten Datei importiert. Die Zuflussdaten können ferner direkt aus einer Excel-Tabelle kopiert werden. Die importierten Daten können anschließend interaktiv bearbeitet werden. Die Zuflussganglinien können außerdem über Export in eine Datei im ASCII-Format ausgeschrieben werden (s. SMS-Handbuch für die Formatbeschreibung).

HYDRO_AS-2D führt immer eine instationäre Berechnung durch; deswegen muss die Vorgabe der Zuflussrandbedingung grundlegend als solche erfolgen. Falls eine stationäre Berechnung durchgeführt werden soll, kann die Zuflussganglinie z.B. wie in der folgenden Abbildung definiert werden.

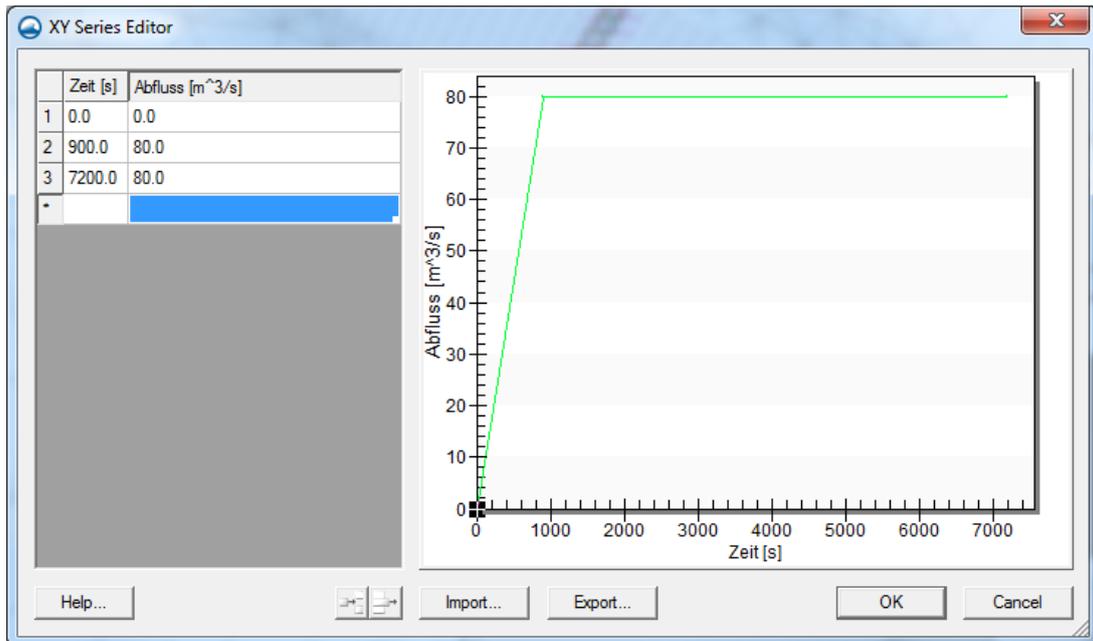


Abbildung 4.9: Define Curve Dialog mit Zuflussganglinie für eine stationäre Berechnung

Die Anstiegszeit kann dabei z.B. zwischen ca. 900 Sekunden und 1.800 Sekunden betragen. Es ist anzumerken, dass eine zu kurz gewählte Anstiegszeit zu einer rechnerischen Instabilität führen kann.

Wichtig: Die Abflussvorgabe erfolgt in m³/s und die Zeit in Sekunden. Die Zuflusswerte müssen positiv sein, damit eine Wasserzugabe erfolgen kann.

4.5.2 Zulauf gebunden an Auslauf

Es gibt die Möglichkeit, den Zulauf an einen bestimmten Auslauf zu koppeln. Das heißt, die gleiche Wassermenge, die am Auslauf entnommen wird, wird am entsprechenden Zulauf wieder zugegeben. Auf diese Art und Weise können steuerbare Bauwerke modelliert werden.

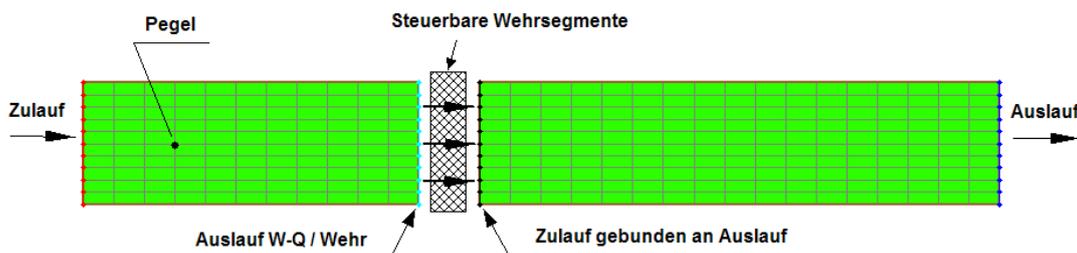


Abbildung 4.10: Zulauf gebunden an Auslauf - Skizze zur Modellierung

Die Koppelung an einen bestimmten Auslauf erfolgt, indem der Name des Auslaufquerschnitts oder die x- und die y-Koordinate eines Punktes am betreffenden Auslaufquerschnitt definiert wird.

Zulauf gebunden an Auslauf	
Name	Value
Name	
Zuordnung Auslauf via...	Name ▼
Auslauf-Name	02_Auslauf
Winkel angeben?	0 - Nein ▼

Abbildung 4.11: Zulauf gebunden an Auslauf - Vorgabe des Namens in SMS

Beachten Sie, dass der Name des Auslaufnodestrings gesetzt und eindeutig sein muss, damit die Zuordnung stattfinden kann. Ist dies nicht der Fall, werden entsprechende Fehlermeldungen ausgegeben.

Zulauf gebunden an Auslauf	
Name	Value
Name	
Zuordnung Auslauf via...	Koordinaten ▼
X-Koordinate [m]	123.0
Y-Koordinate [m]	456.0
Winkel angeben?	1 - Ja ▼
Richtung CCW °	321.0

Abbildung 4.12: Zulauf gebunden an Auslauf - Vorgabe der Koordinaten in SMS

Wie beim Zulauf (Zeitreihe) wird die Strömungsrichtung per Default intern ermittelt und ist in diesem Fall senkrecht zum Nodestring. Die Strömungsrichtung kann aber auch als Winkel angegeben werden.

Wichtig: Zwischen dem Zulauf und dem Auslauf dürfen keine direkten Netzverbindungen bestehen. Falls Elemente vorhanden sind, dann müssen die Elemente direkt unterhalb des Auslaufs und direkt oberhalb des Zulaufs gelöscht oder mit dem Materialtyp Disable belegt sein. Beachten Sie auch, dass das am Auslauf entnommene Wasser direkt im selben Zeitschritt am Zulauf wieder zugegeben wird. Die Strecke zwischen Auslauf und Zulauf wird dadurch übersprungen. Eine Fließzeit wird nicht berücksichtigt.

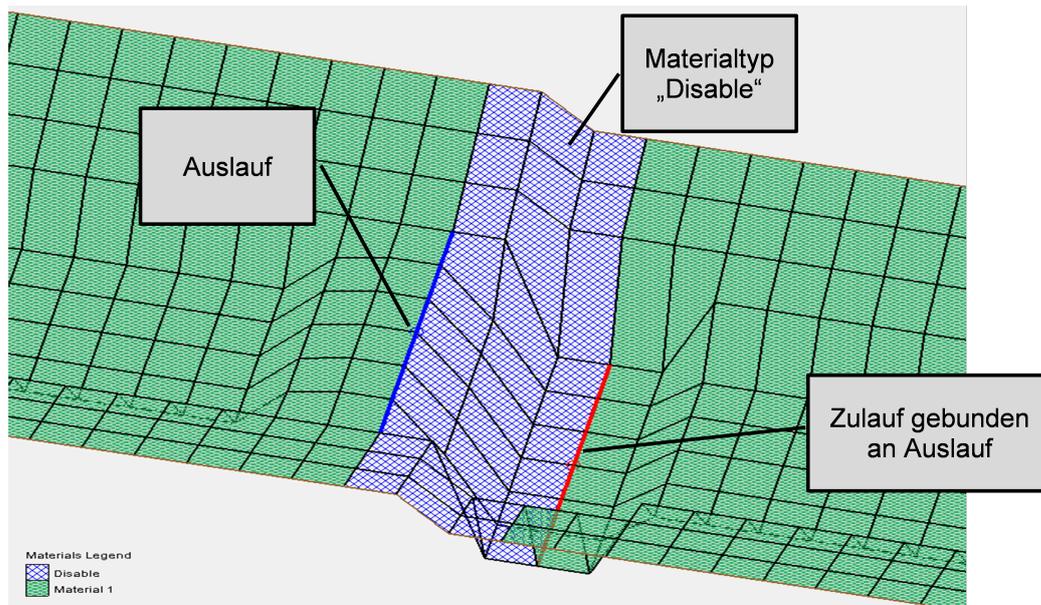


Abbildung 4.13: Zulauf gebunden an Auslauf - prinzipielle Darstellung mit Materialtypen

4.5.3 Auslauf / le

Wie bereits beim Einlauf wird auch hier ein Nodestring erzeugt und selektiert. Bei der Methode Auslauf / le wird ein Energieliniengefälle vorgegeben. Das Energieliniengefälle kann z.B. anhand des mittleren Sohlgefälles abgeschätzt werden. Diese Randbedingung ist relativ leicht zu definieren und kommt deshalb häufig bei der praktischen Anwendung vor. Die besten Ergebnisse werden dann erzielt, wenn der Auslauftrand annähernd senkrecht zur Strömungsrichtung verläuft.

Wichtig: Das Energieliniengefälle wird in Promille angegeben. Der angegebene Wert muss ein positives Vorzeichen haben, damit Wasser ausströmen kann.

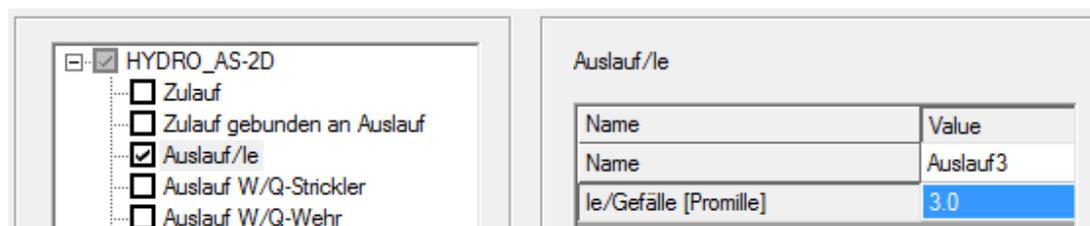


Abbildung 4.14: Auslauf/le

4.5.4 Auslauf W/Q-Strickler

Der Auslauf kann in Form einer Wasserstand-Abfluss-Beziehung (W-Q) definiert werden. Dabei kann die W-Q Beziehung am einfachsten über die Strickler-Formel definiert werden. Die W-Q Beziehung wird in SMS entsprechend der folgenden Abbildung definiert und anschaulich dargestellt. Bitte achten Sie darauf, dass die W-Q Beziehung stetig ansteigend sein und einen ausreichenden Bereich abdecken muss. Alle angegebenen Wasserstandswerte sollten dabei höher sein als der niedrigste Punkt im Nodestring. Die ausgeleitete Wassermenge wird am Auslaufquerschnitt abhängig vom mittleren Wasserstand bestimmt. Die Abflussaufteilung auf die Knoten des Nodestrings erfolgt anhand der aktuellen Wassertiefen. Diese Art von Abflussbedingung eignet sich insbesondere für (natürliche) Fließquerschnitte mit bekannter W-Q Beziehung wie z.B. an einer Pegelmessstelle.

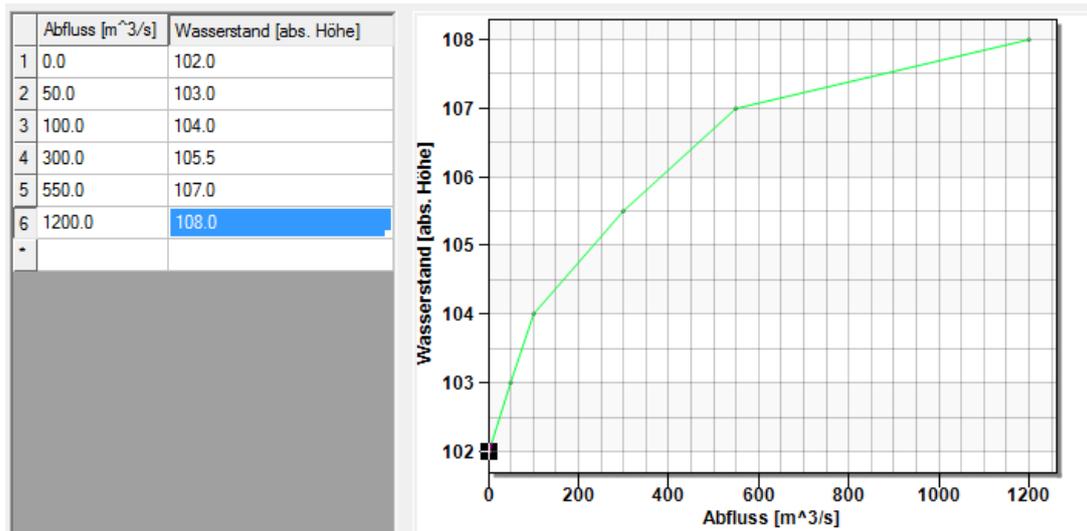


Abbildung 4.15: Definition einer W-Q-Beziehung

4.5.5 Auslauf W/Q-Wehr

Bei der Auslaufrandbedingung W/Q-Wehr kann der für die Abflussermittlung herangezogene Wasserstand einer beliebigen Stelle (Pegel) im Berechnungsgebiet entnommen werden. Der Wasserstand für die Abflussermittlung wird somit nicht am Auslauf, sondern am Pegel entnommen

Diese Art von Abflussbedingung eignet sich insbesondere für Bauwerke (speziell Wehrüberfälle) mit bekannter W/Q-Beziehung wie z.B. gesteuerte Wehrsegmente. Der Pegel für die Wasserstandsermittlung kann über den Namen des Pegelpunktes oder die x- und y-Koordinaten angegeben werden.

Auslauf W/Q-Wehr

Name	Value
Name	Wehr
Zuordnung Pegelpunkt via...	Name
Pegelpunkt-Name	OWPegelWehr
Abflusskoeffizient	0.55
W/Q Beziehung	Define...

Abbildung 4.16: Auslauf W/Q-Wehr: Pegelpunkt per Name angeben

Wird der Pegelpunkt über den Namen referenziert, muss für diesen Modellknoten die Randbedingung Pegelpunkt mit dem entsprechenden Namen gesetzt sein und der Name muss eindeutig sein.

Auslauf W/Q-Wehr

Name	Value
Name	Wehr
Zuordnung Pegelpunkt via...	Koordinaten ▼
Pegel: X-Koordinate [m]	50.0
Pegel: Y-Koordinate [m]	80.0
Abflusskoeffizient	0.55
W/Q Beziehung	Define...

Abbildung 4.17: Auslauf W/Q - Wehr: Pegelpunkt per Koordinaten angeben

Zusätzlich zum Pegel muss der Abflusskoeffizient für die Wehrformel spezifiziert werden.

Wichtig: Die Kronenhöhe entspricht der Sohlhöhe am jeweiligen Punkt des Auslaufquerschnitts. Passen Sie diese deshalb entsprechend an.

Für den Abflusskoeffizienten wird ein Wert von 0,55 empfohlen, was in etwa einem breitkronigen Wehrüberfall entspricht.

Die W/Q-Beziehung wird analog zur Randbedingung Auslauf W/Q - Strickler (vergl. [Kapitel 4.5.4](#)) eingegeben. Zuerst wird der Abfluss (Q_W) im gesamten Auslaufquerschnitt entsprechend der Wehrformel berechnet. Danach wird der Abfluss ($Q_{W/Q}$) gemäß der definierten W/Q-Beziehung und dem an der definierten Stelle befindlichen Wasserstand ermittelt. Der Abfluss am Auslauf (Q_A) wird schließlich als Minimum aus den beiden Werten berechnet:

$$Q_A = \min(Q_W, Q_{W/Q}) \quad (19)$$

Die beschriebene Vorgehensweise sichert, dass die W/Q-Beziehung eingehalten wird und dass der maximal mögliche Abfluss bei maximal vorhandenem Auslaufquerschnitt (alle Wehrsegmente offen) nicht überschritten wird. Das heißt, in Wirklichkeit greift die W-Q-Beziehung nur bis zu einem gewissen Wasserstand ein. Für geringere Wasserstände wird der Abfluss am Auslauf ggf. gemindert. Dies entspricht in der Realität einer geringeren Wehrlänge, d.h., nicht alle Wehrsegmente sind offen.

Anmerkung: Der Abfluss wird dem ermittelten Wasserstand nicht momentan, sondern asymptotisch mit einer gewissen Verzögerung, entsprechend der definierten W/Q Beziehung angepasst, damit es nicht zu einer „Übersteuerung“ und demzufolge zu Instabilitäten kommt.

Die Werte für den Abfluss und den Wasserstand müssen aufsteigend sein.

4.5.6 Auslauf H-Wehr (Zeit)

Mit dieser Randbedingung kann der Modellierer zeitabhängig einen gewünschten Oberwasserstand am Wehr bzw. Auslaufrand definieren.

Wird die Auslaufrandbedingung H-Wehr (Zeit) definiert, so sind folgende Eingaben erforderlich:

- Lage des Oberwasserpegels per Name oder als X- und Y-Koordinaten, in dem die Zielgröße Wasserspiegel vorgegeben werden soll (analog zur Randbedingung W/Q-Wehr)
- Abflusskoeffizient (des geometrisch abgebildeten Wehres, analog zur Randbedingung W/Q-Wehr)
- WSPL-Zeitreihe: Wasserspiegellage (WSPL) des Soll-Wasserstands am Oberwasserpunkt als Zeitreihe
- Wehrgeschwindigkeit in m/s: Anhebungs-/Absenkungsgeschwindigkeit der Wehrkrone

Die Steuerung des Wehres erfolgt in der Form, dass der gewünschte Wasserspiegel zum aktuellen Zeitpunkt am Oberwasserpunkt eingehalten wird, wobei sich das Wehr maximal mit der angegebenen Wehrgeschwindigkeit bewegen kann.

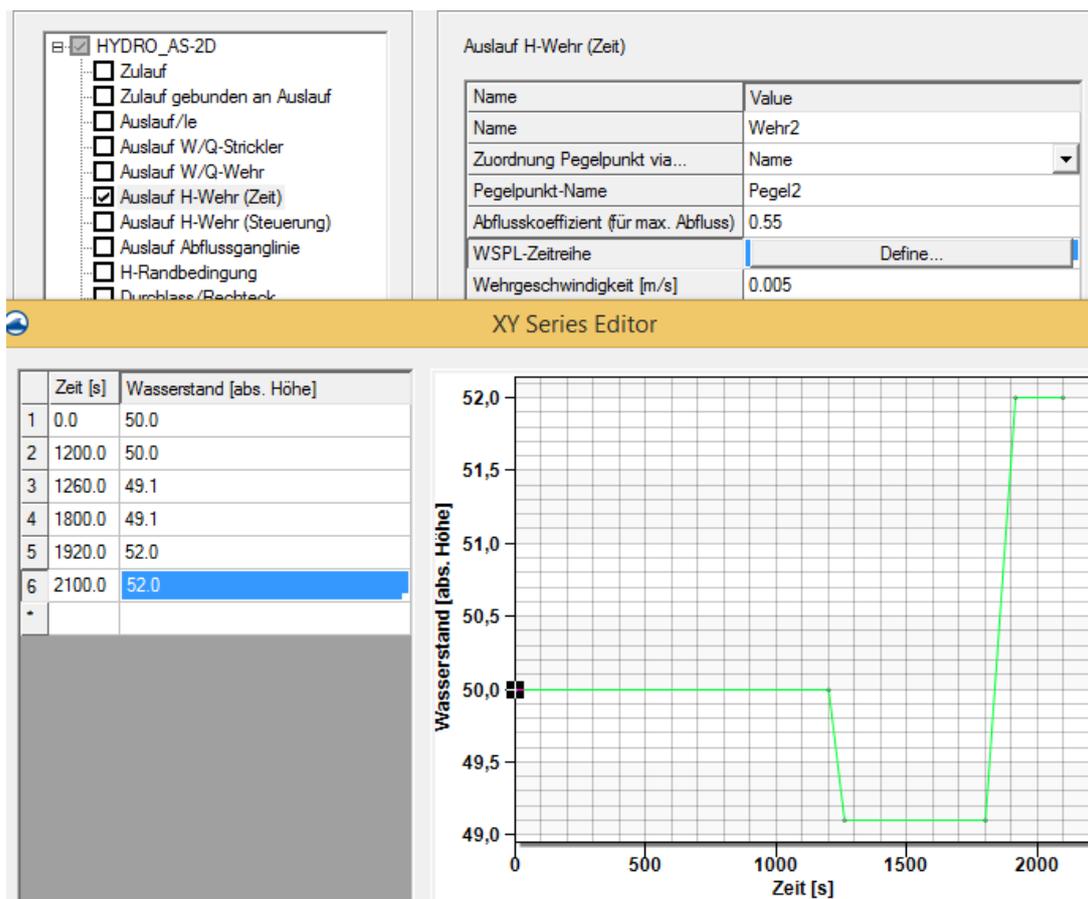


Abbildung 4.18: Eingabe der H-Wehr (Zeit) Randbedingung in SMS

Die Randbedingungen H-Wehr (Zeit) und H-Wehr (Steuerung) (vergl. [Kapitel 4.5.7](#)) haben die weitere Option der Berechnung mit vermindertem Abfluss durch unvollkommenen Überfall.

Standardmäßig wird bei der Auslaufrandbedingung keine Abminderung des Abflusses durch Unterwassereinflüsse berücksichtigt (Annahme eines vollkommenen Überfalls). Der Default- Wehrtyp ist 0 – ohne Unterwassereinfluss. Bei Bedarf kann der Modellierer für die Randbedingung Auslauf H-Wehr einen Rückstau einfluss vorgeben.

Da der Abminderungsfaktor vom Wehrtyp abhängt, muss der entsprechende Typ gewählt werden. Bei allen Wehrtypen außer dem Wehrtyp 0 – ohne Unterwassereinfluss werden die Unterwassereinflüsse berücksichtigt.

Folgende Wehrtypen stehen zur Auswahl:

- 0 – ohne Unterwassereinfluss
- 1 – breitkronig
- 2 – Dachwehr
- 3 – Standardprofil
- 4 – scharfkantig

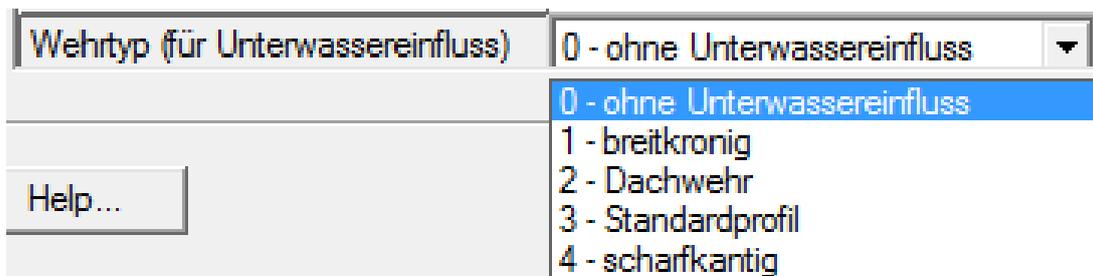


Abbildung 4.19: Wehrtypen für den unvollkommenen Überfall

Der Abflusskoeffizient sollte demnach vom Anwender passend gewählt werden. Als Faustregel kann folgende Einteilung genutzt werden:

Wehrtyp	Abflusswert
ohne Unterwassereinfluss	beliebig, zwischen 0 und 1
breitkronig	0,49 bis 0,55
Dachwehr	0,75 bis 0,79
Standardprofil	0,73 bis 0,75
scharfkantig	0,64

Um die Unterwassereinflüsse in Betracht zu ziehen, bedarf es der Information der Unterwasserwasserspiegellage. Dazu ist ein weiterer Bezug auf einen Unterwasserpegel zu legen. Diese Eingabe erscheint, sobald einer der Wehrtypen 1 bis 4 gewählt wird. Auch hier kann der Pegelpunkt per Name oder Koordinaten referenziert werden.

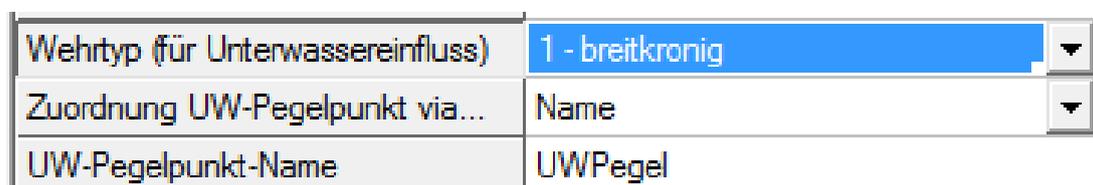


Abbildung 4.20: Angabe des Unterwasser-Pegelpunktes per Name

Eine Besonderheit stellt der Typ 3 dar, da die Kurve des Abminderungsfaktors bei einem Standardprofil zusätzlich von der Wehrhöhe (Wehrkronenhöhe minus Sohle OW) abhängt (vgl. Grundlagen zum Thema „Unvollkommener Überfall“ z. B. in [7]). Die Eingabe der **minimalen** Wehrhöhe wird daher für 3 - Standardprofil bereitgestellt.

Wehrtyp (für Unterwassereinfluss)	3 - Standardprofil	▼
Zuordnung UW-Pegelpunkt via...	Koordinaten	▼
Unterwasser: X-Koordinate [m]	304.5	
Unterwasser: Y-Koordinate [m]	30.4	
Wehrhöhe [m]	3.5	

Abbildung 4.21: Angabe des Unterwasser-Pegelpunktes per Koordinaten und zusätzliche Angabe der Wehrhöhe für ein Standardprofil

4.5.7 Auslauf H-Wehr (Steuerung)

Ist die Steuerung eines Wehres nicht zeitabhängig, sondern abflussabhängig durchzuführen, wird die Randbedingung H-Wehr (Steuerung) eingesetzt. Der Modellierer gibt hierfür einen Kontrollquerschnitt und einen Oberwasserpegel an.

Sowohl für den Oberwasserpegel als auch für den Kontrollquerschnitt gibt es jeweils die Möglichkeit die Zuordnung per Namen oder x- und y-Koordinaten vorzunehmen.

Während einer Simulation wird am Kontrollquerschnitt der Abfluss und am Oberwasserpunkt der Wasserstand abgefragt.

Der Modellierer kann Bedingungen vorgeben, in welchen Abflussspektren welcher Wasserstand im Oberwasserpunkt durch die Wehrsteuerung eingehalten werden soll.

Auslauf H-Wehr (Steuerung)

Name	Value
Name	Wehr3
Zuordnung Pegelpunkt via...	Name ▼
Pegelpunkt-Name	Pegel3
Abflusskoeffizient (für max. Abfluss)	0.55
Querschnitt für Abflussmessung v...	Name ▼
Querschnitt-Name	KQ
Q1 [m ³ /s]	4.0
Q2 [m ³ /s]	12.0
Q3 [m ³ /s]	9.0
WSPL1 [abs. Höhe]	51.0
WSPL2 [abs. Höhe]	49.0
WSPL3 [abs. Höhe]	53.0
WSPL4 [abs. Höhe]	51.0
Wehrgeschwindigkeit [m/s]	0.01

Abbildung 4.22: Eingabe der H-Wehr (Steuerung) Randbedingung in SMS

Auf diese Weise kann mit der Randbedingung H-Wehr (Steuerung) insbesondere eine Vorabsenkung abgebildet werden. Hierzu werden drei Abflussgrenzen (Q1, Q2 und Q3) zum Definieren der Abflussspektren (Spektrum) und die zugehörigen bzw. gewünschten Wasserspiegellagen im Oberwasserpunkt (WSPL1, WSPL2, WSPL3 und WSPL4 (=WSPL1)) vorgegeben. Am Beispiel einer Vorabsenkung gilt:

- Im Spektrum 1 (Sollwasserspiegel = WSPL1) wird der Normalabfluss (normales Stauziel) gesteuert ($Q < Q1$).
- Das Spektrum 2 (Sollwasserspiegel = WSPL2) bildet die Phase der Vorabsenkung ab (Schaffung von Retentionsraum vor dem Hochwasserscheitel), die bei Überschreitung eines Grenzabflusses ($Q > Q1$) greift.
- Das Spektrum 3 (Sollwasserspiegel = WSPL3) bildet die Phase ab, in der der Hochwasserscheitel zurückgehalten werden soll (Beginn $Q > Q2$, Ende $Q < Q3$). Der Wasserspiegel im Oberwasserpunkt wird beispielsweise auf das außergewöhnliche Stauziel (WSPL3) gesteuert.
- Das Spektrum 4 (Sollwasserspiegel = WSPL4) bildet die Phase nach dem Hochwasser ab ($Q < Q3$). Hier kann der Wasserspiegel im Oberwasserpunkt beispielsweise wieder auf Normalstau gesteuert werden (WSPL4 = WSPL1).

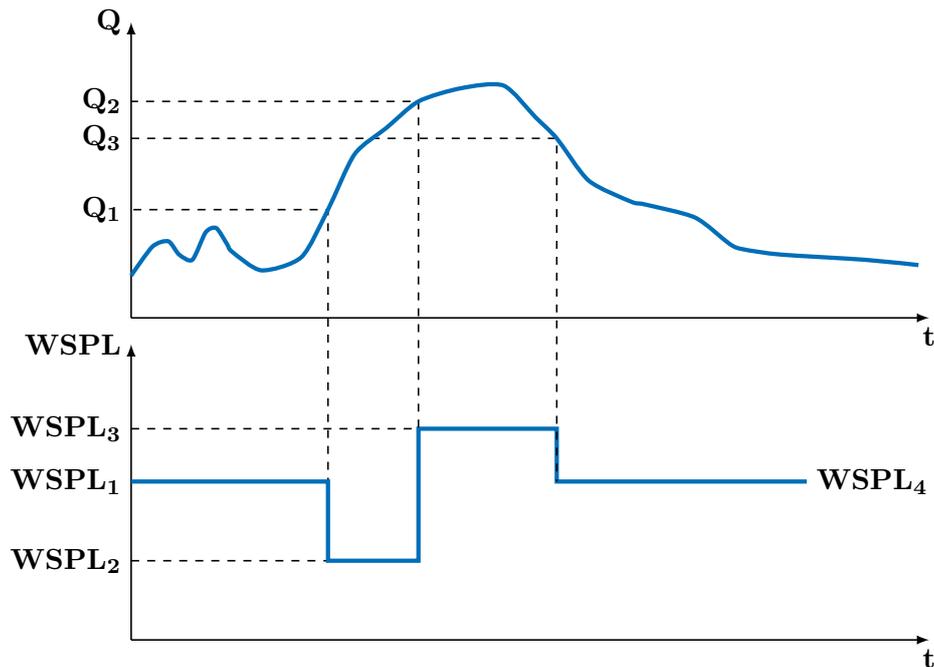


Abbildung 4.23: Beispiel einer Wehrsteuerung bei einem Hochwasser (inkl. Vorabsenkung)

Wie bei der Randbedingung H-Wehr (Zeit) wird die Wehrgeschwindigkeit (Anhebungs-/ Absenkgeschwindigkeit der Wehrkrone) berücksichtigt.

Zusätzlich ist es möglich mit Unterwassereinfluss zu rechnen. Die Eingaben hierzu sind analog zur Randbedingung Auslauf H-Wehr (Zeit), siehe vorherigen Abschnitt [Kapitel 4.5.6](#).

4.5.8 Auslauf Abflussganglinie

Analog zum Zulauf wird eine Ganglinie angegeben, die hier den Abfluss am Auslauf-Nodestring definiert. Die Abflusswerte werden als nicht-negative Werte angegeben. Falls der durch die Ganglinie vorgegebene Abfluss zu einem Zeitpunkt größer ist als der maximal mögliche Abfluss, wird nur der maximal mögliche Abfluss aus dem Modell gezogen. Negative Wasserstände in Modellbereich des Auslaufs können somit nicht auftreten. Die Ergebnisse in der Datei q_strg.dat können durch diesen Mechanismus von der angegebenen Abflussganglinie abweichen.

Bei Erreichen des maximal möglichen Abflusses fallen die Knoten des Auslaufnodestrings i.A. trocken. Dadurch kann ein großes Wasserspiegelgefälle zwischen den Knoten des Auslaufnodestrings und ihren Nachbarknoten entstehen, was wiederum zu hohen Rechenzeiten führt. Achten Sie daher darauf, dass die Werte der vorgegebenen Abflussganglinie nicht zu groß sind.

4.5.9 H-Randbedingung

Auf dem Nodestring wird die Wasserspiegellage (WSPL) als absolute Höhe vorgeschrieben. Dabei ist die WSPL zeitabhängig. Die WSPL-Zeitreihe sollte entsprechend der Anwendung hinreichend genau angegeben werden.

Je nach Höhe der Wasserspiegellage der inneren Nachbarknoten ist die H-Randbedingung ein Zulauf oder Auslauf. Der Unterschied der Wasserspiegellagen der Randknoten und der inneren Nachbarknoten sollte nicht zu groß sein, weil dann die Voraussetzungen für die Anwendbarkeit der Flachwassergleichungen verletzt sind. Außerdem kann ein starkes Wasserspiegelgefälle zu hohen Rechenzeiten führen. Daher wird empfohlen, entweder die WSPL der H-Randbedingung in einer Vorlaufzeit langsam

auf den gewünschten Wert ansteigen zu lassen oder eine passende `wtiefe_0.dat` (siehe [Kapitel 4.7](#)) zu benutzen.

4.5.10 Durchlässe

Durchlässe können in HYDRO_AS-2D mittels sogenannter 1D-Bauwerke (siehe [Kapitel 2.2](#)) abgebildet werden. Ein 1D-Bauwerk wird durch einen Nodestring, der genau zwei Knoten enthält, abgebildet. Dabei entsprechen die Knoten dem Bauwerkseinlauf und -auslauf. Die Länge des Nodestrings ist die Bauwerkslänge. Die Querschnittsfläche kann ein Rechteck oder ein Kreis sein. Je nach Durchlasstyp müssen dann die Breite und Höhe oder der Durchmesser angegeben werden.

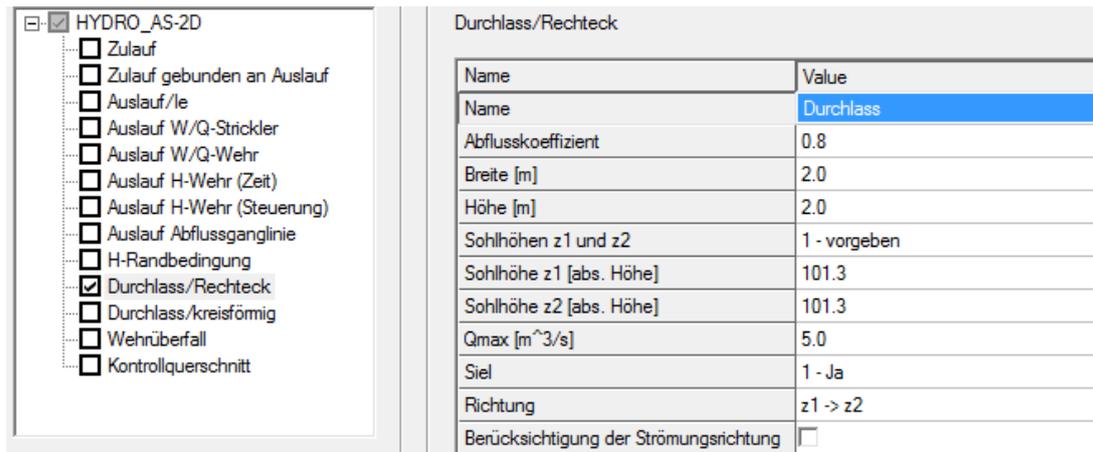


Abbildung 4.24: Durchlass / Rechteck

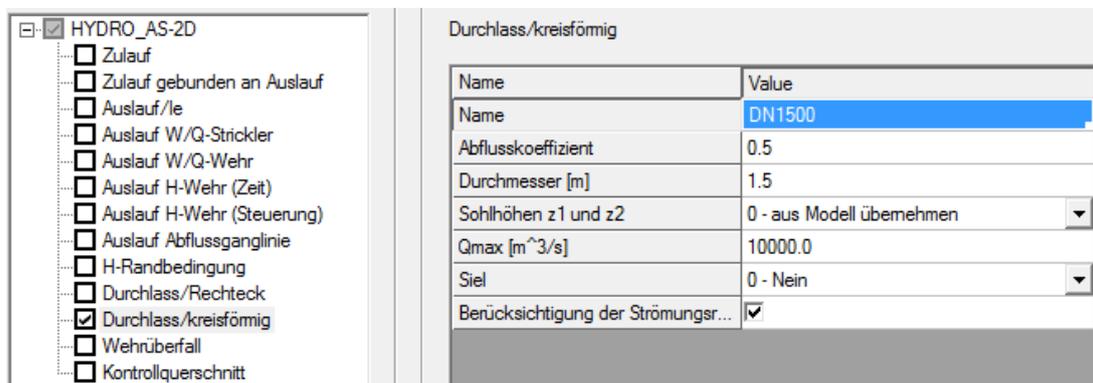


Abbildung 4.25: Durchlass / kreisförmig

Der Abflusskoeffizient ist ein Faktor für die Abflussberechnung und muss zwischen 0 und 1 liegen.

Für jeden 1D-Durchlass werden die Bodenhöhen des Durchlasses an den beiden Knoten des Nodestrings definiert. Dabei wird die Bodenhöhe am ersten Knoten mit z_1 und die Bodenhöhe am zweiten Knoten mit z_2 bezeichnet. z_1 und z_2 sind absolute Höhen. Sie können manuell angegeben werden (Sohlhöhen z_1 und z_2 auf *1 - vorgeben* stellen) oder werden automatisch auf die Sohlhöhen der entsprechenden Netzknoten gesetzt (Sohlhöhen z_1 und z_2 auf *0 - aus Modell übernehmen* stellen). Bei der manuellen Eingabe achten Sie darauf, dass die Höhen z_1 und z_2 jeweils **nicht** unter den Sohlhöhen der zugehörigen Netzknoten liegen.

Es gibt die Möglichkeit, Durchlässe als Drosselbauwerke zu modellieren. Hierzu gibt man in der Zeile

Q_{\max} [m^3/s] den gewünschten maximalen Abfluss in m^3/s ein. Der Wert steht per Default auf 10.000 m^3/s .

Des Weiteren gibt es die Möglichkeit, für einen 1D-Durchlass nur eine mögliche Fließrichtung vorzugeben (Siel). Im Beispiel für den rechteckigen Durchlass (Abbildung 4.24) ist der Abfluss nur von z_1 zu z_2 möglich. Per Default ist ein Durchfluss in beide Richtungen möglich (Siel = 0 - Nein).

Die letzte Einstellung bezieht sich auf die Abflussberechnung. Hier kann ausgewählt werden, ob die Richtung der Strömung im Oberwasser relativ zum Durchlass in Betracht gezogen werden soll oder nicht. Siehe auch [Kapitel 2.2.2](#).

Beim Anlegen der 1D-Bauwerke ist Folgendes zu beachten:

- Die Kontrollvolumen der zugrunde liegenden Netzknoten (Einlauf und Auslauf) sollten eine vergleichbare Größe aufweisen.
- Der Durchlassquerschnitt sollte nicht größer sein als die zugehörigen Kontrollvolumina. Um dies sicher zu stellen, kann folgende Faustregel hilfreich sein: Die Durchlasshöhe und -breite, bzw. der Durchmesser, sollten nicht größer als die an die Knoten des Nodestings angrenzenden Kanten sein.
- Das Netz ist im Bereich der 1D-Bauwerke so anzupassen, dass der Durchfluss im Bauwerk korrekt erfasst werden kann. Hierzu ist es möglich, die Elemente unterhalb des Einlaufbereichs des 1D-Bauwerks auf disabled zu setzen. Falls ein Überströmen des Bauwerks möglich sein soll, müssen die Sohlhöhen des Netzes unterhalb des Einlaufbereichs unter Berücksichtigung der Bauwerksoberkante angepasst werden.

Größere 1D-Bauwerke sollten daher nicht nur mit einem, sondern mit mehreren parallelen Nodestings abgebildet werden. Die wirksame Abflussfläche bzw. Durchlassbreite sollte dann auf die Nodestings aufgeteilt werden. Alternativ kann der Abflusskoeffizient angepasst werden.

Brücken und andere Strukturen, die eine zweidimensionale Berechnung erfordern sollten mit der Knotenrandbedingung KUK abgebildet werden, siehe [Kapitel 4.6.1](#).

4.5.11 Wehrüberfall

Ein 1D-Wehr wird ähnlich wie ein 1D-Durchlass durch einen Nodestring, der aus genau zwei Knoten besteht, abgebildet. Die Wassertiefen und Wasserspiegellagen an diesen beiden Knoten werden für die Abflussbestimmung benutzt. Der Überfallbeiwert wird gemäß des Wehrtyps gewählt. Die Kronenhöhe ist als absolute Höhe anzugeben. Achten Sie darauf, dass die Kronenhöhe höher als die Sohlhöhen der beiden Knoten des Nodestings ist.

Die letzte Einstellung bezieht sich auf die Abflussberechnung. Hier kann ausgewählt werden, ob die Richtung der Strömung im Oberwasser relativ zum Wehr in Betracht gezogen werden soll oder nicht. Siehe auch [Kapitel 2.2.1](#).

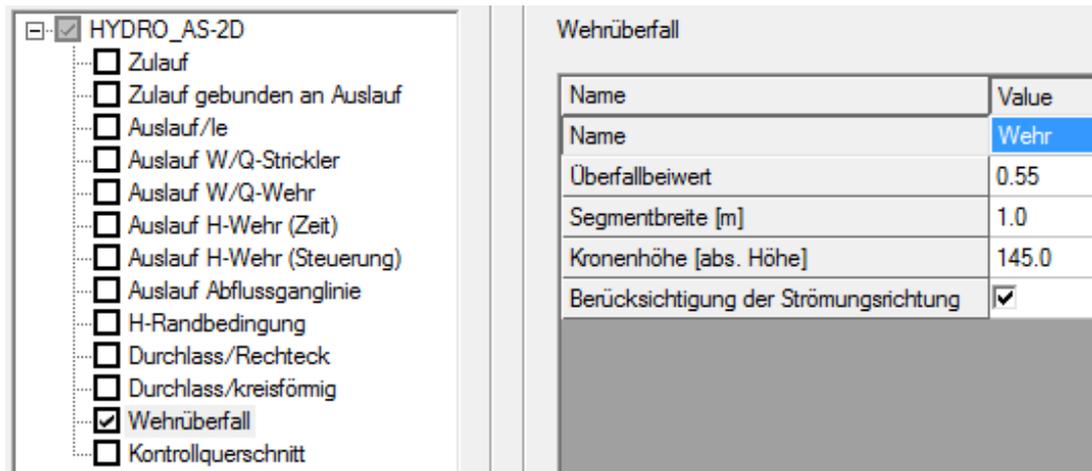


Abbildung 4.26: Parameter für die Randbedingung Wehrüberfall

Beim Anlegen der 1D-Wehre ist Folgendes zu beachten:

- Die Kontrollvolumen der zugrunde liegenden Netzknoten sollten eine vergleichbare Größe aufweisen.
- Die durch die Wehrformel ermittelte Wassermenge sollte das Wasservolumen am Oberwasserknoten nicht überschreiten.

Um dies sicher zu stellen, kann folgende Faustregel hilfreich sein: Die Segmentbreite sollte nicht größer als die an die Knoten des Nodestrings angrenzenden Kanten sein.

- Elemente im Wehrbereich (d.h. zwischen den beiden Knoten des Wehr-Nodestrings) müssen auf disable gesetzt werden.

Größere 1D-Wehre sind daher nicht nur mit einem, sondern mit mehreren parallelen Nodestrings abzubilden. Dabei wird die Segmentbreite auf die Nodestrings aufgeteilt.

Anmerkung: Wehre können auch geometrisch im Netz abgebildet werden. Dann wird der Abfluss nicht durch die Wehrformel nach Du Buat, sondern über die Flachwassergleichung bestimmt.

4.5.12 Kontrollquerschnitt

Die Kontrollquerschnitte zur Abflussermittlung werden mithilfe von Nodestrings definiert. Zunächst muss dazu ein Nodestring definiert und selektiert werden. Die Punktverbindung sollte in Strömungsrichtung gesehen von links nach rechts durchgeführt werden, damit der ermittelte Abfluss ein positives Vorzeichen erhält. (Die Pfeile an den äußeren Knoten des Nodestrings zeigen dann der Strömungsrichtung entgegen). Die durch die Kontrollquerschnitte ermittelten Abflüsse werden gemäß der in den Global Parameters definierten Zeitintervall Q_Strg [s] in die Datei *q_strg.dat* ausgeschrieben.

Die Ganglinien von Zu- und Auslaufnodestrings und von H-Randbedingungen werden ebenfalls in die Datei *q_strg.dat* geschrieben.

Bei der Verwendung von Kontrollquerschnitten beachten Sie bitte Folgendes:

- Sie können mehrere Kontrollquerschnitte nebeneinander legen um deren Abflüsse additiv zu betrachten. Beispielsweise kann so der Abfluss durch das linke Vorland, den Flussschlauch und das rechte Vorland differenziert werden. In diesem Fall sollten Sie jeweils genau eine Elementkante Abstand zwischen den Enden der Nodestrings frei lassen. Andernfalls werden an den Endknoten Abflussanteile doppelt gerechnet.

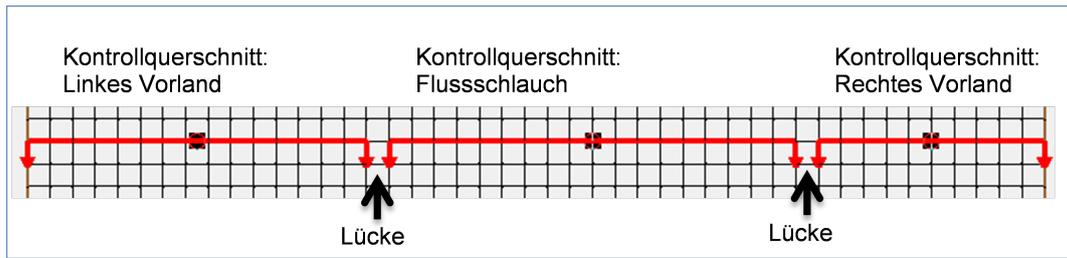


Abbildung 4.27: Anordnung mehrerer Kontrollquerschnitte nebeneinander

- Das Vorzeichen des Abflusses durch den Kontrollquerschnitt hängt von der Reihenfolge der Knoten im Nodestring ab. Nodestrings, die in Fließrichtung von links nach rechts definiert sind, haben einen positiven Abfluss. Nodestrings, die von rechts nach links definiert sind, haben einen negativen Abfluss.
- Kontrollquerschnitte sollen nur entlang von Modellkanten, die vollständig im Inneren des Berechnungsnetzes liegen, definiert werden. Ein Kontrollquerschnitt, dessen Kanten (teilweise) am Rand liegt, wird nicht korrekt berechnet.
- Bei der Interpretation der ausgegebenen Daten beachten Sie bitte, dass Kontrollquerschnitte keine Randbedingungen (Zuflüsse, Abflüsse, 1D-Bauwerke, ...) berücksichtigen. Wenn beispielsweise ein Wehr oder Durchlass den Kontrollquerschnitt schneidet oder auf dem Kontrollquerschnitt beginnt, werden die Abflüsse dieses Bauwerkes nicht mit erfasst.

Hinweis: Die berechneten Abflussmengen durch die definierten Bauwerkssegmente sind in der Datei `bw_tmp.dat` zu finden.

- Die Abflusswerte in der Datei `q_strg.dat` sind die Abflüsse zu den `Q_strg`-Ausgabezeitpunkten. Dies ist bei der Verwendung der Ausgaben für Wasserbilanzbetrachtungen zu berücksichtigen.
- Sie können mit SMS ebenfalls Abflüsse durch Kontrollquerschnitte berechnen. Die folgende Tabelle vergleicht die beiden Verfahrensansätze:

Tabelle 4.2: Unterschiede der Abflussberechnung an Kontrollquerschnitten in SMS und in HYDRO_AS-2D

SMS	HYDRO_AS-2D
Berechnung auf Basis der Wassertiefe und der Geschwindigkeiten an den Knoten	Berechnung auf Basis interner Zwischenergebnisse der Simulation
Unsicherheiten von mehr als 10% bei nicht homogenen Netzen oder Abflussbedingungen	Genaueres Verfahren
Berechnung an beliebigen Stellen nach der Simulation	Kontrollquerschnitte müssen vor der Simulation festgelegt werden

- In früheren HYDRO_AS-2D-Versionen sollten Kontrollquerschnitte aus numerischen Gründen so verlaufen, dass das Netz und der Abfluss in der Umgebung des Nodestrings möglichst gleichförmig waren. Diese Einschränkungen entfallen ab HYDRO_AS-2D 5.0.0. Sie gelten aber noch für die Abflussermittlung mit SMS.

4.5.13 Namen für Nodestrings

Für alle Nodestrings (Randbedingungen, 1D-Bauwerke, Kontrollquerschnitte) können Namen (max. 12 Zeichen) vergeben werden. Wird der Name in SMS nicht gesetzt, wird die Nummer des ersten Knotens im Nodestring als Default-Name benutzt. Die Sortierung der Ausgabe in der `q_strg.dat` erfolgt durch

die Nodestringnamen in alphabetischer Reihenfolge. Damit ist eine einfache Identifizierung der Nodestrings für den Vergleich von verschiedenen Rechenläufen gewährleistet.

Anmerkung: Bei Richtungsänderung des Nodestrings ändert sich der erste Knoten im Nodestring. Das hat zur Folge, dass sich der Default-Name und möglicherweise die Reihenfolge der Nodestrings in der q_strg.dat ändern.

Bei mehrfacher Verwendung desselben Namens erscheint eine Warnung.

4.6 Randbedingungen an Netzknoten

An Netzknoten können verschiedene Arten von Randbedingungen definiert werden.

Die Knoten-Randbedingungen

- KUK/Druckabfluss, siehe [Kapitel 4.6.1](#)
- Pegelpunkte, siehe [Kapitel 4.6.2](#)
- Zufluss/Abfluss, siehe [Kapitel 4.6.5](#)

werden in SMS über den Dialog Node Boundary Conditions gesetzt. Dazu wird der Netzknoten mit der Funktion Select Mesh Node ausgewählt. Dann wird der Dialog mittels Rechtsklick und der Option Assign BC geöffnet, wie das Beispiel einer Konstruktionsunterkante (KUK) zeigt, siehe [Kapitel 4.6.1](#).

Die Knoten-Randbedingungen

- Niederschläge, siehe [Kapitel 4.6.3](#)
- Konstante Zuflüsse/Abflüsse, siehe [Kapitel 4.6.4](#)
- Zeitabhängige Zuflüsse, siehe [Kapitel 4.6.6](#)

werden über zusätzliche Eingabe-Dateien gesteuert und in HYDRO_AS-2D als Quellterme dargestellt.

4.6.1 Konstruktionsunterkante (KUK/Druckabfluss)

Die meisten Brücken werden in SMS mithilfe der Randbedingung KUK/Druckabfluss modelliert. Dazu wird den Netzknoten im Bereich des Bauwerks die zusätzliche Randbedingung KUK/ Druckabfluss zugewiesen. Die Punkte im Bauwerksbereich werden mit der Funktion Select Mesh Node ausgewählt. Dann wird mittels Rechtsklick und der Option Assign BC ein Haken bei KUK/ Druckabfluss gesetzt und die gewünschte Höhe der Konstruktionsunterkante als absolute Höhe in das Parameterfeld eingetragen.

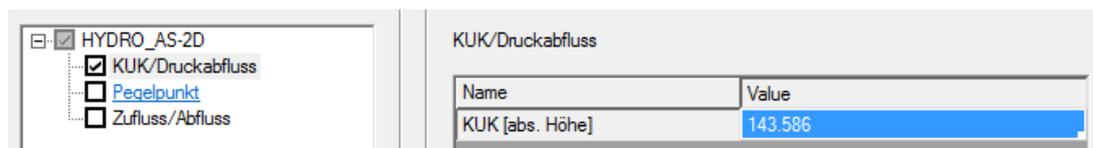


Abbildung 4.28: Definition einer KUK

Die Visualisierung der KUK in SMS kann in den Display Options unter Nodal boundary conditions angepasst werden.

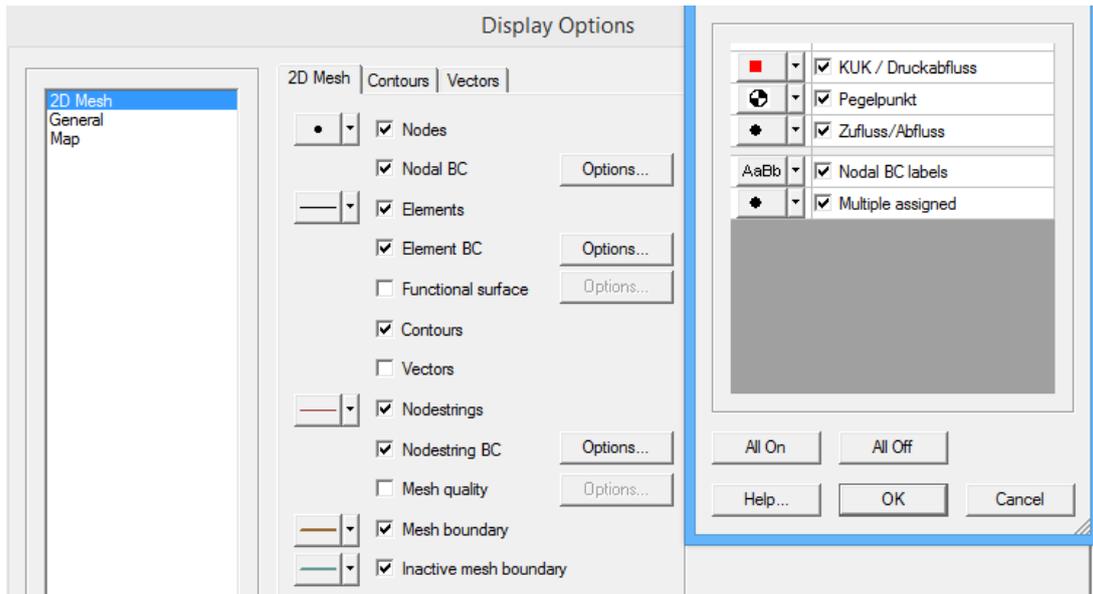


Abbildung 4.29: Nodel Boundary Conditions in den Disply Options

In der folgenden Abbildung wurden die Knoten, denen die Randbedingung KUK zugewiesen ist, mit einem blauen Quadrat markiert.

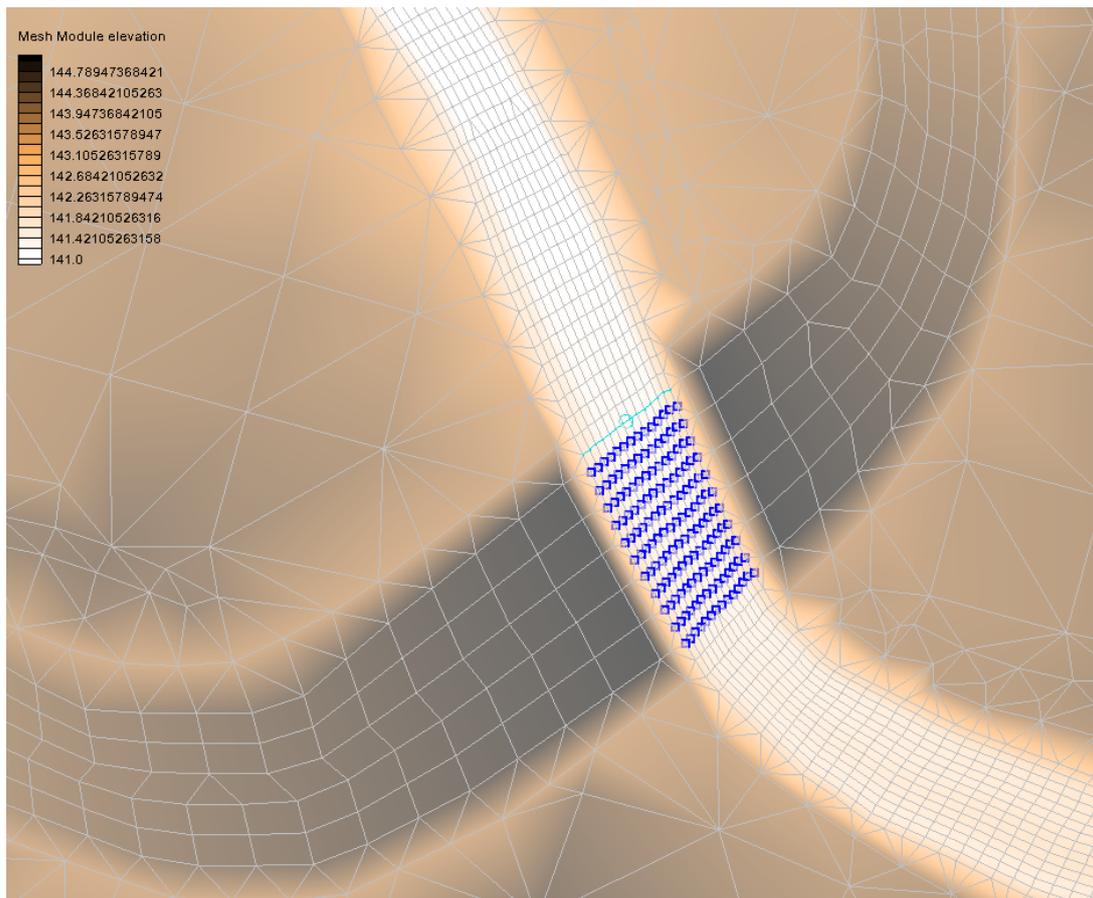


Abbildung 4.30: Markierung der Knoten mit KUK im Brückenbereich

Kleine Durchlässe, die nur eine Hauptströmungsrichtung aufweisen und für die sowohl Wandreibungs-

als auch Einlauf- und Auslaufverluste berücksichtigt werden sollen, können über 1D-Bauwerke abgebildet werden, siehe [Kapitel 4.5.10](#).

4.6.2 Pegelpunkte

Der Wasserstand, der sich während der Simulation einstellt, kann für beliebige Knoten gesondert gespeichert werden. Dazu werden Pegelpunkte angelegt. Zum Anlegen der Pegelpunkte werden ein oder mehrere ausgewählte Knoten des Berechnungsnetzes markiert und mittels Assign BC als Pegelpunkt definiert. Die an den Kontrollpunkten ermittelten Wasserstände werden in der Datei pegel.dat für jeden Q_Strg-Zeitschritt abgespeichert.

Für Pegelpunkte können im Dialog Assign BC Namen (max. 12 Zeichen) vergeben werden. Wird der Name in SMS nicht gesetzt, wird die Nummer des Knotens als Default-Name benutzt. Ist der Name länger als 12 Zeichen werden nur die ersten 12 Zeichen als Name übernommen. Die Sortierung der Ausgabe in der pegel.dat erfolgt durch die (ggf. gekürzten) Namen in alphabetischer Reihenfolge.

4.6.3 Quellterme an Netzknoten: Niederschläge als Zeitreihen in SMS

Sie können in SMS Niederschlagsdaten als Zeitreihen eingeben. Diese werden von HYDRO_AS- 2D als Quellterme genutzt.

Der Haupteinsatz für diese Option ist die Simulation von Starkregenereignissen in urbanen Gebieten. Für die Modellierung eines Gewässers oder eines Gewässerabschnittes werden Sie normalerweise keine Niederschlagsdaten hinterlegen. Ziel dieser Option ist es nicht, die Modellierung von Gewässern „genauer“ zu machen, sondern die einfache Modellierung von Starkregenereignissen in urbanen Gebieten zu unterstützen.

In einem häufigen und sehr einfach zu modellierenden Fall wird für ein kleines Modellgebiet eine einzige Niederschlagszeitreihe mit der Nummer 0 vorgegeben. Dann wird das gesamte Gebiet mit dieser einen Reihe belastet.

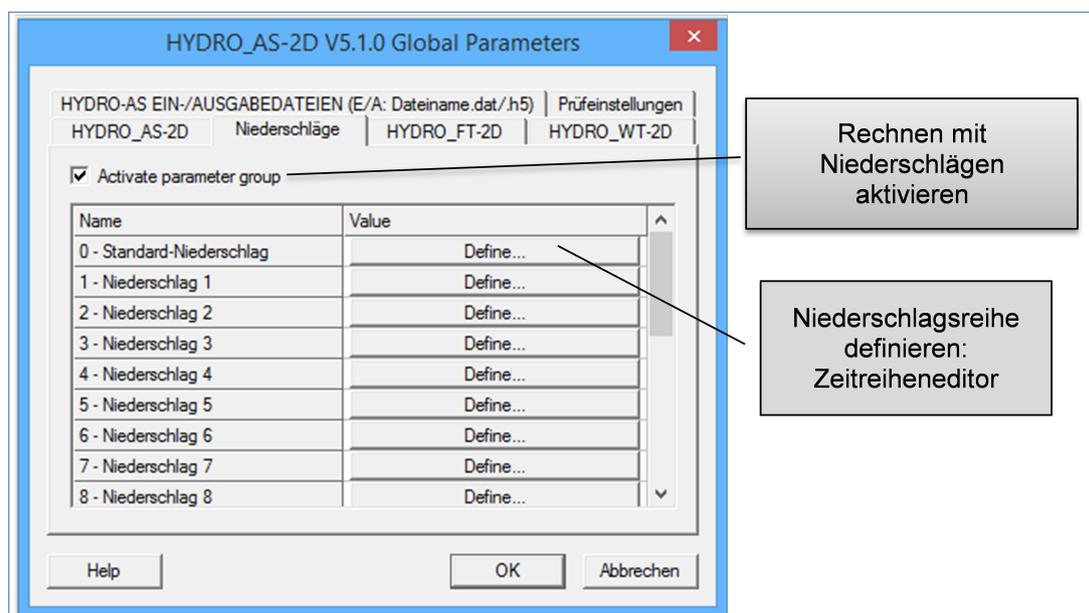


Abbildung 4.31: Eingabe der Niederschlagsreihen

Sie können aber auch bis zu 20 weitere Zeitreihen vorgeben und diese Reihen Gruppen von selektierten Knoten zuweisen, um eine räumliche Differenzierung des Niederschlages zu berücksichtigen.

Diese Option wählen Sie bei größeren Gebieten.

Die Modellierung mit nur einer einzigen Reihe ist erheblich einfacher, da Sie in diesem Fall auf die Zuweisung einzelner Reihen zu Knoten verzichten können. Bei der Modellierung mit mehreren Zeitreihen müssen Sie nicht nur die verschiedenen Zeitreihen angeben, sondern auch die Zuordnung der Reihen zu den einzelnen Knoten durchführen.

Die Niederschlagszeitreihe bearbeiten Sie im Reiheneditor von SMS.

Bitte, beachten Sie:

- Die Niederschlagswerte werden als Intensitäten in der Einheit **mm/h** angegeben.
- Sie können die Niederschlagswerte in freien Zeitschritten angeben. HYDRO_AS-2D interpoliert die Werte für jeden der internen Berechnungsschritte linear.
- Sie können in SMS die Reihenwerte auch aus einer Datei mithilfe des Import-Buttons im Reiheneditor importieren, sodass Sie die Daten nicht eintippen müssen.
- Geht die Simulationsdauer über den letzten angegebenen Zeitpunkt einer Niederschlagszeitreihe hinaus, werden die Niederschlags-Werte für alle Zeitpunkte nach dem letzten Eintrag auf null gesetzt.

Die Zuordnung der Niederschlagszeitreihen zu den Knoten erfolgt durch die Datei `nodeniederschlag.(dat/h5)`. Diese Datei enthält die Nummern der Niederschlagszeitreihen in der Reihenfolge der Knoten, vgl. z.B. `wtiefe_0.(dat/h5)`, [Kapitel 4.7](#).

Da die Zuweisung von Flächeninformationen auf die Knoten in SMS nicht direkt möglich ist, können Sie zur Erstellung dieser Datei wie folgt vorgehen:

- Zunächst öffnen Sie das GIS-Shape oder die CAD-Datei/Map-Datei, der die Niederschläge in der Fläche unterscheidet, in SMS. Jeder Fläche muss eine Niederschlagszeitreihe zugeordnet sein. Dabei kann jede Niederschlagszeitreihe mehreren Flächen zugewiesen sein.
- Über den Menüpunkt Mapping / Shapes ⇒ Feature Objects konvertieren Sie die Daten in das Map-Modul. Die verschiedenen Zeitreihen können als Klassen als Materialbereichsattribut übernommen werden.

Nun wiederholen Sie die folgenden Arbeitsschritte für alle vorhandenen Niederschlagszeitreihen (maximal 20):

- Aktivieren von Select Feature Polygon im Map-Modul.
- Menü Edit / select by ⇒ Material Type anklicken.
- Aus der Liste die erste (bzw. jeweils nächste) Klasse der Niederschlagszeitreihen selektieren (es werden ein oder mehrere Map Polygone selektiert).
- Menü Feature Objects / Delete or Select Data wählen und alle Knoten innerhalb des gewählten Polygons selektieren.
- Den Z-Wert im Mesh-Modul (Kopie der 2dm-Datei) mit der Nummer der Klasse überschreiben.

Nach Abschluss der Zuweisung sollten alle Z-Werte der Netzknoten mit Klassennummern überschrieben sein.

Abschließend exportieren Sie die Z-Werte im Binärformat oder als ASCII-Datei unter dem Namen `nodeniederschlag.(dat/h5)`. Die 2dm-Datei sollte zur Dokumentation z. B. unter dem Namen `zuordnung_niederschlag.2dm` abgespeichert werden.

Ob das ASCII-Format oder das X MDF-Format eingelesen werden soll, muss im Reiter HYDRO- AS EIN-/AUSGABEDATEIEN (E/A: Dateiname.dat/h5) in SMS gewählt werden, siehe [Kapitel 4.9](#). Der Parameter ist *Eingabedateien binär* und ist standardmäßig deaktiviert, also wird das ASCII-Format erwartet.

Hinweis: In [Kapitel 4.6.6](#) wird eine alternative Methode beschrieben, Niederschläge in HYDRO_AS-2D einzusetzen. Dabei werden Block-Niederschläge angesetzt.

4.6.4 Quellterme an Netzknoten: Konstante Zuflüsse/Abflüsse

In der Datei *Data-in/AS/eflow-in.(dat/h5)* können Quellterme für jeden Knoten angegeben werden. Diese Daten sind zeitlich **nicht** variabel und gelten über die gesamte Simulationsdauer.

Diese Datei enthält immer für alle Knoten Quellterme/Senkterme. Daher ist keine eigene Angabe in SMS für die einzelnen Knoten erforderlich.

Quellterme können auch eine Versickerung beschreiben. Ist die Datei leer oder enthält keine Daten, bleiben die Quellterme unberücksichtigt.

Ob die Terme als Quellterme oder als Senkterme interpretiert werden, wird durch die Kennzahl (0 oder ≥ 1) angegeben. Diese Zahl wird vor den Quelltermen/Senktermen eingelesen und muss in der Datei enthalten sein.

- Ist diese Zahl 0 bedeutet das, dass Quellterme in m/s angegeben sind.
Positive Zahlen sind Zugaben zum Modell.
Negative Zahlen sind Entnahmen.
- Ist diese Zahl größer oder gleich 1, werden die Werte als Senkterme in m^3/d interpretiert.
Negative Zahlen sind Zugaben zum Modell.
Positive Zahlen sind Entnahmen.

Die Quelltermdatei wird für folgende Formate unterstützt:

1. Das FEFLOW-Format: ASCII-Datei. Die Datei heißt *eflow-in.dat*. Die erste Zeile der Datei *eflow-in.dat* muss als erstes Zeichen die Kennzahl (0 oder ≥ 1) enthalten. In der ersten Zeile kann hinter der Zahl Text stehen, der nicht eingelesen wird. In den folgenden Zeilen ist für jeden Knoten ein Wert angegeben. Nachteile: Die Datei kann nicht in SMS geöffnet werden. Bei Fehlern in der Datei bricht die Simulation beim Einlesen ab.
2. Das alte ASCII-Format von SMS: Die Datei heißt *eflow-in.dat*. Die Datei enthält einen Header - analog zur *depth.dat* -, der unter anderem den Datentyp (skalar), die Knotenanzahl und den Zeitschritt beschreibt. Der Zeitschritt wird hierbei nicht als Zeit interpretiert, sondern als Kennzahl. D.h. ggf. muss der Zeitschritt auf eine Zahl ≥ 1 gesetzt werden. Vorteile: Die Datei kann in SMS geladen werden. Beim Einlesen in HYDRO_AS/ HYDRO_AS-1STEP wird die angegebene Knotenanzahl mit der Anzahl der Knoten des Modells/der 2dm-Datei verglichen. Wenn die Datei nicht zum Modell passt, wird eine Warnung ausgegeben und die Datei wird nicht eingelesen.
3. Das neue ASCII-Format von SMS: Die Datei heißt *eflow-in.dat*. Die Datei enthält einen Header, der unter anderem den Datentyp (skalar), die Knotenanzahl und den Zeitschritt beschreibt. Der Zeitschritt wird hierbei nicht als Zeit interpretiert, sondern als Kennzahl. D.h. ggf. muss der Zeitschritt auf 1 gesetzt werden. Hinweis: Der Zeitschritt wird durch „TS 0 0“ beschrieben. Hier muss die zweite Zahl angepasst werden, d.h. z.B. „TS 0 1“. Vorteile: Die Datei kann in SMS geladen werden. Die Datei kann in SMS durch den Export erstellt werden. Beim Einlesen in HYDRO_AS/HYDRO_AS-1STEP wird die angegebene Knotenanzahl mit der Anzahl der Knoten des Modells/der 2dm-Datei verglichen. Wenn die Datei nicht zum Modell passt, wird eine Warnung ausgegeben und die Datei wird nicht eingelesen.

4. Das ASCII-Format von SMS mit zusätzlichen Activity-Informationen: Die Datei heißt *fflow-in.dat*. Die Datei enthält einen Header, der unter anderem den Datentyp (skalar), die Knotenzahl, die Elementzahl und den Zeitschritt beschreibt. Zusätzlich zu den knotenbasierten Quell- oder Senktermen enthält die Datei elementbasierte Activity-Informationen, die zu Darstellung in SMS dienen. Für die Simulation spielen diese Werte keine Rolle. Der Zeitschritt wird auch hier nicht als Zeit interpretiert, sondern als Kennzahl. D.h. ggf. muss der Zeitschritt auf 1 gesetzt werden. Hinweis: Der Zeitschritt wird durch „TS 1 0“ beschrieben. Hier muss die zweite Zahl angepasst werden, d.h. z.B. „TS 1 1“. Vorteile: Die Datei kann in SMS geladen werden. Die Datei kann in SMS durch den Export erstellt werden. Beim Einlesen in HYDRO_AS/HYDRO_AS-1STEP wird die angegebene Knotenanzahl mit der Anzahl der Knoten des Modells/der 2dm-Datei verglichen. Genauso wird die Elementanzahl geprüft. Wenn die Datei nicht zum Modell passt, wird eine Warnung ausgegeben und die Datei wird nicht eingelesen.
5. Das XMDF-Format von SMS: Die Datei heißt dann *fflow-in.h5*. Sie kann in SMS erstellt werden. Vorteile: Die Datei kann in SMS geladen werden. Die Datei kann in SMS durch den Export erstellt werden. Beim Einlesen in HYDRO_AS/HYDRO_AS-1STEP wird die angegebene Knotenanzahl mit der Anzahl der Knoten des Modells/der 2dm-Datei verglichen. Wenn die Datei nicht zum Modell passt, wird eine Warnung ausgegeben und die Datei wird nicht eingelesen. Wie Sie in SMS ein XMDF-Dateien erstellen können, die einen von Ihnen gewählten Zeitschritt beinhalten, ist unten beschrieben.

Ob das ASCII-Format oder das XMDF-Format eingelesen werden soll, muss im Reiter HYDRO- AS EIN-/AUSGABEDATEIEN (E/A: Dateiname.dat/h5) in SMS gewählt werden, siehe [Kapitel 4.9](#). Der Parameter ist *Eingabedateien binär* und ist standardmäßig deaktiviert, also wird das ASCII-Format erwartet.

Erstellen von XMDF-Dateien in SMS mit vorgegebenem Zeitschritt

Fall 1: Der Zeitschritt soll null sein.

Dazu erstellen Sie die Datei wie eine *wtiefe_0* in SMS, siehe [Kapitel 4.7](#). Wichtig: Nutzen Sie hier explizit **keine** Datensätze als Grundlage, die Zeitschritte enthalten.

Fall2: Der Zeitschritt soll größer oder gleich 1 sein:

Um eine XMDF-Datei mit einem vorgegebenen Zeitschritt zu erstellen, gibt es mehrere Wege. Hier werden zwei Wege vorgestellt. Der erste Weg ist sehr einfach und nutzt das Programm HDFView (<https://www.hdfgroup.org/downloads/hdfview/>). Der zweite Weg erfolgt ohne HDFView in SMS.

Mit HDFView:

- Falls noch nicht vorhanden, installieren Sie das Programm HDFView.
- Öffnen Sie die mit Zeitschritt null erstellte XMDF-Datei in HDFView z.B. per „drag and drop“.
- Klappen Sie den TreeView am linken Rand auf, so dass Sie die Informationen zum Zeitschritt *Times* finden.

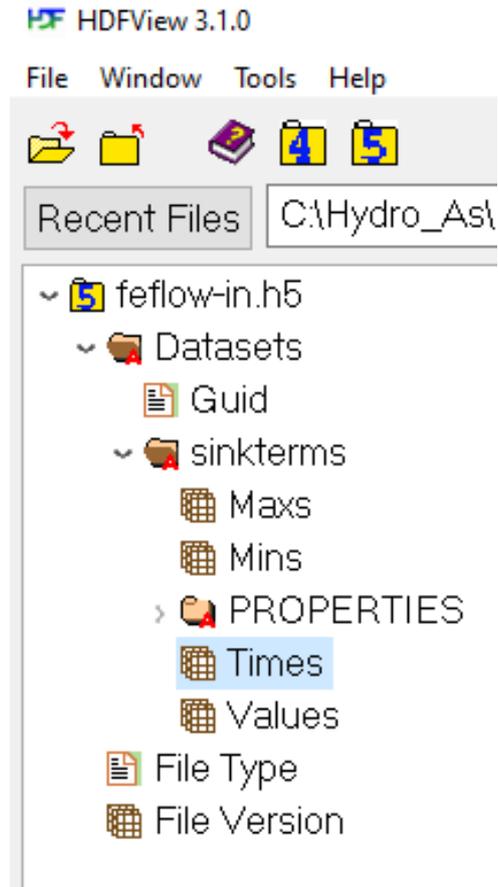


Abbildung 4.32: TreeView in HDFView

- Öffnen Sie *Times* per Doppelklick und editieren Sie den Zeitschritt.
- Falls Sie den Zeitschritt nicht editieren können, laden Sie die Datei als editierbar neu, indem Sie per rechter Maustaste auf den Dateinamen im TreeView das Menü öffnen und *Reload File As -> Read/Write* auswählen.

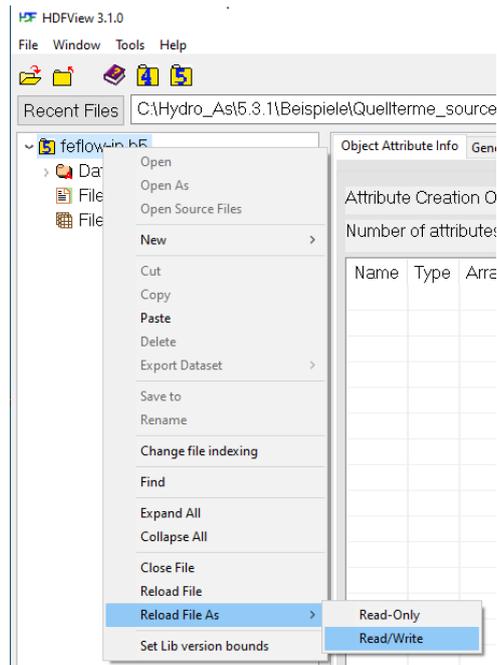


Abbildung 4.33: Datei in HDFView neu laden

Mit SMS:

- Simulieren Sie Ihr Modell (ohne *feflow-in*) mit einer Simulationszeit von 10 Sekunden. Dabei muss mindestens eine der skalaren Ausgabedateien aktiviert sein, beispielsweise die Ausgabe der Wassertiefen *depth*.
- Laden Sie das Modell und die *depth*-Datei (als ASCII- oder Binärdatei) in SMS.
- Erstellen Sie die Einträge für die *feflow-in*-Datei in SMS und benennen Sie diesen Datensatz, z.B. *feflow0*.
- Erzeugen Sie im *Data Calculator* einen neuen Datensatz *feflow-in*. Dabei nutzen Sie Ihren vorher erstellten Datensatz *feflow0* und die *depth*-Datei wie folgt:
- Wählen Sie im *Data Calculator* die *depth*-Datei aus und aktivieren Sie den Haken bei *use all time steps*. Berechnen Sie nun *depth-depth + feflow0*. Der neue Datensatz *feflow-in* enthält so die selben Daten wie der Datensatz *feflow0*, aber alle Zeitschritte der *depth*-Datei.
- Exportieren Sie den Datensatz *feflow-in* für den Zeitschritt 10 Sekunden als X MDF-Datei unter dem Namen *feflow-in.h5*.
- Sie können den Zeitschritt der exportierten Datei prüfen, indem Sie die Datei in SMS laden und im *Data Calculator* für diese Datei *Data Set Info...* aufrufen.
- Hinweis: Auf diese Weise können Sie auch eine ASCII-Datei *feflow-in.dat* exportieren, ohne den Zeitschritt nachträglich manuell zu bearbeiten.

4.6.5 Quellterme an Netzknoten: Zufluss-/Abflusszeitreihen (Schächte)

Zur Interaktion mit einem Kanalsystem wurde die Möglichkeit geschaffen an einzelnen Knoten – bei Schächten oder Regenwassereinflüssen – zeitabhängige Zuflüsse und Abflüsse vorzugeben.

Diese Zuflüsse und/oder Abflüsse werden dem Knoten über eine Kennung zugewiesen. Dazu wird in SMS für den Knoten die Randbedingung *Zufluss/Abfluss* gesetzt und die Kennung eingetragen. Das

Beispiel zeigt, wie der Schacht mit der Kennung SchachtA einem Knoten zugeordnet wird.



Abbildung 4.34: Einem Knoten wird die Kennung SchachtA zugeordnet

Die Zuflüsse oder Abflüsse werden in der Einheit m^3/s in einer gesonderten Datei im Verzeichnis `Data-in` angegeben. Per Default heißt diese Datei `Data-in/AS/nodesources.dat`.

Diese Datei enthält für alle Schächte/Regeneinläufe die Zuflüsse/Abflüsse in m^3/s . Positive Werte werden als Zuflüsse/Quellterme und negative Werte als Abflüsse/Senkterme interpretiert.

Das Format der `nodesources.dat`-Datei enthält Schachtkennung als String und Modellzuflüsse für beliebige Zeitschritte. Das Beispielmmodell „Quellterme_Schaechte“ enthält eine Beispieldatei:

TS	SchachtA	SchachtB	SchachtC
0	0	0	0
100	1.237	-0.0317	1.8621
200	2.397	0.11725	2.7452
300	3.408	0.42993	2.3057
400	4.207	0.87909	1.0556
500	4.744	1.42903	0.0208
600	4.987	2.03778	0.0650
700	4.919	2.65973	1.3192
800	4.546	3.24844	3.0921
900	3.890	3.75952	4.3209

- Die erste Zeile enthält die Kennungen (TS (für Time Step), Schachtkennung, Kennungen der Regeneinläufe). Im Beispiel sind diese SchachtA, SchachtB und SchachtC.
- Die folgenden Zeilen enthalten für aufsteigende Zeitpunkte Abflusswerte zu den Kennungen. Die Zeitpunkte müssen NICHT äquidistant verteilt sein und stehen in der Einheit „Sekunden“ in der ersten Spalte.
- Die Zuflüsse/Abflüsse in m^3/s stehen in den folgenden Spalten.
- Die Zuflüsse/Abflüsse müssen für alle Schächte/Regeneinläufe zu jedem angegebenen Zeitpunkt angegeben werden. Ggf. müssen daher Werte interpoliert werden.
- Innerhalb einer Zeile sind die einzelnen Einträge durch beliebig viele Leerzeichen oder Tabulatoren zu trennen.

HYDRO_AS-2D berechnet für jeden internen Zeitschritt die aktuellen Werte für die Knoten durch lineare Interpolation.

Die Zeitpunkte müssen in aufsteigender Folge vereinbart werden. Die Abstände zwischen Zeitpunkten sind allerdings beliebig.

Geht die Simulationsdauer über den letzten in der *nodesources.dat* angegebenen Zeitpunkt hinaus, werden die Zufluss/Abfluss-Werte für alle Zeitpunkte nach dem letzten Eintrag auf null gesetzt.

4.6.6 Quellterme an Netzknoten: zeitabhängige Zuflüsse (sources-in)

In der Datei *Data-in/AS/sources-in.(dat/h5)* können Quellterme/Senkterme als Intensität in **mm/h** für jeden Knoten und beliebige Zeitschritte angegeben werden. Der Aufbau der *sources-in*-Datei ist zu einer *depth.dat* im ASCII-Format oder einer *depth.h5* im XMDF-Format identisch. Für jeden Knoten des Berechnungsnetzes ist pro Zeitpunkt TS ein Wert für den Quellterm in **mm/h** angegeben. Die Wahl der Zeitpunkte ist beliebig, muss aber aufsteigend sein. Zwischen den Zeitpunkten wird im Gegensatz zu den Niederschlagszeitreihen in SMS (siehe [Kapitel 4.6.3](#)) **nicht interpoliert, sondern** der Wert gilt bis zum nächsten in der Datei angegebenen Zeitpunkt. Es wird also mit **Blockwerten** gerechnet.

Falls der letzte in der *sources-in*-Datei angegebene Zeitpunkt kleiner ist als das Simulationsende, gelten die Quellterme/Senkterme den folgenden internen Zeitschritt lang und werden anschließend auf null gesetzt. Sollen die Quellterme/Senkterme für eine bestimmte Dauer gelten, muss das Ende des Blocks durch einen Eintrag von Null-Werten zum gewünschten Endzeitpunkt definiert werden.

Ob das ASCII-Format oder das XMDF-Format eingelesen werden soll, muss im Reiter HYDRO- AS EIN-/AUSGABEDATEIEN (E/A: Dateiname.dat/h5) in SMS gewählt werden, siehe [Kapitel 4.9](#). Der Parameter ist *Eingabedateien binär* und ist standardmäßig deaktiviert, also wird das ASCII-Format erwartet.

4.7 Anfangswasserstand

In HYDRO_AS-2D besteht die Möglichkeit, die Berechnung mit einem Anfangswasserstand zu starten, anstatt die Berechnung mit einer trockenen Sohle zu beginnen. Hierzu muss die Datei *wtiefe_0.(dat/h5)* im Verzeichnis *Data-in\AS* erzeugt werden.

Die Ordnerstruktur und Benennung der Eingangsdateien finden Sie im Ordner SMS-2DM-Vorlage Ihres HYDRO_AS-2D-Installationsverzeichnisses. Kopieren Sie dieses *Data-in*-Verzeichnis neben ihre *2dm*-Datei, um die richtige Struktur und Benennung der Dateien sicher zu stellen. Ersetzen Sie anschließend die entsprechenden Dateien, wie folgt:

Es sind zwei Vorgehensweisen möglich:

- a) Sie nutzen die Daten einer bereits durchgeführten Berechnung, indem Sie die Datei *depth.(dat/h5)* importieren und für den maßgebenden Zeitschritt mittels Rechtsklick und dem Befehl *Export...* die Tiefen als Anfangswasserstände in die Datei *wtiefe_0.(dat/h5)* speichert. Die Datei kann im Binärformat oder ASCII-Format gespeichert werden. Falls Sie Änderungen am Netz vornehmen müssen, laden Sie die Datei *depth.(dat/h5)* und erstellen einen Scatter, bevor Sie die Änderungen durchführen. Nach Änderung des Netzes interpolieren Sie die *depth* unter Verwendung des Scatter Point Moduls auf das neue Berechnungsnetz. Näheres über den Einsatz des Scatter Point Moduls kann dem SMS Benutzerhandbuch entnommen werden.
- b) Sie erzeugen die Datei *wtiefe_0.(dat/h5)* mittels folgender Arbeitsschritte selbst: Im folgenden Beispiel soll der Fläche des dargestellten Sees eine einheitliche Wasserspiegellage von 143 (absolute Höhe in m) als Anfangsrandbedingung zugewiesen werden.

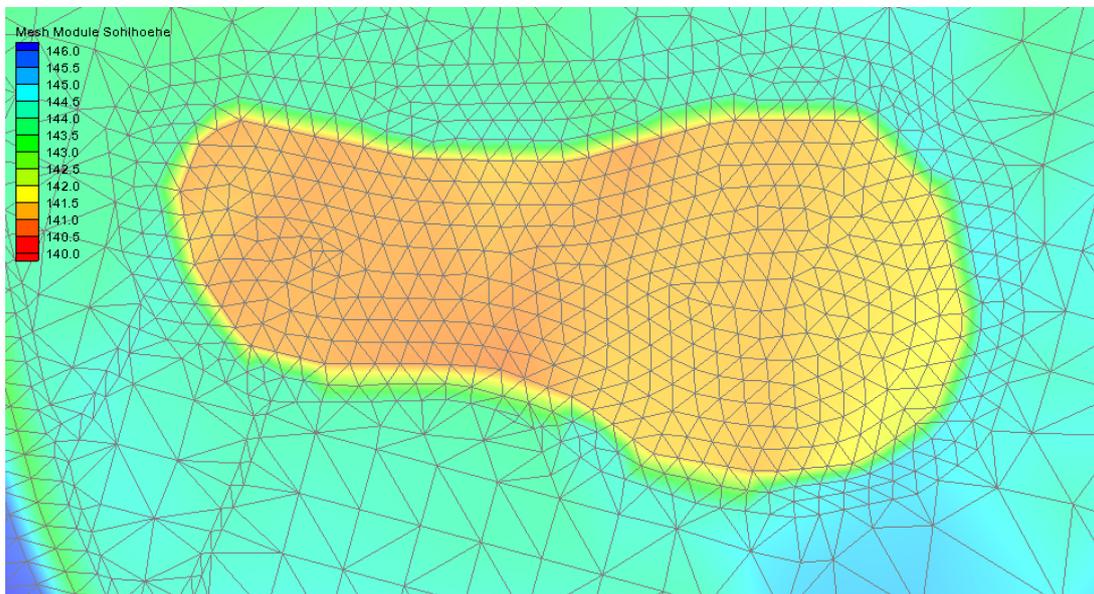


Abbildung 4.35: Beispiel: See im 2D-Modell

Generieren Sie zunächst einen Arc um den See, bei dem alle Knoten die gewünschte Höhe aufweisen. Wandeln Sie dann die Map in ein Scatter Set um. In der folgenden Abbildung sind die Elemente des Scatter Sets grün dargestellt. Die Darstellungsweise kann in den Display Options verändert werden

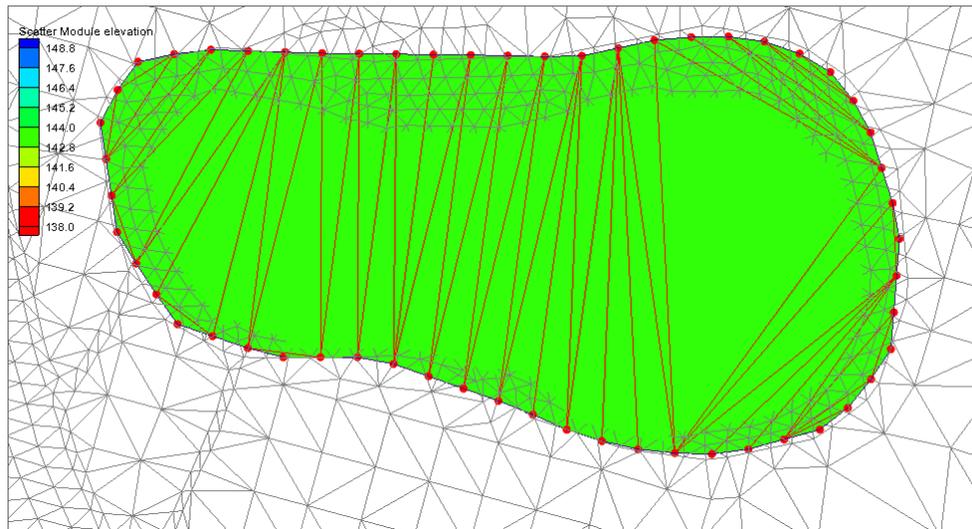


Abbildung 4.36: Scatter Set um den See

Interpolieren Sie bei aktiviertem Scatter Set über den Menüpunkt Scatter und dem Befehl Interpolate to mesh die Höhen des Scatter Sets auf das Berechnungsgitter. Für die Extrapolation stellen Sie die Option Existing Dataset Value ein.

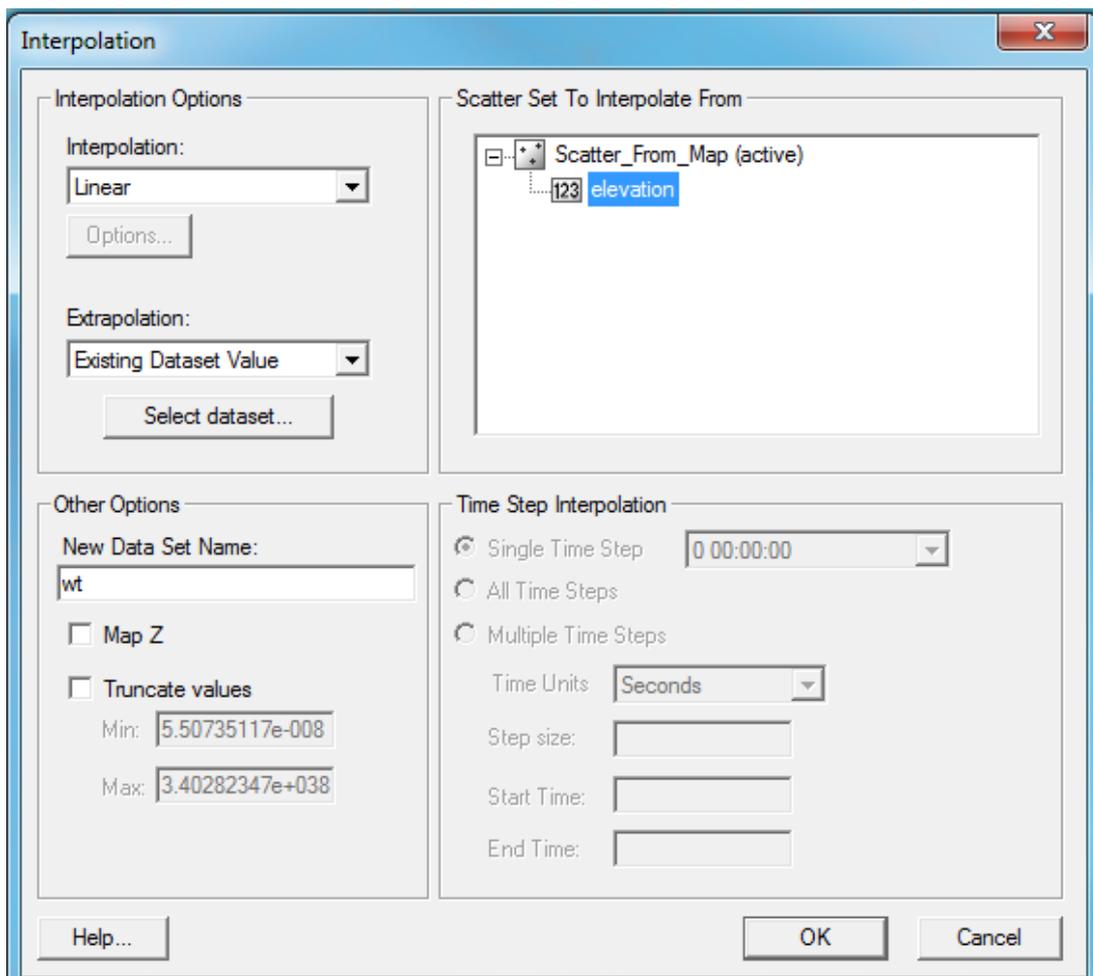


Abbildung 4.37: Dialog zur Interpolation von Scatter Informationen

Nun nutzen Sie den Data Calculator im Menü Data, um die Differenz des soeben erzeugten Datensatzes „wt“ und der Sohlhöhe (elevation) zu bilden und erhalten so die Wassertiefe an jedem Netzknoten.

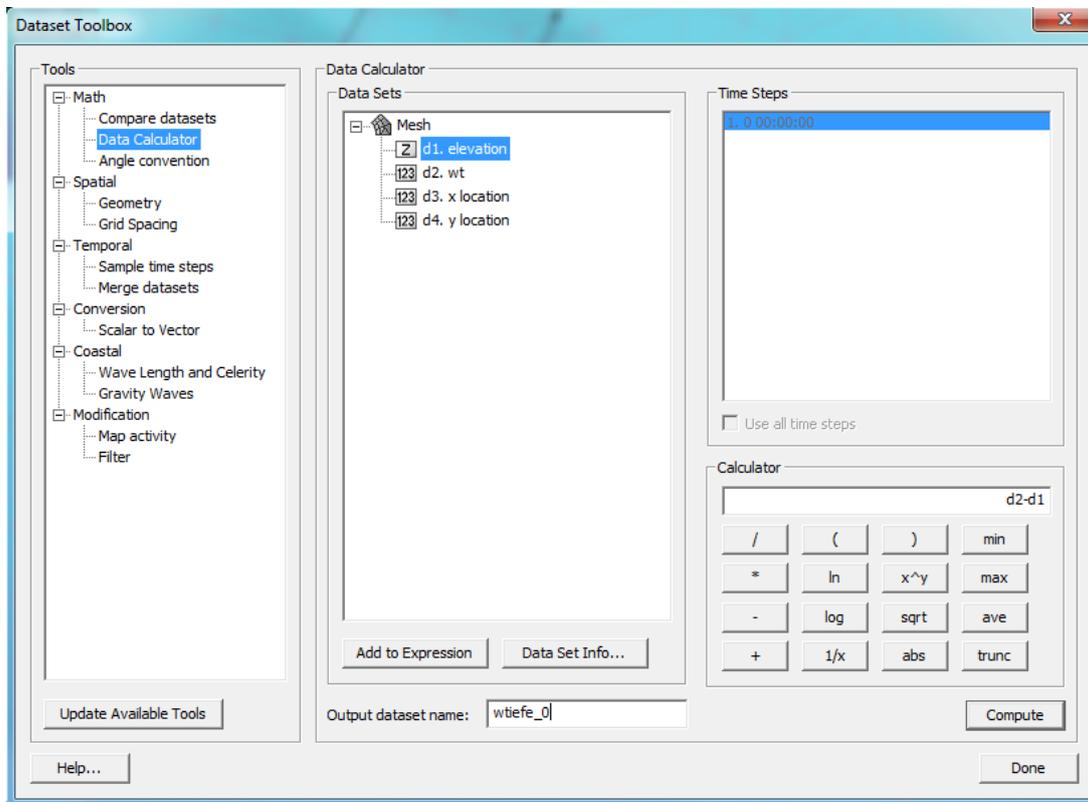


Abbildung 4.38: Nutzung des Data Calculators

Dieser Datensatz wird mit *wtiefe_0* bezeichnet und zum Schluss exportiert. Die exportierte Datei wird als *wtiefe_0.(dat/h5)* im Verzeichnis Data-in\AS abgelegt.

Ob das ASCII-Format oder das X MDF-Format eingelesen werden soll, muss im Reiter HYDRO- AS EIN-/AUSGABEDATEIEN (E/A: Dateiname.dat/h5) in SMS gewählt werden, siehe [Kapitel 4.9](#). Der Parameter ist *Eingabedateien binär* und ist standardmäßig deaktiviert, also wird das ASCII-Format erwartet.

4.8 Anfangsgeschwindigkeit

Ähnlich zur Anfangswassertiefe kann eine Anfangsgeschwindigkeit angegeben werden. Dafür legen Sie die Datei *geschw_0.(dat/h5)* im Verzeichnis Data-in\AS an. Die Datei kann im Binärformat oder ASCII-Format gespeichert werden. Wichtig ist, dass es in jedem Fall ein Vector Data Set verwendet werden muss. Zur Erstellung einer Anfangsgeschwindigkeit können die Ergebnisse der Dateien *veloc.(dat/h5)* oder *veloc_max.(dat/h5)*, falls vorhanden, aus einer vorherigen Berechnung genutzt werden oder die Datei kann ähnlich zur *wtiefe_0.(dat/h5)* manuell erzeugt werden.

Ob das ASCII-Format oder das X MDF-Format eingelesen werden soll, muss im Reiter HYDRO- AS EIN-/AUSGABEDATEIEN (E/A: Dateiname.dat/h5) in SMS gewählt werden, siehe [Kapitel 4.9](#). Der Parameter ist *Eingabedateien binär* und ist standardmäßig deaktiviert, also wird das ASCII-Format erwartet.

4.9 Dateiformate der Eingangsdateien

HYDRO_AS-2D, die Tools h5todat und dattoh5 und MapView unterstützen verschiedene Dateiformate als Eingangsdateien.

- Das alte ASCII-Format von SMS: Die Datei enthält einen Header, der unter anderem den Datentyp (skalar oder Vektor), die Knotenanzahl und den Zeitschritt beschreibt. Dieses Format entspricht auch den ASCII-Ausgabedateien von HYDRO_AS-2D.
- Das neue ASCII-Format von SMS: Die Datei enthält einen Header, der unter anderem den Datentyp (skalar oder Vektor), die Knotenanzahl, die Elementanzahl und den Zeitschritt beschreibt. Die Zeitschrittangaben enthalten zusätzlich zur Zeitangabe ein Flag, das angibt, ob Elementdaten vorhanden sind oder nicht. Ist dieses Flag auf 1 gestellt, müssen Elementdaten folgen. Diese Daten dienen zur Darstellung in SMS und werden von HYDRO_AS-2D, dattoh5 und MapView ignoriert. Ist das Flag auf 0 gestellt, dürfen keine Elementdaten folgen. Die Knotenwerte werden direkt eingelesen.
- Das Binärfomat der HYDRO_AS-2D-Ergebnisdateien: Dieses Format ist in Kapitel 5.7.4 näher beschrieben.
- Das X MDF-Format von SMS: In diesem Format wird zuerst eine Gruppe *Datasets* erstellt, in der sich dann die Daten in weitere Gruppen aufgeteilt befinden.

In welchem Format (ASCII oder binär) die Eingabedateien eingelesen werden sollen, wird in SMS im Reiter HYDRO- AS EIN-/AUSGABEDATEIEN (E/A: Dateiname.dat/h5) über den Parameter *Eingabedateien binär* angegeben.

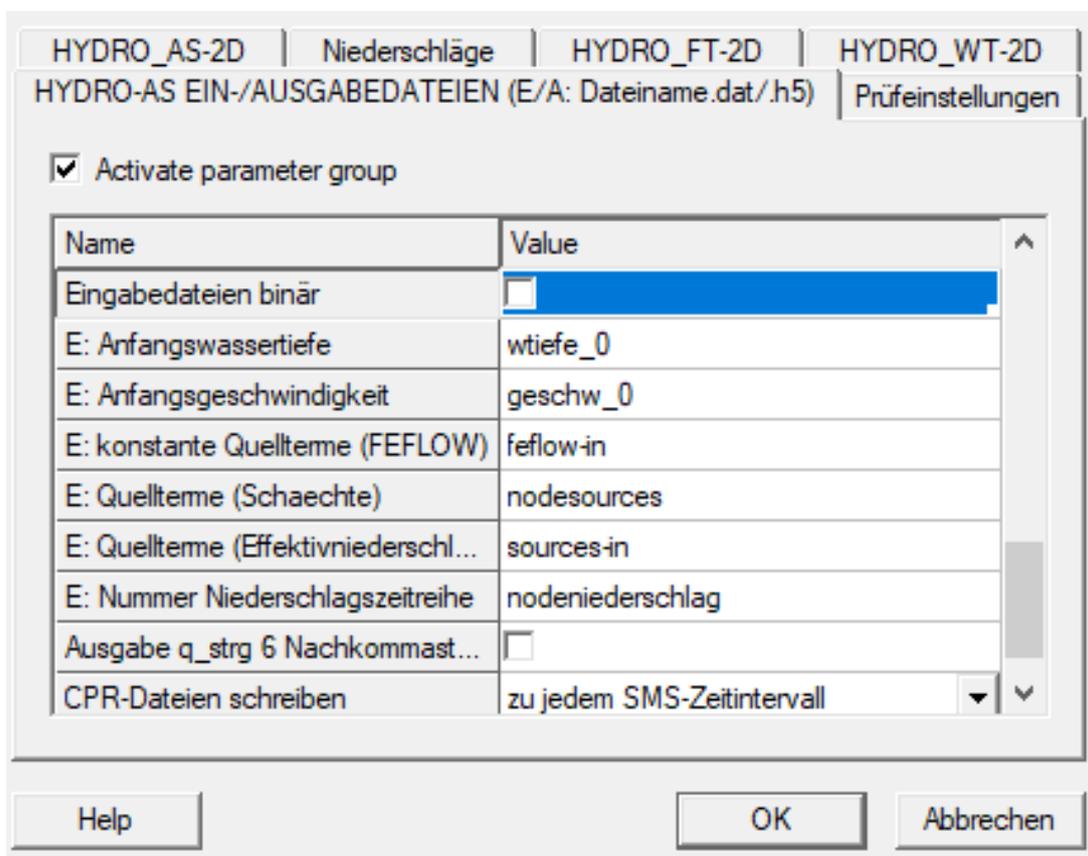


Abbildung 4.39: Parameter *Eingabedateien binär*

Dieser Parameter gilt für **alle** Eingabedateien - bis auf nodesources.dat - und ist standardmäßig deaktiviert, also wird das ASCII-Format erwartet. Folgende Konstellationen können auftreten:

Parameter/Datei	ASCII (.dat)	binär (.h5)
ASCII (.dat)	Dateiname.dat wird eingelesen	Warnung, statt Dateiname.dat wird Dateiname.h5 eingelesen
binär (.h5)	Warnung, statt Dateiname.h5 wird Dateiname.dat eingelesen	Dateiname.h5 wird eingelesen

Falls im Verzeichnis *Data-in/AS* für einen Typ von Eingangsdaten sowohl die ASCII-Datei (.dat) als auch die binäre Datei (.h5) vorliegt, wird die Datei gewählt, die über den Parameter *Eingabedateien binär* vorgegeben wird. Falls die Datei älter ist als die nicht verwendete Datei, wird ein Fehler ausgegeben. Um potentielle Fehler zu vermeiden, achten Sie bitte darauf, dass in dem Verzeichnis nur die Dateien liegen, die auch an den Rechenkern übergeben werden sollen.

4.10 Netzqualität

Das Mesh Modul in SMS enthält eine Prüfoption, mit der die Qualität des erzeugten Berechnungsnetzes kontrolliert werden kann. Diese Option – Mesh Quality - ist unter dem Display Menü zu finden (Display ⇒ Display Options ⇒ Mesh quality).

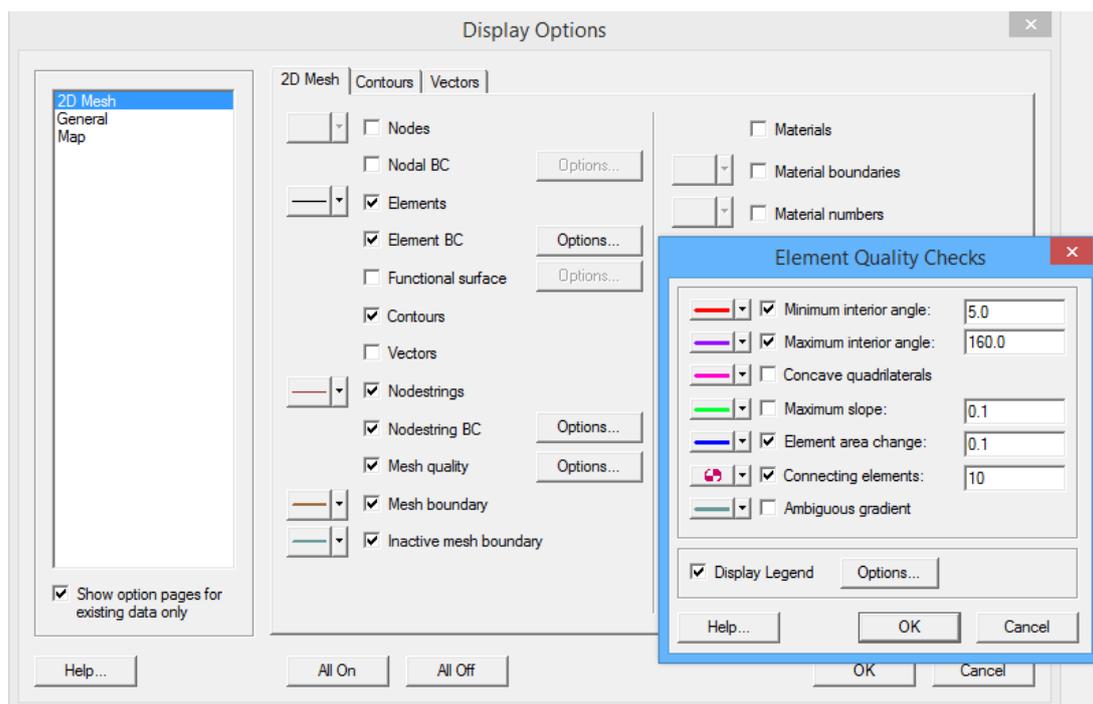


Abbildung 4.40: Dialog Mesh Quality

In der Praxis sollten in der Regel folgende Kriterien für die Modellierung verwendet werden:

- Minimaler innerer Winkel 5°
- Maximaler innerer Winkel 160°
- Nicht mehr als 10 Elemente sollten in einem Knoten anschließen
- Verhältnis benachbarter Elementflächen größer als 0,1

Diese Einstellung ist in der obigen Abbildung dargestellt.

Der Präprozessor liefert zudem Hinweise und Warnungen zur Netzqualität (siehe Kapitel 5 und Kapitel 6).

HYDRO_AS-2D erzielt die besten Ergebnisse, wenn mit einem an die jeweilige Strömungssituation angepassten Netz gerechnet wird. D.h., wenn die Netzlinien approximativ parallel bzw. senkrecht zum betreffenden Strömungsvektor verlaufen, ist die Interpolation in den Berechnungen am genauesten. Dies sollte zumindest im Flussschlauch und in allen weiteren strömungsrelevanten Bereichen der Fall sein.

Möglichkeiten ein Berechnungsnetz (Flussschlauch) mit parallel zur Strömungsrichtung verlaufender Netzstruktur zu erzeugen sind:

- In SMS gibt es die Option Rectangular Patch im Menü Elements. Diese Option existiert lediglich in SMS 13.0 nicht. Mit der Version 13.1 ist sie wieder verfügbar. Bei der Verwendung dieser Option ist zu beachten, dass die Sohlentopographie durch Höhenwerte an allen vier den Flussschlauch begrenzenden definierten Nodestings beeinflusst wird. Es empfiehlt sich, die Uferlinien für die Bestimmung des Flussverlaufs zu verwenden, da sie i.d.R. eine ausreichende Glattheit aufweisen.
- In SMS (ab Version 13 und folgende) kann der Flussschlauch im Map Modul konstruiert werden. Hier wird für die Polygone der Vermaschungstyp *patch* gewählt. Achten Sie darauf, dass die Verteilung der *vertices* auf den *arcs* (annähernd) Rechtecke erzeugen.
- Durch die Nutzung des Flussschlauchgenerators wird der Flussschlauch entsprechend der Kriterien für eine optimale HYDRO_AS-2D Simulation erzeugt.

Bei der zweidimensionalen Modellierung von größeren Bauwerken ist darauf zu achten, dass das Berechnungsnetz in diesem Bereich geometrisch an die zu erwartende Strömungssituation angepasst ist. Die Verwendung einer entsprechenden Elementgröße mit glatten Übergängen zum umliegenden Berechnungsnetz sollte zu einer ausreichend genauen Modellierung in diesem Bereich führen. Verwenden Sie nach Möglichkeit mindestens 3 bis 4 Elemente in Strömungsrichtung und 6 bis 8 Elemente senkrecht zur Strömung, damit der Strömungsvorgang genügend genau erfasst werden kann.

Bei der Abbildung von Bauwerken mittels einer empirischen Formel (1D-Elemente oder W/Q-Beziehungen) muss ebenfalls ein geometrisch passendes Netz aufgestellt werden. Dabei sollten die Einlauf- und Auslaufbereiche möglichst gleichmäßig gestaltet werden. Ein geometrisch unpassendes oder unregelmäßiges Netz kann oft zu unregelmäßigen Wasserspiegellagen und Fließgeschwindigkeiten in der Umgebung einer solchen Stelle führen.

Es empfiehlt sich besonders am Anfang an kleinen Modellausschnitten zu üben, wie das Netz in diesen Bereichen zu gestalten ist. Untersuchen Sie mögliche Auswirkungen unterschiedlicher Netzgestaltungen bei verschiedenen Bauwerkstypen auf das Berechnungsergebnis durch sukzessive Netzanpassung und Verfeinerung. Dadurch werden Sie mehr Sicherheit beim praktischen Einsatz gewinnen.

Werden Seen bzw. Bereiche mit hohen Wassertiefen modelliert, so sollten die gewählten Elementgrößen begrenzt werden. Es wird empfohlen, dass die Grundfläche eines Kontrollvolumens so gewählt wird, dass das Produkt aus Grundfläche und Wassertiefe deutlich weniger als 10.000 m^3 beträgt.

4.11 Räumliche Auflösung

Die räumliche Auflösung des Netzes hat Einfluss auf die Genauigkeit der Ergebnisse. Auch wenn zu kleine Elemente und Knotenabstände - wie oben beschrieben - vermieden werden müssen, darf das Netz in hydraulisch relevanten Bereichen nicht zu grob sein. Die Elementgrößen und Knotenabstände hängen dabei vom Untersuchungsgebiet und der Fragestellung ab. Daher ist es nicht möglich eine

allgemeine Faustregel anzugeben. Ein wichtiger Aspekt für die räumliche Auflösung ist die Geländetopologie. Ist der Gradient (die Änderung) der Höhe Z in eine bestimmte Richtung hoch, z.B. an Böschungen, sind geringere Knotenabstände in diese Richtung erforderlich. Anderfalls können Fehler in der Wassertiefe entstehen.

4.12 Speichern von Modelldaten

Die Angaben zur Abflusssimulation können mit Save HYDRO_AS-2D, Save Project oder mit Save as (*.2dm) im 2dm-Format abgespeichert werden.

Hinweis: Die abgespeicherten Daten zum Simulationslauf können über File → Open erneut in SMS geladen werden.

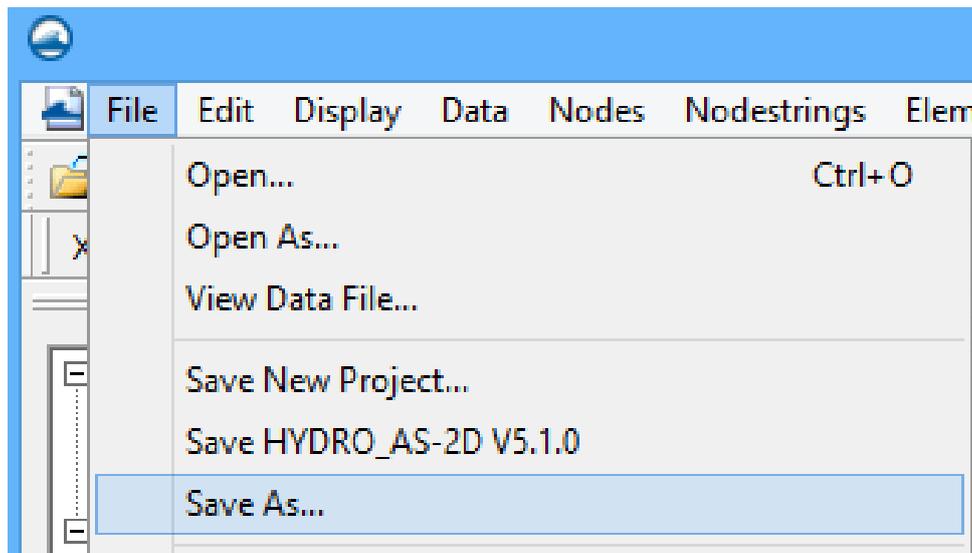


Abbildung 4.41: Menü nach dem Aufruf des Save as Dialogs

4.13 Berechnungskontrolle und Sensitivitätsuntersuchung

Nicht nur aus Genauigkeitsgründen wird empfohlen, kritische Stellen im Modell (Bauwerke, ufernahe Bereiche, Überflutungsgrenzen) detaillierter auf Plausibilität zu prüfen. Dies betrifft insbesondere diejenigen Stellen im Modell, die den Abfluss-/ Überflutungsvorgang stark beeinflussen können. Im gleichen Zusammenhang empfiehlt es sich, die Berechnung einer sog. Sensitivitätsuntersuchung durch veränderte Rauheitswerte zu unterziehen. Damit wird zum einen die mögliche Schwankungsbreite der hydraulischen Berechnung näher aufgezeigt. Zum anderen hilft dies häufig beim Lokalisieren kritischer Stellen im Gebiet und bringt gleichzeitig zusätzliche Sicherheit für das Berechnungsergebnis. Sollten dabei größere Unterschiede in den ermittelten Ergebnissen herauskommen, müssen diese genauer geprüft und gegebenenfalls entsprechend korrigiert werden.

Hinweis: Die Leistungsfähigkeit verschiedener Bauwerke kann mittels empirischer Ansätze relativ einfach auf Plausibilität geprüft werden.

Starke Geländeneigungen können besonders dann zu Ungenauigkeiten im Modell führen, wenn eine bedeutende Strömung in dieser Richtung existiert und das Wasserspiegelgefälle relativ groß ist. Überprüfen Sie deshalb sorgfältig Stellen im Gebiet mit großem Wasserspiegelgefälle und mit großen Strömungsgradienten.

5 Durchführung der Berechnung

Zur Nutzung von HYDRO_AS-2D schalten Sie zunächst den Softwareschutz frei.

Das Simulationsmodell HYDRO_AS-2D besteht aus zwei Hauptmodulen:

- Präprozessor: HYDRO_2DM
- Simulationsprogramme: HYDRO_AS oder HYDRO_AS-1STEP (Modulabhängig)

Das Programm HYDRO_2DM muss immer zuerst aufgerufen werden, denn mit diesem Modul werden die Ausgabedaten aus SMS – sowie andere für die Berechnung erforderliche Daten, die nicht mittels SMS eingegeben werden können - zum Einlesen in das Berechnungsmodul HYDRO_AS oder HYDRO_AS-1STEP vorbereitet und bereitgestellt. Zudem werden über HYDRO_2DM alle Daten auf Qualität und Konsistenz geprüft und ggf. Fehlermeldungen und Warnungen herausgegeben, die in die Datei `hydro_2dm.mel` geschrieben werden. Wurden von HYDRO_2DM Fehler in den Eingangsdaten festgestellt, kann das Berechnungsprogramm HYDRO_AS / HYDRO_AS-1STEP nicht gestartet werden.

Alle Programme können auf verschiedene Weise aufgerufen werden:

- über die Kommandozeile: direkt in der Konsole (Windows und Linux)
- Batch-Dateien: Unter Windows stehen Batch-Dateien im Installationsverzeichnis zur Verfügung. Der Präprozessor wird automatisch vor dem entsprechenden Simulationsprogramm aufgerufen.
- Graphische Benutzeroberfläche (nur Windows), siehe [Kapitel 6](#).

Hinweis: Wenn Sie Umgebungsvariablen im Kombination mit der GUI nutzen wollen, öffnen Sie eine Konsole, setzen die gewünschten Umgebungsvariablen und starten die GUI.

Beispiel: `set OMP_PROC_BIND=close`

5.1 Upgrade von Modelldaten auf aktuelle Version

Modelle ab der HYDRO_AS-2D Version 3.0 können direkt genutzt werden. Ältere Modelle der Version 2.x (SMS10-Format) müssen vor dem Aufruf der Programme konvertiert werden.

Weiterhin müssen Modelle in die aktuelle Version konvertiert werden, wenn Sie alle neuen Erweiterungen der Einstellungen nutzen möchten.

Ein Upgrade kann nur unter Windows durchgeführt werden.

Für ein Upgrade auf die aktuelle Version nutzen Sie die im HYDRO_AS-Programmverzeichnis bereitgestellte Batch-Datei `upgrade2dm<Versionsnummer>.bat`. Diese Batch-Datei wird in das Verzeichnis der zu konvertierenden 2dm-Datei mit dem Namen `hydro_as-2d.2dm` kopiert und dort ausgeführt. Alle nötigen Konvertierungen werden automatisch ermittelt und durchgeführt. Die ursprüngliche Datei wird unter dem Namen `hydro_as-2d_alt.2dm` gesichert.

Alternativ können Sie die 2dm-Datei mit der graphischen Oberfläche öffnen, die Prüfungen durchführen und anschließend exportieren, vergl. [Kapitel 6.8](#).

Hinweis: Ein „Downgrade“ der 2dm-Datei auf eine ältere Version ist nicht möglich.

5.2 Konvertieren von Ein- und Ausgabedateien

Für die weitere Verarbeitung von Ein- und Ausgabedateien kann es hilfreich sein, diese Dateien in ein anderes Format zu konvertieren. Zwischen dem ASCII-Format (.dat) und dem Binärformat (.h5) der Ein- und Ausgabedateien von HYDRO_AS-2D können Sie beliebig wechseln ohne den Rechenlauf erneut zu starten. Dazu stehen Ihnen im Installationsverzeichnis unter *Hilfsprogramme* zwei Tools zur Verfügung.

Ein- und Ausgabedateien im binären XMDF-Format (.h5) können mithilfe des Tools *h5todat* in das ASCII-Format umgewandelt werden. Analog können Ein- und Ausgabedateien im ASCII-Format (.dat) in Binärdateien (h5) im dem Tool *dattoh5* umgewandelt werden. Dazu werden unter Windows zwei Batch-Dateien - *h5todat_<Versionsnummer>.bat* und *dattoh5_<Versionsnummer>.bat* - im Installationsverzeichnis unter *Hilfsprogramme* zur Verfügung gestellt. Diese Batch-Dateien kopieren Sie in das Verzeichnis des Modells (neben die 2dm-Datei), für das die Ein- oder Ausgaben konvertiert werden sollen, passen ggf. die Aufrufe an und führen die Batch-Datei aus. Unter Linux werden entsprechen die ausführbare Dateien *Hilfsprogramme/h5todat* bzw. *Hilfsprogramme/dattoh5* per Kommandozeile aufgerufen.

Achtung:

Unter Linux muss der LD_LIBRARY_PATH erweitert werden. Dazu rufen Sie folgende Anweisung im Installationsverzeichnis auf:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH: <Installationsverzeichnis/Versionsnummer>/lib
```

Für die Konvertierung muss der Name der zu konvertierenden Datei ohne Dateiendung angegeben werden. Zusätzlich gibt es weitere Optionen.

Parameter	Beschreibung	erforderlich/ optional	Beispiel
-fname <Dateiname>	Angabe des Namens der zu konvertierenden Datei ohne Dateiendung.	erforderlich	-fnamedepth
-fpath<Pfad>	Gibt den Pfad zur zu konvertierenden Datei an. Default ist ./Data-out/AS/. Muss nicht angepasst werden, wenn das Skript aus dem 2dm-Verzeichnis aus aufgerufen wird und das Verzeichnis dem Default ./Data-out/AS/ entspricht.	optional	- fpath./Test1/AS/
-last	Nur der letzte Zeitschritt ist in der konvertierten Datei enthalten.	optional	-last
-nr<i1,i2,...>	Liefert in der konvertierten Datei die angegebenen Zeitschritte i1,i2,... i1,i2,... müssen ganze Zahlen sein, maximal dreistellig	optional	-nr2,10, 35,203
-n<i>	Liefert in der konvertierten Datei jeden i-ten Zeitschritt. i muss eine ganze Zahl sein, maximal dreistellig	optional	-n10
-longformat	Nur für h5todat: Schreibt die Werte in der ASCII-Datei im freien Format. Einsatz bei Simulationszeiten länger als 99999 Sekunden für die Ausgabedateien <i>max_time</i> und <i>max_time_veloc</i> empfohlen.	optional	-longformat

Beispiel: Sie möchten von der Datei *wspl.h5* im Ausgabeverzeichnis *Data-out/AS/* nur den letzten Zeitschritt konvertieren, weil es sich um einen stationären Rechenlauf handelt. Dazu rufen Sie folgenden

Befehl - unter Linux per Kommandozeile und unter Windows per Kommandozeile oder im angepassten Batch-Skript - aus dem Verzeichnis der 2dm-Datei auf:

```
<HYDRO_AS-2D Installationsverzeichnis/Versionsnummer>/Hilfsprogramme/h5todat
-fnamewspl -last
```

Beispiel: Sie möchten vor der Simulation die Anfangswassertiefe als Binärdatei erstellen. Eine passende Datei *wtiefe_0.dat* im ASCII-Format liegt bereits im Ordner /Data-in/AS/. Dazu rufen Sie folgenden Befehl - unter Linux per Kommandozeile und unter Windows per Kommandozeile oder im angepassten Batch-Skript - aus dem Verzeichnis der 2dm-Datei auf:

```
<HYDRO_AS-2D Installationsverzeichnis/Versionsnummer>/Hilfsprogramme/dattoh5
-fpath./Data-in/AS/ -fnamewtiefe_0
```

5.3 Multiprozessorumgebung

Da von den Simulationsprogrammen HYDRO_AS und HYDRO_AS-1STEP in der CPU-Version OpenMP zur Parallelisierung eingesetzt wird, muss die sogenannte Umgebungsvariable OMP_NUM_THREADS gesetzt werden. Dies ist in den Batch-Dateien berücksichtigt. Standardmäßig werden 2 threads eingesetzt. Durch Editieren des Ausdrucks `set OMP_NUM_THREADS=2` kann eine beliebige Anzahl gewählt werden. Die Anzahl ist allerdings durch das Doppelte der vorhandenen Lizenzen beschränkt, da pro Lizenz zwei threads eingesetzt werden können. Wir empfehlen die Anzahl der threads insgesamt (für gleichzeitig laufende Simulationen) höchstens auf die Anzahl der Cores des Rechners zu setzen.

Die GPU-Version berechnet einen Großteil der Simulation auf der Grafikkarte (GPU).

Falls der Computer über mehrere GPU verfügt, kann eine bestimmte GPU durch den Befehl `set ACC_DEVICE_NUM=[Nummer der Karte]` in der entsprechenden Batch-Datei gewählt werden.

5.4 Auswahl von Lizenzen

Falls Lizenzen für verschiedene Knotenanzahlen gleichzeitig zur Verfügung stehen, wird von HYDRO_AS-2D zuerst die höchste Lizenz verwendet. Dieser Mechanismus kann deaktiviert werden, um gezielt eine bestimmte Lizenz zu nutzen.

Dazu ergänzen Sie hinter dem Aufruf des Simulationsprogramms den Kommandozeilenparameter `-l<num>`, wobei num die Werte

- 1 für die Vollversion
- 2 für 100.000 Knoten
- 3 für 10.000 Knoten
- 4 nicht belegt
- 5 für GPU (wird automatisch beim Starten der GPU-Version benutzt)

annehmen kann.

Beispiel:

Sie haben eine Lizenz für die Vollversion und eine Lizenz für 10.000 Knoten. Sie möchten ein kleines Modell mit der 10.000-Knoten Lizenz rechnen. Dazu starten Sie den Rechenlauf mit dem Aufruf

```
<Installationsverzeichnis>\hydro_as.exe -l3
```

bzw. ergänzen `-l3` hinter dem Aufruf der *hydro_as.exe* (oder *hydro_as-1step.exe*) in der batch-Datei.

5.5 Eingabedateien für den Präprozessor HYDRO_2DM

Der Präprozessor liest und verarbeitet eine 2dm-Datei in einem gültigen Format (Kopplung an HYDRO_AS-2D durch die 2dm-Vorlagedatei notwendig). Diese Hauptdatei muss den Namen *hydro_as-2d.2dm* haben (Ausnahme bei Nutzung der graphischen Benutzeroberfläche, vergl. [Kapitel 6](#)).

5.6 Ausgabedateien des Präprozessors/Eingabedateien für die Simulationsprogramme

Die Ausgabedateien des Präprozessors werden von den Simulationsprogrammen eingelesen. Folgende Dateien müssen daher für den Start der Simulationsprogramme im korrekten Format vorhanden sein.

- *hydro_as-2d.pod*: Angaben über das Berechnungsnetz (Geometrie, Elemente, Kanten, Kontrollvolumina, Rauheitsfläche), Geländehöhe, Rauheitsbeiwerte, Bauwerksunterkanten
- *hydro_as-2d.inp*: Modellsteuerung (Simulationszeit, Zeitpunkte zum Ausschreiben der Ergebnisdaten), Randbedingungen, Kontrollquerschnitte – und Punkte, Bauwerksangaben

Bei aktivierten Feststofftransport-Modulen (FT, GS1, GSM, ST) zusätzlich:

- *hydro_ft-2d.inp*: Modelldaten für den Feststofftransport

Bei aktiviertem Wärmetransport-Modulen (WT) zusätzlich:

- *hydro_wt-2d.inp*: Modelldaten für den Wärmetransport

Weitere Ausgaben des Präprozessors dienen zur Kontrolle:

- *hydro_2dm.mel*: Fehlermeldungen und Warnungen.
- *strickler.dat*: Enthält die Rauheitswerte k_{St} nach Strickler bezogen auf die Knoten des Berechnungsnetzes.
Dient vor allem zur Information sowie zur Kontrolle (Visualisierung als Konturplot) oder zur Ermittlung sekundärer Ergebnisse (z.B. Ermittlung von Schleppspannung mit dem SMS Data Calculator).
- *hydro_as-2d_kuk.dat*: Enthält die Höhenwerte für die abflusswirksamen Brückenunterkanten bezogen auf die Knoten des Berechnungsnetzes.
Dient vor allem zur Information sowie zur Kontrolle (Visualisierung als Konturplot).
- *volumen.dat*: Die Fläche des Kontrollvolumens (AKV, siehe [Kapitel 3.1](#)). Dient vor allem zur Information sowie zur Kontrolle im Zusammenhang mit dem A_{min} -Wert.

Optional werden durch die Simulationsprogramme weitere Dateien eingelesen.

Eingangsdateien des Programms HYDRO_AS / HYDRO_AS-1STEP:

- *hydro_as-2d.pod*
- *hydro_as-2d.inp*

sowie optional:

- *wtiefe_0.(dat/h5)*, siehe [Kapitel 4.7](#)
- *geschw_0.(dat/h5)*, siehe [Kapitel 4.8](#)

- nodeniederschlag.(dat/h5), siehe [Kapitel 4.6.3](#)
- nodesource.dat, siehe [Kapitel 4.6.5](#)
- sources-in.(dat/h5), siehe [Kapitel 4.6.6](#)
- feflow-in.(dat/h5), siehe [Kapitel 4.6.4](#)

sowie zum Fortsetzen eines Rechenlaufs:

- hydro_as-2d.cpr
- hydro_as-2d-timesteps.cpr
- ggf. hydro_dauer.cpr
- ggf. hydro_hmax.cpr
- ggf. hydro_schubmax.cpr
- ggf. hydro_tmax.cpr

5.7 Ausgabedateien der Simulationsprogramme

Die Berechnungsergebnisse aus HYDRO_AS / HYDRO_AS-1STEP werden in mehreren Dateien abgespeichert. Dabei gibt es einerseits Dateien, die nur in SMS eingelesen werden können, und andererseits Dateien, die auch zur weiteren Bearbeitung außerhalb von SMS geeignet sind.

Die SMS-Ausgabedateien können sowohl im ASCII-Format als auch im XMDF-Format (binär) erstellt werden.

Die gewünschten Ausgabeformate können im Modul HYDRO-AS EIN-/AUSGABEDATEIEN (E/A: Dateiname.dat/.h5) in SMS gewählt werden. Die Standardeinstellung ist das binäre Format. Die restlichen Ausgabedateien werden nur im ASCII-Format erstellt.

Die Namen der Dateien können in SMS vom Benutzer festgelegt werden. Dazu muss im Dialog Global Parameters das Modul HYDRO-AS EIN-/AUSGABEDATEIEN (E/A: Dateiname. dat/.h5) aktiviert werden. In der Spalte *Value* sind unter den Ausgabeformaten die Default-Namen der Dateien angegeben, die ersetzt werden können. In der folgenden Abbildung wurde beispielhaft der Default-Name „depth“ durch „Wassertiefe“ ersetzt.

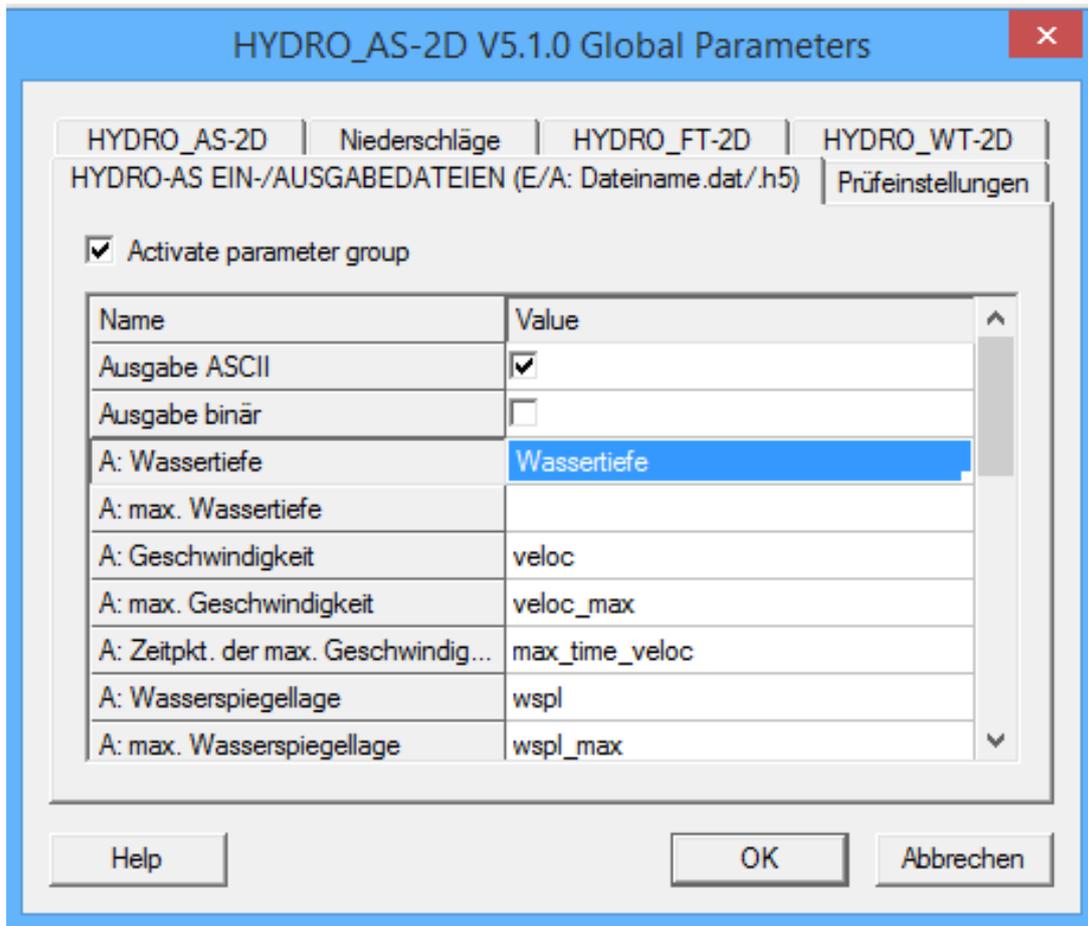


Abbildung 5.1: Benutzerdefinierte Dateinamen

Die Ausgabe von einzelnen Dateien kann deaktiviert werden. Dazu wird in SMS der Dateiname im Modul HYDRO-AS EIN-/AUSGABEDATEIEN gelöscht (leerer String). Im Beispiel ist die Ausgaben der Dateien für die maximale Wassertiefe deaktiviert.

In der folgenden Erläuterung der Dateien werden die Default-Namen verwendet.

Tabelle 5.2: ASCII-Dateien (.dat) /X MDF-Dateien (.h5) zum Einlesen in SMS

Default-Name	Einheit	Beschreibung	Aktivierung (Default)
depth	m	Berechnete Wassertiefe für jeden Knoten des Berechnungsnetzes für die SMS-Ausgabe-Zeitschritte	✓
depth_max	m	Maximal erreichte Wassertiefe für jeden Knoten des Berechnungsnetzes während der Simulationsdauer	
veloc	m/s	Berechnete Fließgeschwindigkeit für jeden Knoten des Berechnungsnetzes für die SMS-Ausgabe-Zeitschritte	✓
veloc_max	m/s	Maximal erreichte Fließgeschwindigkeit für jeden Knoten des Berechnungsnetzes während der Simulationsdauer	
wspl	abs. Höhe in m	Berechneter Wasserstand für jeden Knoten des Berechnungsnetzes für die SMS-Ausgabe-Zeitschritte	✓

Default-Name	Einheit	Beschreibung	Aktivierung (Default)
wspl_max	abs. Höhe in m	Maximal erreichter Wasserstand für jeden Knoten des Berechnungsnetzes während der Simulationsdauer	✓
max_time	s	Zeitpunkt des maximalen Wasserstands für jeden Knoten des Berechnungsnetzes während der Simulationsdauer. Kann nur im Zusammenhang mit wspl_max und/oder depth_max bestimmt werden.	
max_time_velocity	s	Zeitpunkt der maximalen Geschwindigkeit für jeden Knoten des Berechnungsnetzes während der Simulationsdauer. Kann nur im Zusammenhang mit veloc_max bestimmt werden.	
dauer	h	akkumulierte Überflutungsdauer für jeden Knoten des Berechnungsnetzes für die SMS-Ausgabe-Zeitschritte	
schub_max	N/m ²	Maximal erreichte Schubspannung für jeden Knoten des Berechnungsnetzes während der Simulationsdauer	
fe-flow-out	abs. Höhe in m	Wasserspiegel (ohne Fortsetzung an den trocken/nass Grenzen) für jeden Knoten des Berechnungsnetzes zum Endzeitpunkt der Simulation	
sources-out	m ³ /s	Quellterme für jeden Knoten des Berechnungsnetzes für die SMS-Ausgabe-Zeitschritte	
kst_gesamt	m ^(1/3) /s	Rauheit für jeden Knoten des Berechnungsnetzes, bei konstanten Rauheitswerten nur zum Endzeitpunkt der Simulation, bei tiefenabhängigen Rauheitswerten für die SMS-Ausgabe-Zeitschritte	✓
timestep	s	Minimale Zeitschrittweite für jeden Knoten des Berechnungsnetzes für die SMS-Ausgabe-Zeitschritte	

ASCII-Dateien zum Einlesen in EXCEL- oder in ein vergleichbares Programm

- q_strg.dat: Abflüsse in m³/s an den Nodestings (außer 1D-Bauwerke) – tabellarisch und zeitabhängig aufbereitet. Die Anzahl der Nachkommastellen kann in SMS umgestellt werden (siehe Abbildung).

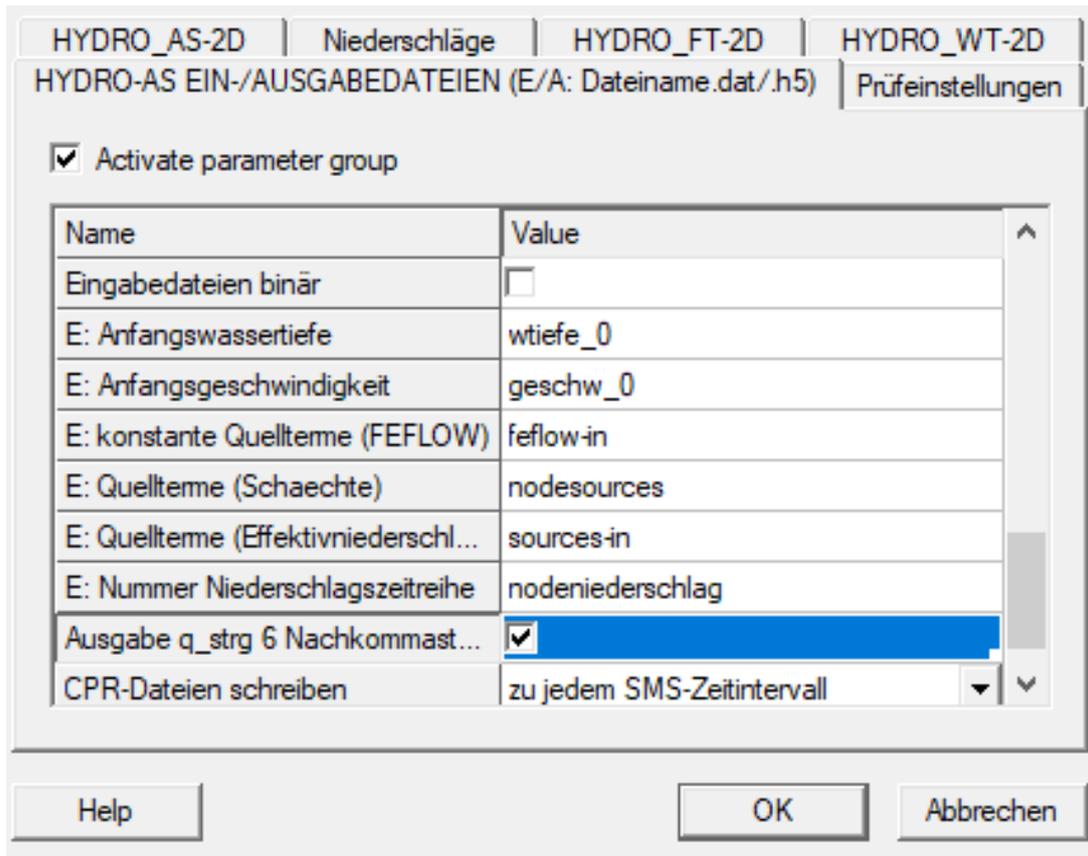


Abbildung 5.2: Einstellung von 6 Nachkommastellen in der Ausgabedatei q_strg.dat

- pegel.dat: Ermittelte Wasserstände als absolute Höhen in m an den Kontrollstellen im Modellgebiet – tabellarisch und zeitabhängig aufbereitet.
- wehr.dat: Ermittelte und vorgegebene Wasserstände als absolute Höhen in m an steuerbaren Wehren und den zugehörigen Oberwasserpegelpunkten sowie der ggf. intern ermittelte Abminderungsfaktor – tabellarisch und zeitabhängig aufbereitet.
- schacht.dat: Abflüsse der Schächte (Knotenrandbedingungen Zufluss/Abfluss) in m^3/s – tabellarisch und zeitabhängig aufbereitet.
- volumenbilanz.dat: Wasservolumen in m^3 im Zeitverlauf nach Strukturen (Zufluss-Nodestrings, Auslauf-Nodestrings, Restvolumen im Modell, etc.) aufgeschlüsselt.

Ermittelte Abflüsse für die 1D-Bauwerke im ASCII-Format

- bw_tmp.dat: Berechnete Abflüsse für die 1D-Bauwerke für verschiedene Zeitzustände. Dient vor allem der Ergebniskontrolle.

Dateien zum Fortsetzen eines Rechenlaufs

Diese Dateien sind -bis auf hydro_as-2d-timesteps.cpr- unformatierte Binärdateien.

- hydro_as-2d.cpr: Enthält ausschließlich Berechnungsdaten und -ergebnisse zum Fortsetzen eines Rechenlaufs.
- hydro_as-2d-timesteps.cpr: Angabe der erfolgten Ausgabezeitschritte im ASCII-Format. Beim Fortsetzen eines Rechenlaufs werden die vorhandenen Zeitschritte dieser Datei entnommen.

- hydro_dauer.cpr: Enthält Berechnungsdaten und -ergebnisse der Einstaudauer zum Fortsetzen eines Rechenlaufs. Ist die Datei beim Fortsetzen nicht vorhanden, wird die Dauer ab dem Fortsetzungszeitpunkt bestimmt.
- hydro_hmax.cpr: Enthält Berechnungsdaten und -ergebnisse der maximalen Wassertiefe und Wasserspiegellage zum Fortsetzen eines Rechenlaufs. Ist die Datei beim Fortsetzen nicht vorhanden, werden die maximale Wassertiefe und die maximale Wasserspiegellage ab dem Fortsetzungszeitpunkt bestimmt.
- hydro_schubmax.cpr: Enthält Berechnungsdaten und -ergebnisse der maximalen Sohlschubspannung zum Fortsetzen eines Rechenlaufs. Ist die Datei beim Fortsetzen nicht vorhanden, wird die maximale Sohlschubspannung ab dem Fortsetzungszeitpunkt bestimmt.
- hydro_tmax.cpr: Enthält Berechnungsdaten und -ergebnisse des Zeitpunkts des maximalen Wasserstands zum Fortsetzen eines Rechenlaufs. Ist die Datei beim Fortsetzen nicht vorhanden, wird der Zeitpunkt ab dem Fortsetzungszeitpunkt bestimmt.

Für das Schreiben der cpr-Dateien gibt es drei Möglichkeiten; Schreiben zu jedem SMS-Zeitintervall, das Schreiben komplett zu deaktivieren oder immer nur die Werte des letzten gültigen Zeitschritts in die Dateien zu schreiben. Dazu wählen Sie die entsprechende Option in SMS im Dialog Global Parameters → Modul HYDRO-AS EIN-/AUSGABEDATEIEN (E/A: Dateiname.dat/. h5) für den Parameter CPR-Dateien schreiben.

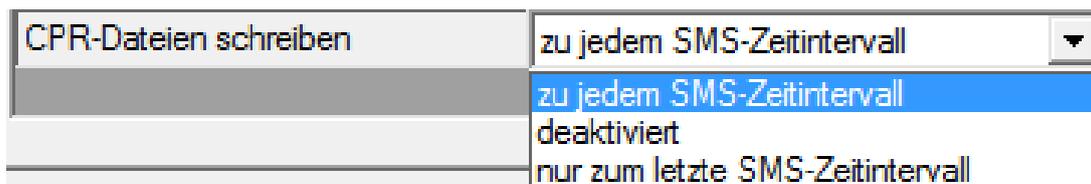


Abbildung 5.3: Optionen für das Schreiben der cpr-Dateien in SMS

Beachten Sie, dass die gewählte Option Auswirkungen auf das Fortsetzen eines Rechenlaufs haben kann.

Dateien zur Benutzung des FEFLOW-interface-managers

- feflow-out.dat: Die Datei enthält den Wasserspiegel (ohne Extrapolation an den trocken/nass Grenzen) zum aktuellen Zeitpunkt. Die Datei wird für die Nutzung von FEFLOW benötigt. Die Werte werden daher fortlaufend mit den neu berechneten überschrieben.
- feflow-in.dat: Die Datei enthält in FEFLOW berechnete Quellterme, die in HYDRO_AS-2D eingelesen werden. Sie kann auch entkoppelt von FEFLOW zur Eingabe von konstanten Quelltermen benutzt werden, siehe [Kapitel 4.6.4](#).

5.7.1 Ausgabe der Wasserspiegellagen

Die berechneten Wasserstände werden um eine Elementreihe nach außen verlängert, damit die Verschneidung mit der Geländeoberfläche erfolgreich durchgeführt werden kann. So ist an den Randbereichen der Wasserspiegel niedriger als die Geländehöhe. In komplett trockenen Elementen werden die WSP-Werte auf einen festen Wert gesetzt, der hinreichend kleiner ist als die niedrigste Sohlhöhe. In den Dateien wspl.dat und wspl_max.dat wird dieser Wert in der ersten Zeile durch den Ausdruck „NaN=Wert“ angegeben (NaN = Not a Number).

5.7.2 Berechnung von Maximalwerten und Summenwerten

Für die Berechnung der Maxima in den Dateien *wspl_max.(dat/h5)*, *depth_max.(dat/h5)*, *veloc_max.(dat/h5)* und *schub_max.(dat/h5)* und für die Bestimmung der Einstaudauer in der Datei *dauer.(dat/h5)* werden nicht die SMS-Ausgabe-Zeitschritte, sondern die Q_Strg-Zeitschritte verwendet. Für eine möglichst genaue Maxima- bzw. Summen-Bestimmung sollte der Q_Strg-Zeitschritt daher nicht zu groß gewählt werden. Des Weiteren wird der maximale Wasserspiegel aus der maximalen Wassertiefe ermittelt und anschließend die Wasserspiegellagen-Extrapolation durchgeführt. Die Maximalwerte der *wspl.(dat/h5)* stimmen also nicht notwendigerweise mit den Werten der *wspl_max.(dat/h5)* überein.

5.7.3 Ausgabe der Volumenbilanz

In der Datei *volumenbilanz.dat* werden Wasservolumen für jeden Q_Strg-Zeitschritt ausgegeben. Positive Werte bedeuten Zugaben, negative Werte Entnahmen. Die Volumeneinheiten werden nach Struktureinheiten gruppiert und jeweils für das gesamte Modell bestimmt. Folgende Struktureinheiten werden betrachtet:

- Zulauf: gibt die Wassermenge an, die insgesamt bis zu diesem Zeitpunkt dem Modell durch alle Zufluss-Nodestrings zusammen zugegeben wurde. Der Wert steigt oder stagniert also im zeitlichen Verlauf (unter der Voraussetzung, dass keine negativen Zuflusswerte eingetragen sind).
- Auslauf: gibt die Wassermenge an, die insgesamt bis zu diesem Zeitpunkt dem Modell durch alle Auslauf-Nodestrings zusammen entnommen wurde. Die Entnahme ist durch ein negatives Vorzeichen gekennzeichnet. Der Wert sinkt oder stagniert also im zeitlichen Verlauf.
- Restvolumen: gibt das zu diesem Zeitpunkt im Modell verbliebene Wasservolumen an. Diese Menge stellt die Summe der Wassertiefen im gesamten Modell dar.
- Schacht: gibt die Wassermenge an, die insgesamt bis zu diesem Zeitpunkt dem Modell durch alle Knoten-Randbedingungen vom Typ *Zufluss/Abfluss* zusammen entnommen oder zugegeben wurde. Da die Knoten-Randbedingungen vom Typ *Zufluss/Abfluss* (Schacht) sowohl als Quellterm also auch als Senkterm wirken können, hängt die Entwicklung des Wertes im zeitlichen Verlauf vom Modell ab. Die berechneten Volumen-Werte können sowohl positive also auch negative Vorzeichen aufweisen.
- source-in: gibt die Wassermenge an, die insgesamt bis zu diesem Zeitpunkt dem Modell durch alle Knoten über die in der Datei *source-in* definierten Werte zusammen entnommen oder zugegeben wurde. Da in der Datei *source-in* sowohl Quellterme als auch Senkterme angegeben werden können, hängt die Entwicklung des Wertes im zeitlichen Verlauf vom Modell ab. Die berechneten Volumen-Werte können sowohl positive also auch negative Vorzeichen aufweisen.
- feflow-in: gibt die Wassermenge an, die insgesamt bis zu diesem Zeitpunkt dem Modell durch alle Knoten über die in der Datei *feflow-in* definierten Werte zusammen entnommen oder zugegeben wurde. Da in der Datei *feflow-in* sowohl Quellterme als auch Senkterme angegeben werden können, hängt die Entwicklung des Wertes im zeitlichen Verlauf vom Modell ab. Die berechneten Volumen-Werte können sowohl positive also auch negative Vorzeichen aufweisen.
- NiederschlagZR: gibt die Wassermenge an, die insgesamt bis zu diesem Zeitpunkt dem Modell durch alle Niederschlags-Zeitreihen zusammen zugegeben wurde. Rein technisch können in den Niederschlags-Zeitreihen auch negative Werte eingegeben werden. In diesem Fall hängt die Entwicklung des Wertes im zeitlichen Verlauf vom Modell ab. In den meisten Anwendungsfällen werden allerdings nur positive Niederschlagswerte angegeben. Dann steigt oder stagniert der Wert also im zeitlichen Verlauf.

Beachten Sie: Die hinzugefügten und entnommenen Volumen werden **bis zum jeweiligen Ausgabezeitpunkt** bestimmt. Der einzige Wert, der **zum Ausgabezeitpunkt** gilt, ist das Restvolumen.

Die Zeilensummen dieser Ausgabe resultiert (bis auf Rundungsfehler durch die Summationen) im Wert Null oder dem Volumen, das durch eine vorgegebene Anfangswassertiefe zu Simulationsstart vorhanden war.

Hinweis: Das Restvolumen im ersten Ausgabeschritt entspricht ungefähr dem dem Volumen, das durch eine vorgegebene Anfangswassertiefe zu Simulationsstart vorhanden war.

In den Kopfzeilen finden Sie über der Angabe der Struktureinheiten die Anzahl der entsprechenden Strukturelemente, d.h. die Anzahl der Zuflussrandbeingung, Auslaufrandbedingungen, Modellkonten usw.

5.7.4 Ausgabedateien im Binärformat

Die Ausgabedateien im Binärformat, die von HYDRO_AS-2D geschrieben werden, haben das folgende Format:

- Skalare Größen wie beispielsweise die Wassertiefe (depth) sind jeweils in separaten Dateien enthalten.
- Vektorwertige Größen wie die Fließgeschwindigkeit (veloc) enthalten die Komponenten in x- und y-Richtung als Vektor-Dateityp, sowie die Magnitude (Länge des Vektors) als separaten skalaren Datensatz.
- Die Datensätze in den Dateien sind nach dem Schema <Dateiname>_<Modellname> benannt.
- Die Datensätze sind direkt auf der ersten Ebene in die h5-Datei eingetragen. Es wird vorab KEINE Gruppe angelegt.
- Die Ausgabedateien der Feststofftransportmodule umfassen in der Regel mehrere Datensätze. Dabei sind unterschiedliche Berechnungsgrößen für alle Kornfraktionen enthalten.

Hinweis: Nutzen Sie HDFView, um die Struktur der Dateien einzusehen.

5.8 Fortsetzung eines Rechenlaufs

In SMS gibt es drei Optionen einen Rechenlauf zu starten zur Auswahl. Die letzten beiden Optionen beziehen sich dabei auf das Fortsetzen des Rechenlaufs.



Abbildung 5.4: Auswahl *Rechenlauf fortsetzen?* in SMS

Ist die Option 1 – *fortsetzen mit Auswahl des Zeitschritts* gewählt, ist nach erneuter Ausführung des Präprozessors und dem Ausführen des Simulationsprogramms HYDRO_AS/HYDRO_AS-1STEP anzugeben, ab welchem Zeitpunkt die Berechnung fortgesetzt werden soll.

```

*****
HYDRO_AS-2D 4.1.0      1-Step
*****

Die Datei HYDRO_AS-2D.CPR wird eingelesen...

NUMREAD_CPR = 1  ZEIT = 10.00000
NUMREAD_CPR = 2  ZEIT = 300.0107

Anzahl der gespeicherten Schritte = 2

Rechenlauf fortsetzen beim Schritt Nr...  <bitte die gewünschte Nr. eingeben>

```

Abbildung 5.5: Fortsetzen eines Rechenlaufs

Will man die manuelle Eingabe vermeiden, kann das Simulationsprogramm HYDRO_AS/HYDRO_AS-1STEP mit dem Kommandozeilenparameter `-c<Zeitschrittnummer>` aufgerufen werden.

Ist die Option 2 – *beim letzten Zeitschritt fortsetzen* gewählt, rechnet das Simulationsprogramm nach Ausführen des Präprozessors ab dem letzten gespeicherten Zustand weiter.

Zum Fortsetzen eines Rechenlaufs müssen die Datei `hydro_as-2d.cpr` und `hydro_as-2dtimesteps.cpr` im aktuellen Format vorhanden sein. Die weiteren `cpr`-Dateien (vergl. [Kapitel 5.7](#)) können auch vorab gelöscht werden. Dann werden die entsprechenden Ausgaben ab dem Zeitpunkt des fortgesetzten Rechenlaufs ermittelt.

Änderungen am Netz dürfen vor dem Fortsetzen eines Rechenlaufs nicht vorgenommen werden, da die Daten in der Dateien `hydro_as-2d.cpr` dann in der Regel nicht mehr zum Netz passen und nicht korrekt eingelesen werden können. Ausnahmen stellen die Werte von Zuflussganglinien (aber nicht die Anzahl der Stützstellen!) und globale Parameter wie z.B. das Ende der Simulationszeit dar.

Hinweis: Beim Fortsetzen eines Rechenlaufs werden die aus den Dateien `wtiefe_0.dat` und `geschw_0.dat` eingelesenen Daten durch die entsprechenden Werte aus der Datei `hydro_as-2d.cpr` überschrieben.

Die Ausgabedateien `q_strg.dat`, `pegel.dat`, `bw_temp.dat`, `wehr.dat` und die Dateien zur Stoff- und Wärmetransportraten werden beim Fortsetzen eines Rechenlaufs vorab so gekürzt, dass keine doppelten Zeitschritte enthalten sind.

Die SMS-Ausgabedateien im binären X MDF-Format (`.h5`) werden bei Option 1 – *fortsetzen mit Auswahl des Zeitschritts* auf den angegebenen Zeitschritt gekürzt, bevor neue Daten geschrieben werden. Dadurch enthalten diese Ausgabedateien KEINE doppelten Zeitschritte. Die SMSAusgabedateien im ASCII-Format (`.dat`) werden NICHT vorab gekürzt. Hier können doppelte Zeitschritte entstehen.

5.9 Kontrolliertes Beenden eines Rechenlaufs und Aktivierung zusätzlicher Ausgaben

Beim Start eines Rechenlaufs wird die Datei `hydro_as-sim.inp` erzeugt. Sie liegt im selben Verzeichnis wie die `hydro_as-2d.2dm`-Datei. Mit der `hydro_as-sim.inp`-Datei kann die Simulation kontrolliert beendet oder es können zusätzliche Ausgaben verschiedener Simulationsergebnisse gesteuert werden. Dazu editieren Sie die `hydro_as-sim.inp`-Datei.

Es gibt folgende Optionen, den Rechenlauf zu beeinflussen:

- STOP: Die Simulation wird sofort mit Ende des internen Zeitschritts beendet
- STOP_SMS: Die Simulation wird nach dem nächsten Rausschreiben der SMS-Ergebnisse beendet
- DEBUG: Der aktuelle Zeitschritt und die Zeitschrittweite werden für alle internen Zeitschritte ausgegeben

- SMS: SMS-Ergebnisse werden einmalig zusätzlich rausgeschrieben
- RESULT: Es werden einmalig zusätzliche Ergebnisse in die Dateien q_strg.dat, pegel.dat und bw_tmp.dat geschrieben

Beim Start des Rechenlaufs sind alle Optionen ausgeschaltet. Das entspricht einer „0“ am jeweiligen Zeilenanfang. Die ersten drei Optionen werden aktiviert, indem man die „0“ durch eine „1“ ersetzt und die Datei abspeichert. Die Option DEBUG kann wieder rückgängig gemacht werden, indem die „1“ wieder durch eine „0“ ersetzt wird. Die Optionen SMS und RESULT können mehrmals während eines Rechenlaufs aufgerufen werden. Dazu muss die Zahl am Zeilenanfang erhöht werden.

Beispiel:

Man möchte zu mehreren verschiedenen Zeitpunkten zusätzliche SMS-Ergebnisse rausschreiben. Dazu ersetzt man zuerst die „0“ am Anfang der Zeile „0 SMS - SMS Ergebnisse rausschreiben“ durch eine „1“. Die Ergebnisse werden dann einmal rausgeschrieben. Ist der zweite Zeitpunkt erreicht, ersetzt man die „1“ durch eine „2“, usw.

Die angegebenen Optionen zum Abbrechen des Rechenlaufs oder zur Ausgabe von weiteren Zeitschritten können auch direkt über die graphische Oberfläche angesteuert werden, siehe [Kapitel 6.7](#).

5.10 Optimieren von Rechenzeiten

Die Rechenzeiten werden von verschiedenen Faktoren beeinflusst.

5.10.1 1step vs. 2step

Das 2step-Verfahren (HYDRO_AS, vergl. [Kapitel 3.3](#)) ist genauer, aber daher auch langsamer als das 1step-Verfahren (HYDRO_AS-1STEP). Falls es die Anforderungen der Aufgabenstellung zulassen, kann das 1step-Verfahren benutzt werden, um kürzere Rechenzeiten zu erzielen.

5.10.2 Analyse der Zeitschrittweiten

Die intern ermittelten Zeitschrittweiten können für jeden SMS-Ausgabezeitschritt und jeden Netzknoten ausgegeben werden.

Dazu geben Sie in SMS unter Global Parameters → HYDRO-AS EIN/AUSGABEDATEIEN (E/A: Dateiname.dat/h5) für A: Zeitschrittweiten einen Dateinamen an. Die so erzeugte Ausgabedatei wird im Weiteren mit *timestep.(dat/h5)* bezeichnet.

Laden Sie die Datei *timestep.(dat/h5)* in SMS ein, um die Netzgeometrie hinsichtlich der Zeitschrittweite zu verbessern.

Die Zeitschrittweite wird maßgeblich durch die Länge der Kanten bestimmt. Daher sind sehr kleine Knotenabstände und Elemente zu vermeiden.

Achten Sie bei der Netzerzeugung außerdem darauf, dass die Zeitschrittgröße nicht durch die Randzellen bedeutend eingeschränkt wird. Die für die Bestimmung der Zeitschrittgröße maßgebende Länge ist für die Randpunkte nur noch halb so groß wie im inneren Bereich, weil das Kontrollvolumen für einen Randpunkt nur aus einem halben bzw. für einen Eckpunkt aus einem Viertel des Randelementes besteht. Setzen Sie deshalb entsprechend große bzw. nach Möglichkeit nicht allzu kleine Elemente am Modellrand ein. Beachten Sie auch, dass ausgestanzte Gebäude und Widerlager Modellränder erzeugen.

5.10.3 Konstruktionsunterkanten

Ein geringer Abstand von Geländehöhe und Höhe einer KUK kann zu hohen Fließgeschwindigkeiten führen. Dadurch wird unter Umständen auch der interne Zeitschritt sehr klein. Dies können Sie in der Ausgabe der Zeitschrittweiten (timestep.dat/h5) nachvollziehen. Achten Sie daher darauf, dass die KUKs plausibel gesetzt sind. Hinweise finden Sie auch in den Ausgaben des Präprozessors.

5.10.4 Wassertiefe

Allzu große Wassertiefen - z.B. in einem See, der nicht Hauptgegenstand der Untersuchung ist - schränken oft unnötig den Zeitschritt ein. Falls man nicht an der Größe von Fließgeschwindigkeiten im See interessiert ist und der See keinen Einfluss auf das Retentionsvolumen hat, kann der See im Modell deutlich flacher abgebildet werden. Dies hat auf die Berechnung des Wellenablaufs und auf die Ermittlung von Wasserspiegelhöhen / Überschwemmungsflächen nahezu keinen Einfluss und führt gleichzeitig u.U. zu bedeutend geringeren Rechenzeiten.

Alternativ können in vielen Fällen die Elemente im See vergrößert werden. Kontrollieren Sie auch die Zeitschritte in der Ausgabedatei timestep.(dat/h5).

5.10.5 Amin

Im Allgemeinen sind für die Abbildung hydraulisch relevanter Strukturen unterschiedliche Knotenabstände erforderlich. Das führt dazu, dass die interne Zeitschrittweite sehr klein werden kann, da in der Berechnung der Zeitschrittweite

$$\Delta t = \min \left(\frac{\Delta L}{|v| + \sqrt{gh}} \right) \quad (20)$$

die charakteristische Länge ΔL linear eingeht. Dabei wird ΔL aus den Flächen der Kontrollvolumen A_{KV} ermittelt. D.h. für kleine A_{KV} wird Δt klein.

Für die Berechnung von ΔL und des Residuums im Finite Volumen Verfahren wird

$$A_{KV}^* = \max(A_{KV}, A_{min}) \quad (21)$$

eingesetzt, so dass eine geeignete Wahl von A_{min} die Rechenzeit beschleunigen kann.

Bei der Wahl von A_{min} ist darauf zu achten, dass die gesamte Strömung dadurch nicht zu stark verändert wird. Ein zu großer Wert für A_{min} kann das Erreichen des stationären Zustands sogar verlangsamen. Weiterhin wird die Leistungsfähigkeit geometrisch abgebildeter Strukturen verändert.

Hinweis: Bei instationären Berechnungen können Volumendifferenzen und eine Veränderung des Wellenablaufs auftreten. Daher sollte A_{min} nur für stationäre und quasi-stationäre Berechnungen größer als null gewählt werden.

Die Default-Einstellung ist $A_{min} = 0$, d.h. alle Kontrollvolumen bleiben für die Berechnung unverändert. Diese Einstellung beeinflusst daher weder den Wellenablauf noch das Retentionsvolumen im Modell. Allerdings bedeutet dies auch die längsten Rechenzeiten.

Da ein A_{min} -Wert ungleich null die Simulationsergebnisse verändern kann, muss A_{min} je nach Modell und Fragestellung passend gewählt werden.

Dabei kann die Datei `volumen.dat`, die vom Präprozessor erzeugt wird, hilfreich sein. In dieser Datei finden Sie für jeden Knoten die Fläche des unveränderten Kontrollvolumens A_{KV} .

Der Präprozessor liefert außerdem einen Hinweis dazu, wie viele Knoten ein kleineres Kontrollvolumen als A_{min} haben und welche Flächenänderung sich daraus ergibt.

5.10.6 Schreiben der Ergebnisse

Achten Sie darauf, dass die Ergebnisse nicht allzu oft ausgeschrieben werden, da dies die Rechenzeit erhöht.

6 Graphische Benutzeroberfläche

Die Benutzeroberfläche zu HYDRO_AS-2D ermöglicht es, Prüfungen auf den Modelldaten durchzuführen und im Anschluss eine Simulation zu starten.



Abbildung 6.1: Startbildschirm der Benutzeroberfläche

Wenn die Benutzeroberfläche ohne die Angabe einer Modelldatei (*.2dm) gestartet wird, öffnet sich der Startbildschirm. Hier hat der Nutzer nun die Möglichkeit eine von drei Startoptionen auszuwählen, mit der dann eine Modelldatei behandelt wird, vergl. [Kapitel 6.1.2](#).

Die Benutzeroberfläche umfasst die vollständige Funktionalität des Präprozessors. Es können alle 2dm-Dateien ab Version HYDRO_AS-2D 3.0 direkt eingelesen werden. Eine externe Konvertierung der 2dm-Datei ist nur für ältere HYDRO_AS-2D 2.x Versionen erforderlich, da diese das ältere 2dm-Format von SMS 10 verwenden.

In der Benutzeroberfläche kann – im Gegensatz zur aktuellen Kommandozeilenversion von HYDRO_AS-2D – eine 2dm-Datei mit beliebigem Namen eingelesen werden. Die Datei muss nicht zwingend hydro_as-2d.2dm heißen. Beachten Sie, dass die Simulation immer im gleichen Verzeichnis durchgeführt wird, in

dem die 2dm-Datei liegt. Wir empfehlen deshalb immer nur eine 2dm-Datei pro Arbeitsverzeichnis vorzuhalten, um zu verhindern, dass während der Simulation Konflikte zu Ergebnisdateien einer anderen Modelldatei entstehen.

6.1 Allgemeines zur Benutzeroberfläche

6.1.1 Gruppierung von Bedienelementen

An mehreren Stellen in der graphischen Oberfläche werden Bedienelemente zu Gruppen zusammengefasst. Diese Gruppen sind mit einem Pfeilsymbol links neben der Überschrift gekennzeichnet. Per Mausklick auf die Überschrift oder das Pfeilsymbol werden die Elemente der Gruppe angezeigt oder ausgeblendet.

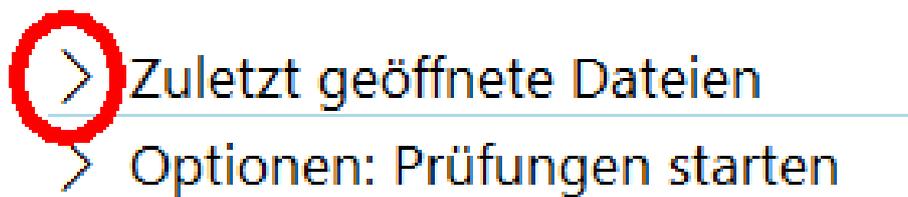


Abbildung 6.2: Gruppierung von Bedienelementen

6.1.2 Optionen

Um festzulegen, welche Schritte nach dem Öffnen einer 2dm-Datei ausgeführt werden sollen, öffnen Sie auf der Startseite der Benutzeroberfläche die Bediengruppe Optionen und wählen Sie einen der folgenden drei Punkte:

- Nur Öffnen – Die Datei wird nur geöffnet und in den Speicher eingelesen.
- Prüfungen starten – Nach dem Einlesen der Datei wird automatisch der Prüfvorgang gestartet. (Dies ist die Standard-Option.)
- Prüfungen und Simulation starten – Nach dem Einlesen wird der Prüfvorgang automatisch gestartet und anschließend die Simulation gestartet, falls keine Fehler bei der Prüfung gemeldet wurden.

6.2 Öffnen einer 2dm-Datei

Nach der Installation von HYDRO_AS-2D können Sie aus dem Windows Explorer über einen Rechtsklick auf eine 2dm-Datei und dem Menüpunkt Öffnen mit direkt eine Datei mit der Benutzeroberfläche öffnen oder die Benutzeroberfläche über das Startmenü öffnen. Wenn die Benutzeroberfläche über das Startmenü geöffnet wird, haben Sie drei Möglichkeiten eine 2dm-Datei zu öffnen:

1. Klicken Sie auf die Schaltfläche 2dm-Datei öffnen und wählen Sie die gewünschte Datei im Auswahlfenster aus.
2. Klicken Sie im Windows Explorer auf eine 2dm-Datei und ziehen Sie sie mit gehaltener Maustaste in den Startbildschirm der HYDRO_AS-2D Benutzeroberfläche („Drag and drop“).
3. Öffnen Sie die Bediengruppe Zuletzt geöffnete Dateien, wählen Sie eine der Dateien aus der Liste aus und klicken Sie auf die Schaltfläche Öffnen, die unter dem Dateinamen angezeigt wird.

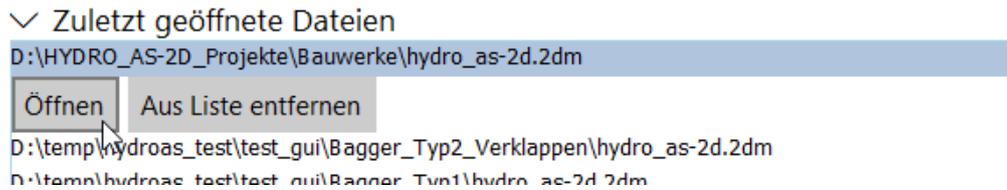


Abbildung 6.3: Liste zuletzt geöffneter Dateien

6.3 Bedienelemente der Benutzeroberfläche

Nach dem Öffnen einer 2dm-Datei wechselt die Ansicht der Benutzeroberfläche. Am linken Rand werden nun Schaltflächen angezeigt, mit denen auf eine von vier Seiten umgeschaltet werden kann. Die aktuelle ausgewählte Seite wird mit einem blauen Rechteck am linken Rand markiert.

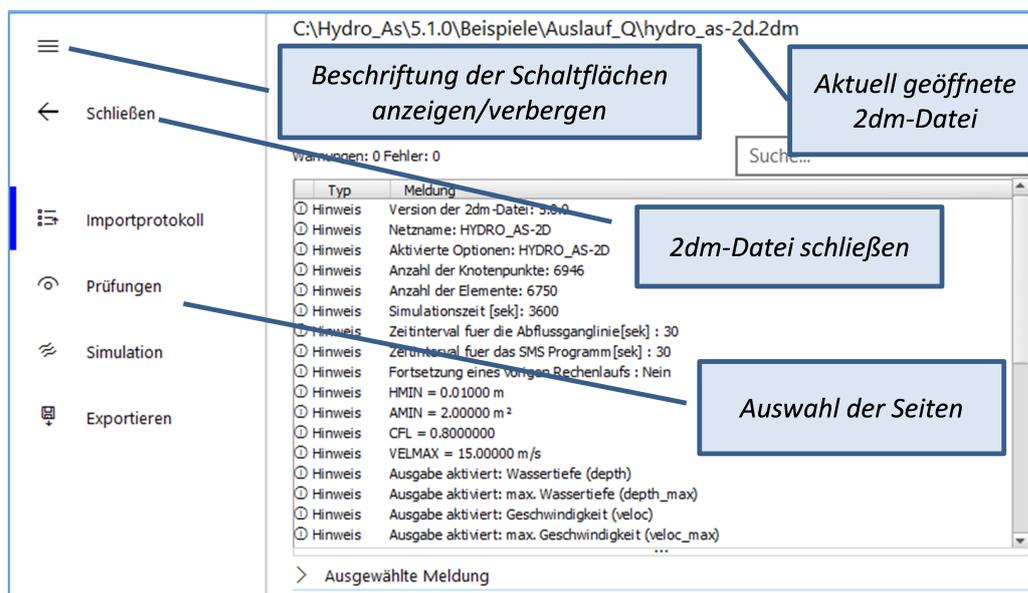


Abbildung 6.4: Allgemeine Bedienelemente

Der Name der aktuell geöffneten Datei wird auf allen Seiten am oberen Rand dargestellt.

6.4 Importprotokoll

Auf der Seite Importprotokoll werden die Meldungen des Präprozessors mit Informationen zum Modell und den gesetzten Parametern angezeigt.

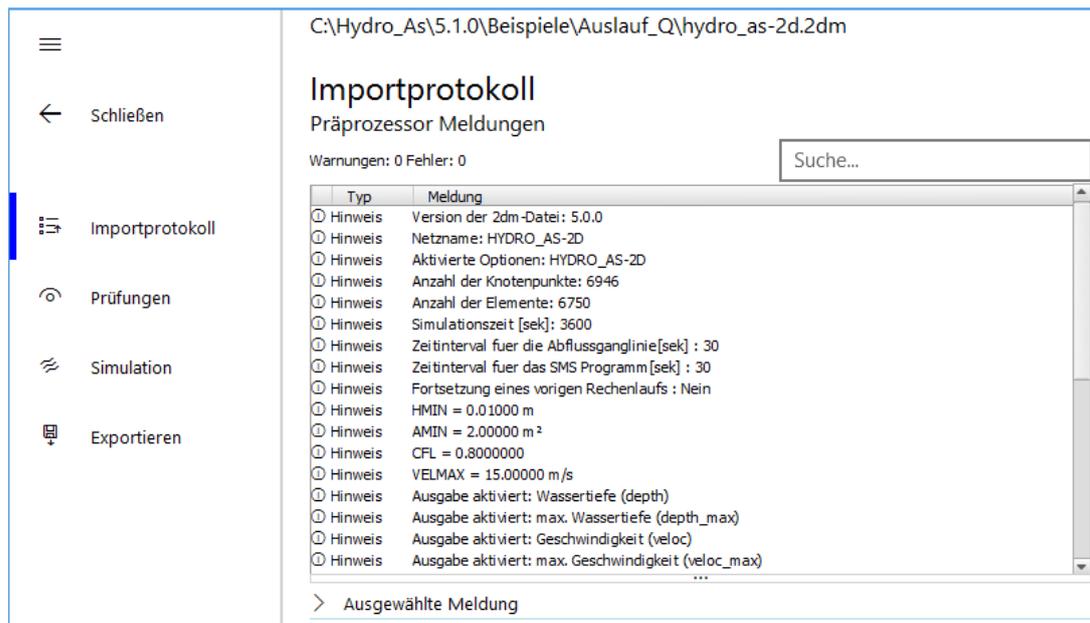


Abbildung 6.5: Seite Importprotokoll

Die Meldungen lassen sich über das Eingabefeld am rechten oberen Rand („Suche. . .“) filtern. Nach Eingabe eines beliebigen Suchtexts werden nur noch die Zeilen angezeigt, die diesen Text enthalten.

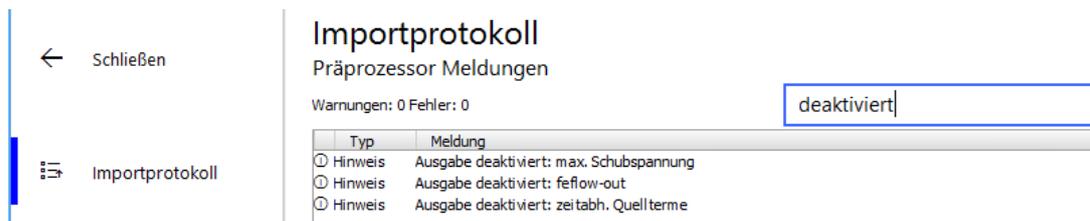


Abbildung 6.6: Filtern von Meldungen

Durch einen Rechtsklick in die Liste der Meldungen öffnet sich ein Menü, über das man die Meldungen in die Zwischenablage kopieren oder in eine Textdatei speichern kann.

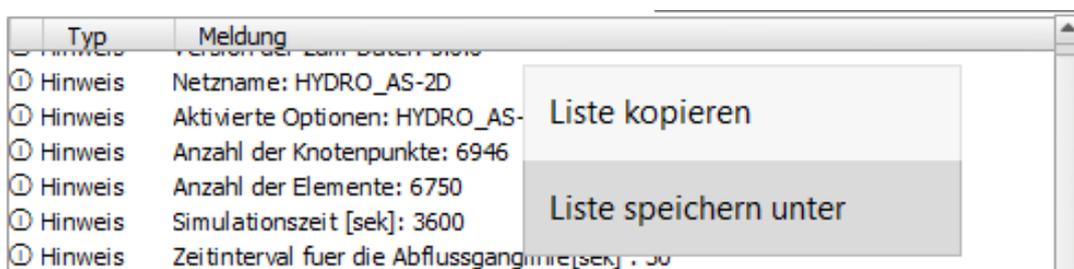


Abbildung 6.7: Kopieren und Speichern der Meldungsliste

Ist beim Kopieren oder Speichern ein Suchfilter aktiv, wird auch nur die gefilterte Liste behandelt.

Falls beim Öffnen der 2dm-Datei Fehler aufgetreten sind, werden diese in Rot im Meldungsfenster aufgelistet. Im Fehlerfall sind alle weiteren Seiten deaktiviert. Die Datei muss dann geschlossen und nach Korrektur der Fehler erneut geöffnet werden.

6.5 Prüfungen

Die Seite Prüfungen listet die Prüfmeldungen des Präprozessors auf. Die Prüfungen beziehen sich sowohl auf Semantik, als auch auf Plausibilität der Modelldaten.

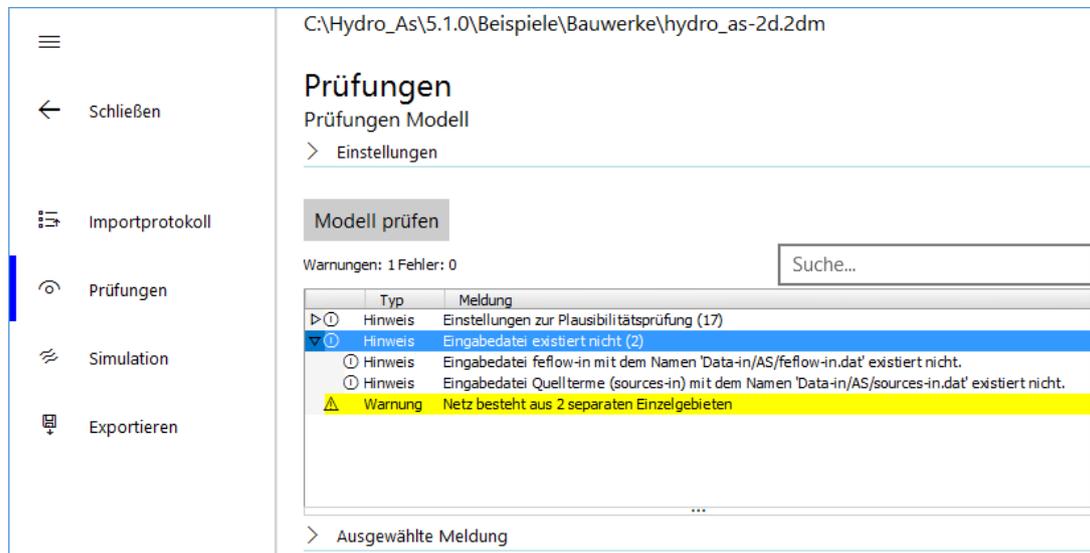


Abbildung 6.8: Seite Prüfungen

Die Meldungen lassen sich analog zu den Meldungen im Importprotokoll filtern, kopieren und speichern. Wenn mehrere Meldungen des gleichen Typs auftreten, werden sie unter einer neuen Überschrift gruppiert. Am Rand erscheint ein Pfeilsymbol, mit dem man die Liste der einzelnen Nachrichten ein- und ausblenden kann. In der Überschrift wird außerdem die Anzahl der einzelnen Nachrichten dieses Typs in Klammern angezeigt.

Die Bediengruppe Einstellungen ermöglicht es, verschiedene Parameter für die Plausibilitätsprüfung anzupassen. Klicken Sie auf die Überschrift „> Einstellungen“ und/oder „> Erweiterte Einstellungen“ um die Bediengruppen einzublenden und geben Sie in die Eingabefelder die gewünschten Grenzwerte für die Plausibilitätsprüfung ein.

Prüfungen
Prüfungen Modell

← Schließen

☰ Importprotokoll

🔍 **Prüfungen**

🔄 Simulation

📄 Exportieren

∨ **Einstellungen**

Minimaler Innenwinkel (in Grad)	Maximaler Innenwinkel (in Grad)
10	120
Maximale Anzahl Nachbarknoten	Minimaler Knotenabstand (in m)
10	0,11 (aus 2dm)
Krit. Flächenverhältnis Nachbarelemente	Krit. Kantenlängenverhältnis für Rechteckselemente
0,05 (aus 2dm)	4

∨ **Erweiterte Einstellungen**

Max. Bauwerksbreite (in m)	Max. Bauwerkshöhe (in m)
150	10
Min. Gefälle für Auslauf le	Max. Gefälle für Auslauf le
0,000010 (aus 2dm)	0,1 (aus 2dm)
Max. Abstand Pegel zu Wehr (in m)	Max. Abstand Bauwerkssohle zu Sohlhöhe (in m)
25	1,00 (aus 2dm)
Minimale Wehrhöhe (in m)	
0,20 (aus 2dm)	
Min. kst-Wert (in $m^{1/3}/s$)	Max. kst-Wert (in $m^{1/3}/s$)
10	200
Min. Wirbelviskosität (in m^2/s)	Max. Wirbelviskosität (in m^2/s)
0,00 (aus 2dm)	2,00 (aus 2dm)

Modell prüfen

Abbildung 6.9: Einstellungen zu den Prüfungen des Modells

Nachdem Sie die Grenzwerte angepasst haben, klicken Sie auf die Schaltfläche **Modell prüfen**, um die Prüfungen mit diesen Werten erneut durchzuführen.

Beim Export des Modells werden die Werte in die 2dm-Datei geschrieben. In SMS finden Sie die Einstellungen unter „Global Parameters > Prüfeinstellungen“.

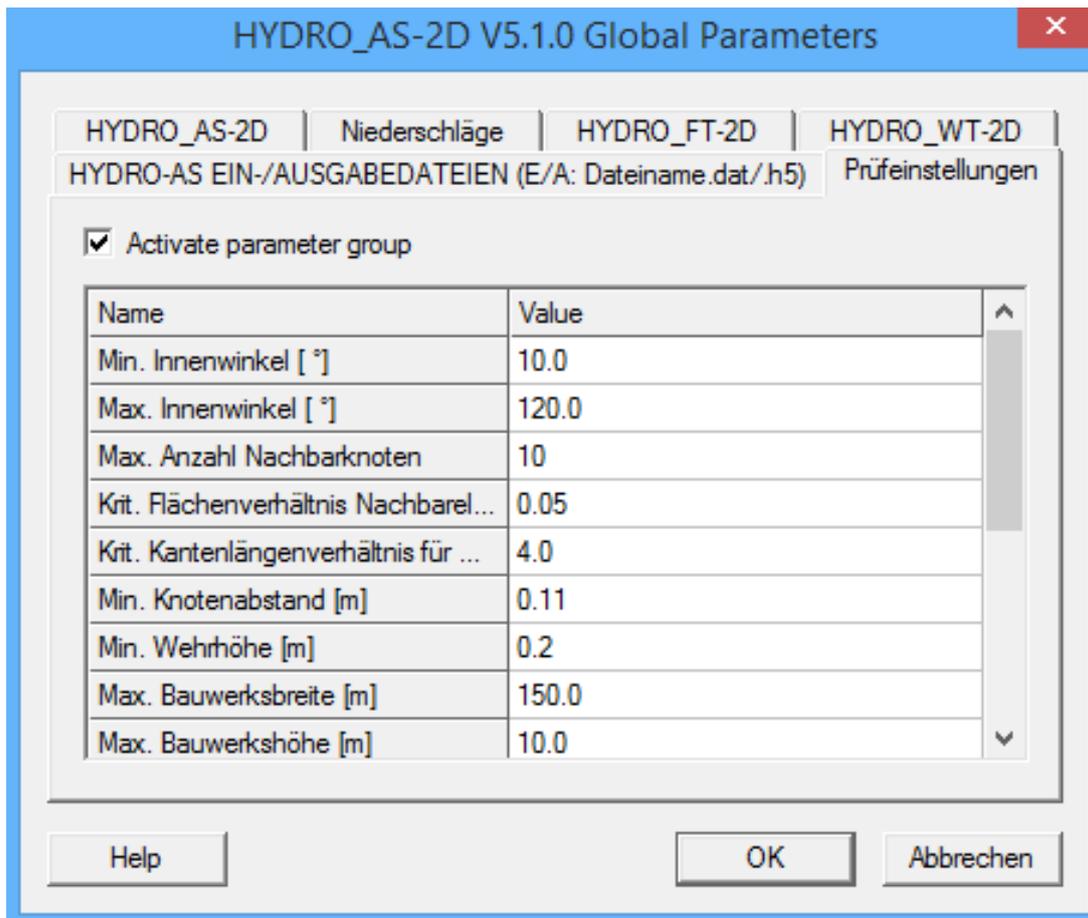


Abbildung 6.10: Reiter Prüfeinstellungen in SMS

Wenn bei den Prüfungen Warnungen und/oder Fehler aufgetreten sind, werden im Unterverzeichnis Preproc Shapefiles (hydro_as-warn.shp bzw. hydro_as-err.shp) erzeugt. Diese Shapefiles enthalten alle Meldungen, die einen Ortsbezug haben. Das sind z.B. Meldungen bei Verletzung der Grenzwerte zum Innenwinkel. Am unteren Rand der Seite Prüfungen werden die aktuellen Pfade zu diesen Shapefiles angezeigt. Klicken Sie auf die darunterliegende Schaltfläche, um den Pfad in die Zwischenablage zu kopieren und ihn in SMS oder eine GIS Oberfläche einzufügen. Die Shapefiles werden nur angelegt, wenn auch Meldungen mit Ortsbezug aufgetreten sind. Falls es keine Meldungen mit Ortsbezug gibt, werden keine Shapefiles erzeugt und die Bediengruppe für den Pfad wird nicht angezeigt.

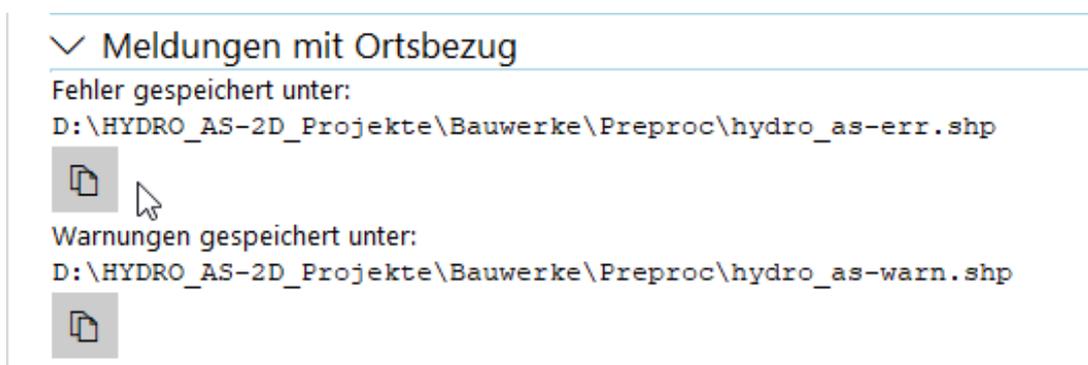


Abbildung 6.11: Meldungen mit Ortsbezug

Falls bei den Prüfungen Fehler aufgetreten sind, müssen diese vor der Simulation korrigiert werden.

Die Seite Simulation ist im Fehlerfall deaktiviert.

6.5.1 Anzeigen der Prüfergebnisse in SMS

Um die Warnungen und Fehlern mit Ortsbezug in SMS anzuzeigen, kopieren Sie zunächst den Pfad zum jeweiligen Shapefile in die Zwischenablage. Wechseln Sie zu SMS und wählen Sie im Menü den Punkt File/Open. Fügen Sie den kopierten Pfad mit der Tastenkombination Strg-V in das Eingabefeld zum Dateinamen und klicken Sie auf Öffnen. Das Shapefile wird nun eingelesen und SMS wechselt zum GIS Module.

Hier können Sie mit einem Rechtsklick auf den Shapefile-Eintrag und den Menüpunkt Open Attribute Table die Liste aller enthaltenen Meldungen einsehen.

Mit dem Werkzeug Get Attributes im GIS Module können Sie nun durch einen Mausklick auf einen der angezeigten Punkte die Informationen zu der jeweiligen Warnung oder Fehlermeldung in einem Info-Fenster aufrufen.

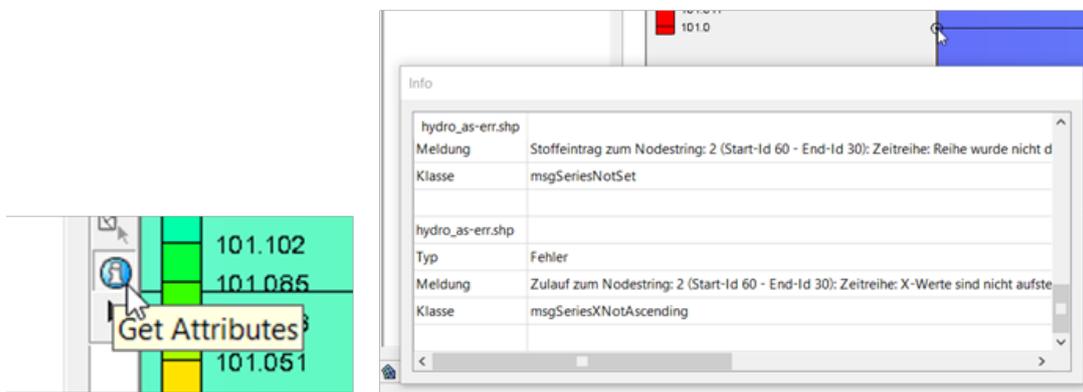


Abbildung 6.12: SMS Werkzeug „Get Attributes“

Zu Beschriftung der Punkte in der Karte muss das Shapefile zuvor in ein Observation Coverage kopiert werden. Gehen Sie dazu wie folgt vor:

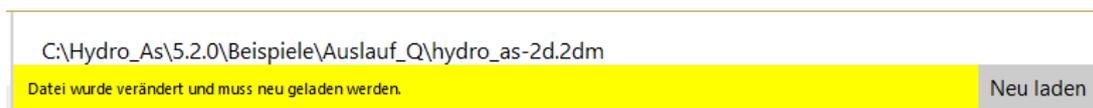
1. Klicken Sie mit der rechten Maustaste auf Map Data und wählen Sie den Menüpunkt Add Coverage
2. Im Auswahldialog New Coverage wählen Sie Generic/Observation und klicken Sie auf OK.
3. Klicken Sie mit der rechten Maustaste auf das geladene Shapefile und wählen Sie aus dem Menü „Convert/Shapes → Feature Objects“.
4. Im folgenden Assistenten wählen Sie das in Schritt 2 erzeugte Coverage und klicken Sie auf „Next“.
5. Wählen Sie nun bei einer der Attributspalten, die Sie als Beschriftung der Punkte darstellen möchten, das Mapping „Name“ wie in untenstehender Abbildung und klicken Sie auf Finish.



Abbildung 6.13: Zuweisen einer Attributspalte als Name eines Observation Coverages

6.6 Neu laden der 2dm-Datei

Wird das Modell während es in der GUI geöffnet ist, (z.B. in SMS) verändert, erscheinen unter dem Dateipfad ein Hinweis, dass sich die Datei verändert hat, und ein Neu-laden-Button, unabhängig von der Seite der GUI.



6.7 Simulation durchführen

Auf der Seite Simulation kann nach erfolgreicher Prüfung der Modelldaten die HYDRO_AS-2D Simulation gestartet werden.

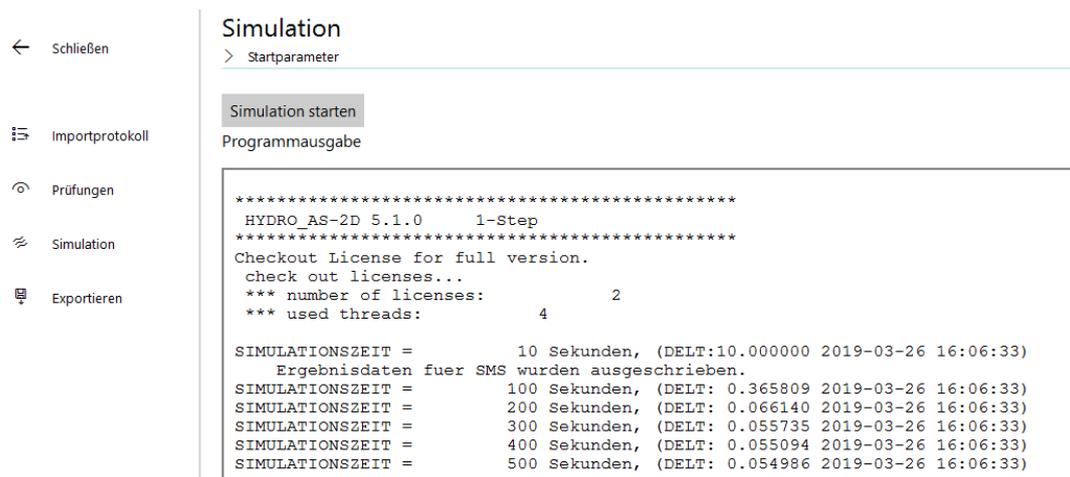


Abbildung 6.14: Seite Simulation

In der Bediengruppe Startparameter können der Rechenkern ausgewählt und einige Parametern angepasst werden.

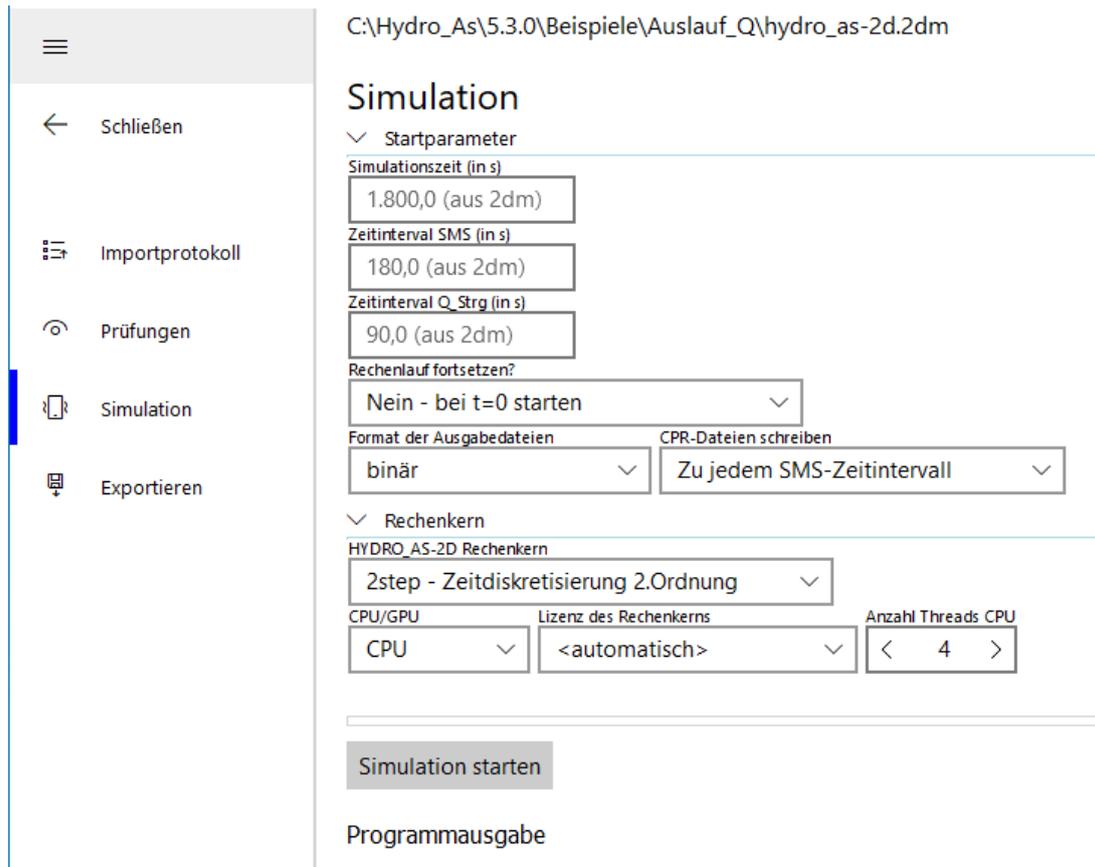


Abbildung 6.15: Startparameter der Simulation

Die drei ersten Parameter zur Simulationszeit und den Zeitintervallen SMS und Q_Strg erhalten als Standard die Werte aus der 2dm-Datei und können hier überschrieben werden. Falls das Verzeichnis Dateien einer vorhergehenden Simulation enthält, können Sie über das Auswahlfeld *Rechenlauf fortsetzen?* die Simulation an einem ausgewählten oder dem letzten Zeitschritt wiederaufnehmen. Bei der Option *Ja – Mit Auswahl des Zeitschritts* erscheint ein weiteres Auswahlfeld, in dem der Zeitschritt zur Wiederaufnahme selektiert werden kann. Es folgen zwei Felder zur Auswahl des Formats der Ausgabedateien (ASCII, binär oder beides) und für die Optionen zum Schreiben der cpr-Dateien. Für beide Felder werden die Einstellungen zunächst aus der 2dm-Datei übernommen und können in den Auswahlfelder noch geändert werden.

Unter HYDRO_AS-2D Rechenkern wählen Sie, ob die Simulation mit Zeitdiskretisierung 1. oder 2. Ordnung durchgeführt werden soll. Eine weitere Option bestimmt, ob die Simulation auf dem Hauptprozessor (CPU) oder einer Grafikkarte (GPU), mit welcher Lizenz und mit wie vielen Threads des Hauptprozessors durchgeführt werden soll. Bei Auswahl von GPU kann gleichzeitig noch die Nummer des Grafikkarten-Adapters ausgewählt werden.

Nachdem alle Einstellungen vorgenommen sind, kann der Rechenkern über die Schaltfläche *Simulation starten* aufgerufen werden. Alle Ausgaben des Rechenkerns lassen sich im darunter liegenden Textfeld verfolgen. Die Ausgaben werden zusätzlich – wie bei der Kommandozeilenversion – in die Textdatei *hydro_as-2d.mel* herausgeschrieben.

Wenn die Simulation gestartet wurde, werden weitere Schaltflächen für den Abbruch der Simulation und zusätzliche Ausgaben angezeigt.

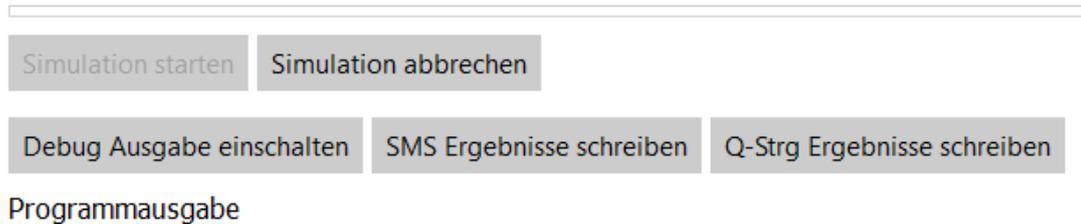


Abbildung 6.16: Weitere Schaltflächen nach dem Simulationsstart

Klickt man auf die Schaltfläche *Simulation abbrechen*, erscheint zunächst eine Rückfrage zur Bestätigung, ob der Nutzer tatsächlich die Simulation abbrechen möchte.

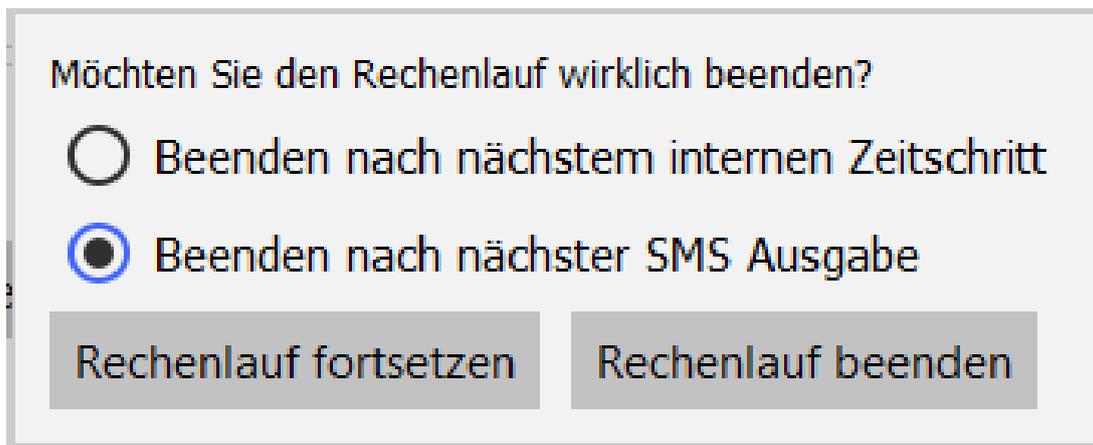


Abbildung 6.17: Rückfrage Simulation abbrechen

Hier können Sie wählen, ob der Rechenlauf nach dem nächsten internen Zeitschritt (sofort) oder erst nach dem nächsten Rausschreiben der Ergebnisse (SMS-Zeitschritt) abgebrochen werden soll. Wenn diese Rückfrage mit *Rechenlauf beenden* bestätigt wird, erhält die Simulation das Signal, nach dem nächsten Berechnungszyklus bzw. Schreiben der Ergebnisse anzuhalten. Dabei werden alle Ergebnisse bis zu dem Zeitpunkt des Abbruchs herausgeschrieben und der Nutzer kann die Simulation mit Auswahl eines Zeitschritts zu einem späteren Zeitpunkt fortsetzen.

Wenn Sie die Debug-Ausgabe starten, werden während der Berechnung weitere Informationen zu den internen Zeitschritten ausgegeben. Nach dem Klicken auf die Schaltfläche ändert sich die Beschriftung in *Debug Ausgabe ausschalten*. Mit einem weiteren Klick auf diese Schaltfläche können Sie nun die Debug-Ausgabe wieder beenden.

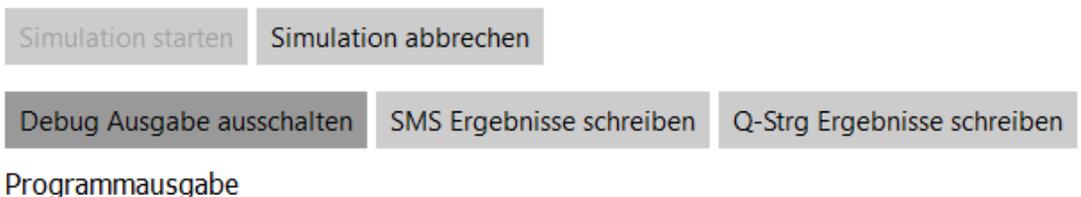


Abbildung 6.18: Debug-Ausgabe ausschalten

Über die Schaltflächen *SMS Ergebnisse schreiben* und *Q-Strg Ergebnisse schreiben* können Sie zusätzliche Ergebnisse außerhalb der SMS- und Q_Strg-Zeitschritte anfordern. Die entsprechenden Ergebnisse werden dann separat in Ergebnisdateien mit entsprechendem Zeitpunkt im Dateinamen geschrieben (vergl. [Kapitel 5.9](#)).

6.8 Exportieren und Konvertieren in aktuelle Programmversion

Auf der Seite *Exportieren* kann die aktuell geladene 2dm-Datei in eine neue Datei geschrieben werden.



Abbildung 6.19: Seite Exportieren

Klicken Sie auf die Schaltfläche *Speichern unter*, um ein Verzeichnis und einen Dateinamen für die neue Datei zu wählen. Beim Export wird die Datei automatisch in die aktuelle Version von HYDRO_AS-2D konvertiert.

Beachten Sie, dass in der Oberfläche weiterhin die ursprünglich geladene Datei verbleibt. Ein Export in die aktuelle Version ist nur nötig, wenn Sie neue Funktionen der aktuellen HYDRO_AS-2D Version nutzen möchten. Dazu exportieren Sie die Datei, wie beschrieben, und öffnen Sie zunächst in SMS, um die Parameter der gewünschten neuen Funktionen einzustellen. Anschließend muss die neue 2dm-Datei dann in der Benutzeroberfläche erneut geladen werden.

6.9 MapView-Datensatz erzeugen

Im Folgenden wird beschrieben, wie Sie einen Datensatz zur Visualisierung der Ergebnisse in MapView erzeugen. Der erzeugte Datensatz ist eine Datei mit Endung **.hmv**.

Für das Erstellen der hmv-Datei benötigen Sie eine separate Lizenz.

Informationen zu MapView und zur Ergebnisdarstellung in MapView finden Sie in [Kapitel 7.3](#).

Hinweis: Das Programm MapView gibt es nur für Windows.

6.9.1 MapView Benutzeroberfläche

Unter dem Reiter *Exportieren* haben Sie die Möglichkeit, aus Modellergebnissen einen Datensatz für die Visualisierung mit HydroAS MapView zu erstellen.

Für die Erzeugung eines MapView-Datensatzes werden die in der .2dm-Datei definierten Ausgabe-dateien der Simulationsprogramme (vergl. [Kapitel 5.7](#)) genutzt. Es ist mindestens eine **wspl**- oder **wspl_max**-Datei erforderlich.

Tabelle 6.1: Alle von MapView ausgewerteten Ausgabedateien

Datei	Erforderlich	Benutzung
wspl	mindestens wspl oder wspl_max	Darstellung des Zeitverlaufs
wspl_max	mindestens wspl oder wspl_max	Darstellung des Maximal-Zustands
veloc	Nein	Partikelanimation im Zeitverlauf
veloc_max	Nein	Partikelanimation im Maximal-Zustand
q_strg.dat	Nein	Darstellung der Querprofile
pegel.dat	Nein	Darstellung der Pegel

MapView-Datensatz erzeugen

Modellname

Zoom Level Auflösung 0,78 m/Pixel

Projektion

Zentrum des Modells (Längen-\Breitengrad):

Geschwindigkeiten visualisieren?

Zeitschritte

Stationärer Zustand (letzter Zeitschritt)

Startzeitschritt Endzeitschritt

Nur jeden i-ten Zeitschritt auswerten

Einfärbung

Pfad: C:\Hydro_As\5.2.0\Hilfsprogramme\ha2web\colormaps\isoArea5_100cm.colormap

vordefinierte Höheneinfärbung

Legende Vorschau (Höhen in cm):

0 - 4,999
5 - 9,999
10 - 24,999
25 - 49,999
50 - 74,99
75 - 100

Regel exportieren?

Programmausgabe

Abbildung 6.20: MapView-Datensatz erzeugen

Im Folgenden werden die Parameter erläutert:

Modellname - Der Modellname erscheint im Titel der MapView-Anwendung.

Zoom Level - Das Zoom Level entspricht dem Zoom Level der Web Mercator Projektion (EPSG:3857). Hier haben Sie die Auswahl des Zoom Levels für die höchste Auflösungsstufe. Daneben steht die Auflösung in Meter pro Pixel auf dieser Stufe. Der Auflösungswert variiert mit der Position des Längengrades.

Projektion - Hier muss die Projektion vorgegeben werden, in der die Koordinaten des Modells vorliegen. Unterhalb der Projektionauswahl haben Sie eine Box, die den Mittelpunkt der Ausdehnung des Modells in Längen- und Breitengrad angibt. Mit einem Klick auf Anzeigen in Karte können Sie sich den Mittelpunkt auf dem ausgewählten Zoom Level in einem Browser anschauen.

Geschwindigkeiten visualisieren - Steuert, ob Kacheln für die Geschwindigkeiten erzeugt werden sollen.

Mit *Export starten* starten Sie die Berechnung des Datenstazes. Neben dem Export starten Button können Sie die Argumente, die Sie durch die Angabe der Felder in der GUI getroffen haben, in die Zwischenablage kopieren und ggf. das Kommandozeilentool mit diesen Argumenten aufrufen.

Hinweis: das Argument `--outdir` ist nicht beim Kopieren enthalten, diese müssen Sie ggf. ergänzen. Im nächsten Kapitel können Sie mehr über das Kommandozeilentool erfahren.

Zeitschritte

Hier können Sie die Zeitschritte einschränken, die Sie in Ihren Datensatz aufnehmen möchten.

Wählen Sie den stationären Zustand aus, beinhaltet der Datensatz nur den allerletzten Zeitschritt.

Unter *Start- und Endzeitschritt* können Sie Zeitpunkte in Sekunden angeben um die Ausgabe einzuschränken. Ist eine `.cpr`-Datei nach der Simulation vorhanden, können Sie Zeitschritte aus der Combobox auswählen.

Unter *Nur jeden i-ten Zeitschritt auswerten* können Sie Zeitschritte überspringen. 1 gibt alle Zeitschritte aus, 2 gibt jeden 2ten aus, 3 jeden dritten, usw..

Einfärbung

Hier können Sie die Einfärbung der Einstautiefen auswählen. Für die Einfärbung gibt es `.colormap`-Dateien im Verzeichnis

```
<HydroAS-Installationsverzeichnis>\Hilfsprogramme\ha2web\colormaps.
```

Auf der Rechten Seite sehen Sie eine Vorschau der Colormap-Definition. Ist neben der Farbe nur ein Wert angegeben, wird der Farbwert zwischen der vorherigen und dem nachfolgenden Farbwert linear interpoliert.

Wenn Sie eine eigene Colormap-Datei nutzen wollen, können Sie diese über den Eigene Einfärbungsdatei auswählen-Button auswählen. Für Tiefen, die sich außerhalb der Definition befinden wird respektive der unterste oder der höchste Farbewert genutzt.

Die Colormap-Dateien folgen dem Schema:

```
<Tiefe in cm> <Rot[0-255]> <Grün[0-255]> <Blau[0-255]> <Alpha[0-255]>
<Tiefe in cm> <Rot[0-255]> <Grün[0-255]> <Blau[0-255]> <Alpha[0-255]>
...
```

Pegelpunkte und Randbedingungen

Pegelpunkte und Randbedingungen aus der `pegel.dat` und `qstrg.dat` können Sie unter dem ausklappbarem Menü Pegelpunkte auswählen oder Querprofile auswählen entfernen oder hinzufügen. In der Oberfläche wird die Nodestring- bzw. Node-Id gefolgt von dem Namen der Randbedingung angezeigt.

6.9.2 MapView Kommandozeilen Tool

Das Kommandozeilentool liegt unter

<HydroAS-Installationsverzeichnis>\Hilfsprogramme\ha2web\ha2web.exe.

Notwendige Parameter

```
-g <Georeferenzierung>, -georef <Georeferenzierung>
```

Die Projektion muss nach einem bestimmten Pattern angegeben werden. In der folgenden Übersicht der Projektionen muss xy durch eine Streifennummer ergänzt werden. Beispielsweise für Für DHDN / 3-degree Gauss-Kruger zone 2 entspricht der Parameter -g DHDNGK2.

Unterstützte Projektionen

DHDN / 3-degree Gauß-Krüger

Pattern: **DHDNGKxy**

Gauss-Krueger-Koordinaten im Deutschen Hauptdreiecksnetz mit Angabe des (3-Grad breiten) Streifens.

Beispiel: DHDNGK2

Projektionen

EPSG:31466 DHDN	/ 3-degree Gauss-Kruger zone 2
EPSG:31467 DHDN	/ 3-degree Gauss-Kruger zone 3
EPSG:31468 DHDN	/ 3-degree Gauss-Kruger zone 4
EPSG:31469 DHDN	/ 3-degree Gauss-Kruger zone 5

ETRS89 / UTM

Pattern: **UTMxy**

UTM-Koordinaten bzgl ETRS89/WGS84 mit Angabe des (6-Grad breiten) Streifens bei denen die Rechtswerte der Koordinaten die Streifennummer nicht als Prefix führen. Die Rechtswerte sind im Allgemeinen 5-stellig.

Beispiel: UTM32

Projektionen

EPSG:25831	ETRS89 / UTM 31N
EPSG:25832	ETRS89 / UTM 32N
EPSG:25833	ETRS89 / UTM 33N

Pattern: **Prefix-UTMxy**

UTM-Koordinaten bzgl. ETRS89/WGS84 mit Angabe des (6-Grad breiten) Streifens bei denen die Rechtswerte der Koordinaten die Streifennummer als Prefix führen. Die Rechtswerte sind i.Allg. 7-stellig.

Beispiel: Prefix-UTM32

Projektionen

EPSG:5651	ETRS89 / UTM 31N (zE-N)
EPSG:4647	ETRS89 / UTM 32N (zE-N)
EPSG:5650	ETRS89 / UTM 33N (zE-N)

MGI / Austria

Pattern: FalseNorthing-MGIAustriaGK-xy

Gauss-Krueger-Koordinaten im Datum Austria (MGI, AT_MGI); West/Ost-Koordinate misst ab dem Zentralmeridian (kann negativ sein); Nord-Koordinate ist Abstand zum Äquator - 5.000km

Beispiel: FalseNorthing-MGIAustriaGK-34

Projektionen

EPSG:31257	MGI / Austria GK M28
EPSG:31258	MGI / Austria GK M31
EPSG:31259	MGI / Austria GK M34

Pattern: Prefix-MGIAustriaGK-xy

Gauss-Krueger-Koordinaten im Datum Austria (MGI, AT_MGI); West/Ost-Koordinate ist 6 stellig und beginnt mit 1,4 oder 7; Nord-Koordinate ist Abstand zum Äquator.

Beispiele: Prefix-MGIAustriaGK-28, Prefix-MGIAustriaGK-31, Prefix-MGIAustriaGK-33

Projektionen

EPSG:31284	MGI / Austria M28
EPSG:31285	MGI / Austria M31
EPSG:31286	MGI / Austria M34

Optionale Parameter

`-f, -ignoreerroronread`

Verhindert, dass bei zu vielen Fehlern in der 2DM-Datei das Lesen abgebrochen wird.

`-lowmemory`

Signifikant weniger Speicherverbrauch, dafür aber längere Berechnungszeit. Wächst stark mit der Anzahl der erzeugten Zoomstufen.

Wird ein Modell über die grafische Beutzeroberfläche berechnet, dann wird diese Argument immer gesetzt.

`-nomeshring`

Meshumring wird nicht erzeugt.

`-excludedisabledmaterials`

Elemente, die auf disabled gesetzt sind, werden vom Polygonumring abgezogen.

`-withholes`

Innere Polygone nicht aus dem Modellumring entfernen.

`-noarchive`

Ergebnis nicht in Tar-Archiv(.hmv) schreiben, sondern in Ausgabeordner.

`-novectorfield`

Tile-Vektorfeld nicht erzeugen

`-nocrosssections`

Keine Querprofile erzeugen

`-c <Kontrollquerschnitt>, -crosssection <Kontrollquerschnitt>(accepted multiple times)`

Liste von Nodestring-Id's für die Profilvereinerungen (Höhenprofile) ausgegeben werden sollen. Wenn dieses Flag fehlt, werden alle HYDRO_AS-2D-Kontrollquerschnitte ausgegeben. Beispiel für Nodestring 12, 200 und 1300 `-c 12 -c 200 -c 1300`

`-nogauges`

Keine Pegel erzeugen

`-p <Pegel>, -gauge <Pegel> (accepted multiple times)`

Liste von Node-Ids für die Pegel ausgegeben werden sollen. Wenn dieses Flag fehlt, werden alle Pegel ausgegeben. Nutzung ist analog zu `crosssection`. Ausgabe für Pegelpunkt 1, 3, 4

`-p 1 -p 3 -p 4`

`-stationary`

Stationären Zustand erzeugen

`-notilesmax`

Keine Maximal-Tiles mit Höhen erzeugen

`-notilestime`

Keine Zeitschritt-Tiles mit Höhen erzeugen

`-mintimestep <Minimaler Zeitschritt>`

Minimaler Zeitschritt für die anzuzeigenden Ergebnisse in Sekunden

Default: 0.0

`-max timestep <Maximaler Zeitschritt>`

Maximaler Zeitschritt für die anzuzeigenden Ergebnisse in Sekunden

Default: 1.79769e+308

`-timestepsteps <Zeitschritt>`

Festlegen welche SMS-Ausgabezeitschritte verwertet werden sollen: 1=Jeder Zeitschritt; 2=Jeder 2. Zeitschritt; 3=Jeder 3. Zeitschritt; 4=...

Default: 1

`-colormap <Colormapdatei>`

Dateiname einer Colormap. Die Datei muss von der folgenden Form sein:

```
<Tiefe in cm> <R[0-255]> <G[0-255]> <B[0-255]> <Alpha[0-255]>
<Tiefe in cm> <R[0-255]> <G[0-255]> <B[0-255]> <Alpha[0-255]>
...
```

dabei sind die Tiefen absteigend und die Werte für die Farben ganze Zahlen zwischen 0 und 255. Sollte eine Tiefe unterhalb der Definition liegen, so wird diese mit der niedrigsten Höhenfarbe eingefärbt. Liegt eine Tiefe über dem höchsten Werte so wird diese auch mit der Farbe des höchsten Wertes eingefärbt.

Beispiel für eine Colormap, die Tiefen unter 5 Metern sehr Dunkel macht

```
0.001 255 255 255 128
5      0 0 255 0
10     0 0 128 0
```

`-lowercolor <Farbe minimaler Wasserstande>`

Farbe bei minimalen Wasserstand

Default: 255 255 255 0

`-uppercolor <Farbe maximaler Wasserstand>`

Wasserstände, die über dem maximale Wasserstand liegen, werden einheitlich und nicht unterscheidbar in dieser Farbe angezeigt.

Default: 0 0 255 255

`-lowerheight <Minimaler Wasserstand>`

Minimaler darzustellender Wasserstand. Geben Sie einen Wert zwischen 0 und dem HYDRO_AS-2D-Parameter 'HMIN' ein

Default: 0.01

`-upperheight <Maximaler Wasserstand>`

Maximaler darzustellender Wasserstand. Wasserstände größer diesem Wert werden nicht mehr farblich unterschieden.

Default: 3.0

`-mintilelevel <Startlevel für Tiles>`

Zoomstufe, ab der mit der Tileerzeugung begonnen wird (niedrigste Auflösung). Zoomstufe wird normalerweise automatisch berechnet. Sollte der berechnete Wert nicht korrekt sein, kann dieser manuell angegeben werden.

`-maxtilelevel <Endlevel für Tiles>`

Zoomstufe, bis zu dieser Tiles erzeugt werden. Alternativ kann man den `resolution` Parameter nutzen. Wird beides angegeben gewinnt `maxtilelevel`.

`-resolution <Maximale Auflösung>`

Auflösung der genauesten zu erzeugenden Karte in Meter/Pixel. Wenn Sie diesen Wert verkleinern wird die Rechenzeit deutlich zunehmen

Default: 1.0

`-outdir <Ausgabeverzeichnis>`

Dateiname für das Ergebnis. Hier kann ein relativer oder absoluter Pfad angegeben werden.

Default: mapview.hmv

`-n <Projektname>, -name <Projektname>`

Name des Projektes. Dieser Wert überschreibt den Namen der 2DM-Datei (Define Model>Model Parameters>Name). Der Text wird im Kopf der Web-Anwendung angezeigt.

`-2 <2DM-Datei>, -2dm <2DM-Datei>`

Dateiname der 2DM-Datei mit den Modelldaten

7 Ergebnisdarstellung

Die Simulationsergebnisse können in verschiedenen Programmen dargestellt werden. Die Programme bieten unterschiedliche Darstellungsmöglichkeiten. Für eine Auswahl an Programmen wird die Ergebnisdarstellung in den folgenden Kapiteln beschrieben.

7.1 Ergebnisdarstellung mit SMS

Die verschiedenen berechneten und gespeicherten Strömungszustände können mittels SMS veranschaulicht dargestellt werden. Dazu bietet SMS, abhängig vom Datentyp, verschiedene Möglichkeiten. An dieser Stelle wird nur ein kurzer Überblick über diese Möglichkeiten gegeben. Detailliertere Informationen hierzu können dem SMS Benutzerhandbuch entnommen werden.

- Skalare Daten: Wassertiefe, Wasserstand, Schubspannung, Überflutungsdauer, Froude Zahlen, Größe der Geschwindigkeit, etc.
 - Darstellungsart: Flächendeckend mittels Konturlinien, Längs- und Querschnitte
- Vektordaten: Fließgeschwindigkeit
 - Darstellungsart: Flächendeckend mittels Strömungspfeilen

Die Skalar- und die Vektordaten können über File Open als sog. Generic file Datentyp importiert werden. Die Darstellungen beziehen sich dabei immer auf einen bestimmten Berechnungszeitpunkt (ab Simulationsbeginn und in Sekunden ausgedrückt). SMS bietet zudem bei zeitabhängigen Ergebnissen die Möglichkeit, an ausgewählten Stellen im Gebiet (sog. Observation Points) Zeitreihen zu bilden und diese in Form von Ganglinien darzustellen.

Alle Darstellungen am Bildschirm können mit der Print Option ausgedruckt werden. Der Bildschirminhalt (nur das Anzeigefenster) kann des Weiteren mit Copy to Clipboard (unter Edit Menü zu finden) als Bitmap in ein anderes Programm direkt (d.h. mit Kopieren und Einfügen) übernommen werden.

Hinweis: Die Ergebnisdaten können in ASCII- bzw. in DXF- sowie in Shape Format exportiert werden.

7.2 Ergebnisdarstellung mit JabPlot

JabPlot ermöglicht Ihnen, mit nur wenigen Mausklicks Querprofil- und Längsschnittplots aus Ihren 2D-Berechnungsergebnissen zu erstellen, sodass Sie schnell und komfortabel qualitativ hochwertige Grafiken erhalten. JabPlot wird getrennt von HYDRO_AS-2D lizenziert.

JabPlot stellt damit eine Ergänzung zu HYDRO_AS-2D dar. Mit dem in JabPlot integrierten 2D-/3DViewer können sich 2D-Modellierer Netze im 2dm-Format und Berechnungsergebnisse direkt anzeigen lassen. Über frei definierbare Schnittlinien, die im Shapefile-Format vorliegen, können Sie aus den Netzdaten beliebige Schnitte berechnen.

Wiederkehrende Arbeitsschritte zur Darstellung von Berechnungsergebnissen lassen sich mit dem Zusatzmodul JabPlot-Skript effizient durchführen.

Zusätzlich liest JabPlot Vermessungsdaten verschiedener Formate (TXT, CSV, GAF, ...).

Jedem Nutzer stehen durch das MS-ACCESS-Datenbankformat viele Wege zur weiteren Verwendung der Daten offen, wie z. B. die Nutzung in Geo-Informationssystemen (GIS) oder CAD-Programmen.

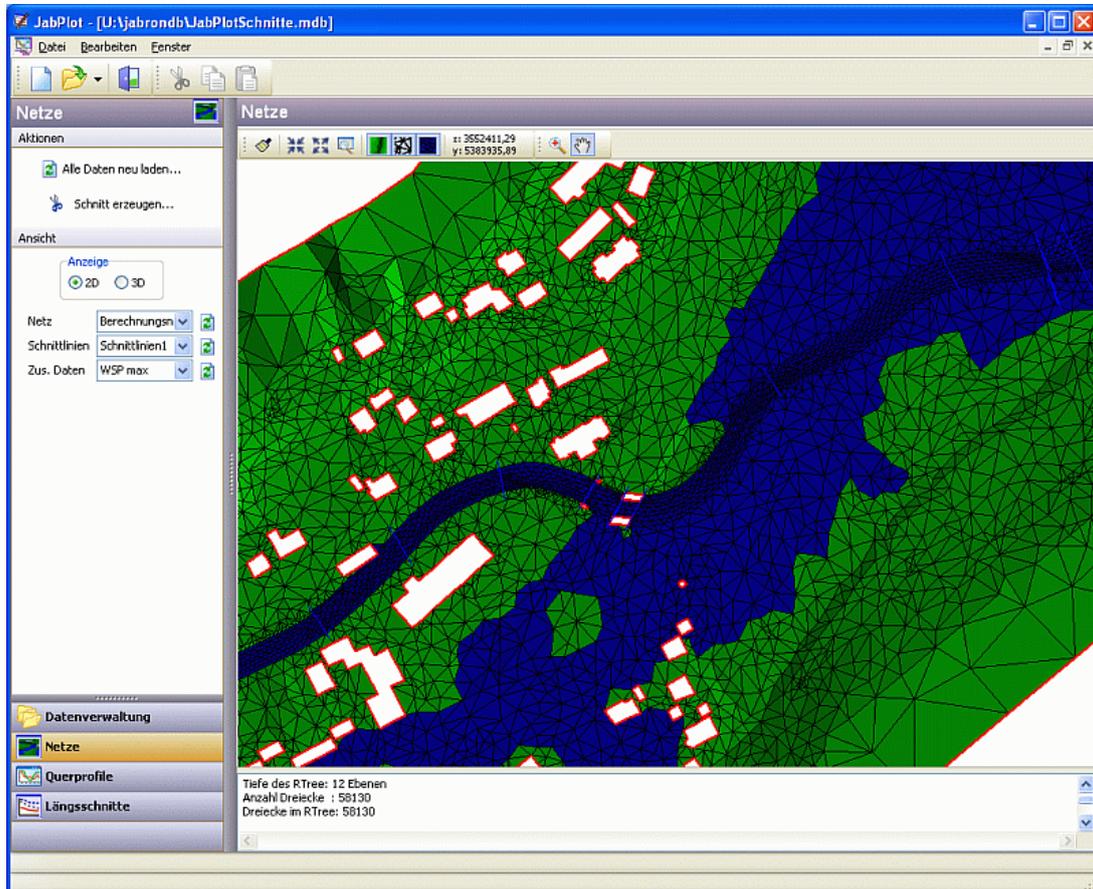


Abbildung 7.3: Modellansicht mit Wassertiefe in JabPlot

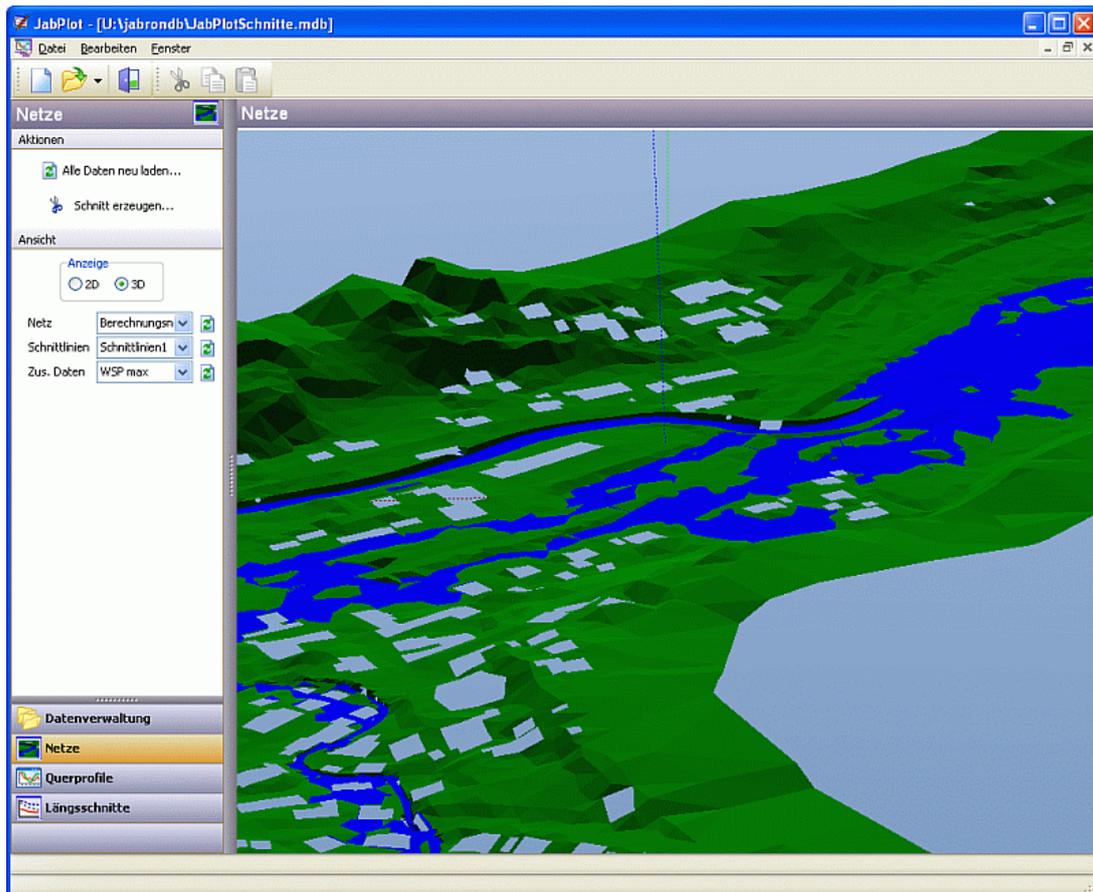


Abbildung 7.4: 3D-Ansicht in JabPlot

7.3 Ergebnisdarstellung mit MapView

HYDRO_AS-2D Ergebnisse können unter Windows mit dem Programm MapView in einer interaktiven Karte dargestellt werden. Um Ergebnisse in MapView darzustellen, sind folgende Schritte nötig:

1. MapView muss einmalig installiert werden. Das Setup finden Sie im HYDRO_AS-2D Installationsverzeichnis unter Hilfsprogramme oder auf der Hydrotec-Homepage unter www.hydrotec.de/MapView.
2. Der MapView-Datensatz (hmv-Datei) muss für das entsprechende Modell erstellt werden. Dazu nutzen Sie entweder die Benutzeroberfläche (siehe [Kapitel 6.9.1](#)) oder das Kommandozeilen Tool, siehe [Kapitel 6.9.2](#).

Ist der Datensatz vorhanden und MapView installiert, öffnen Sie den Datensatz in MapView durch einen Doppelklick oder über rechte Moustetaste auf den Datensatz (die hmv-Datei), dann öffnen mit MapView.



Abbildung 7.5: Ergebnisdarstellung in MapView

Das Konzept von MapView sieht vor, dass Berechnungsergebnisse auch ohne HYDRO_AS-2D-Installation, ohne SMS- oder JabPlot-Kenntnisse und ohne 2dm-Datei interaktiv betrachtet werden können.

Alle nötigen Informationen sind im MapView-Datensatz enthalten. Die Installation von MapView reicht aus, um die Daten zu visualisieren. Sie können die Ergebnisse mit der Download-URL für MapView an Dritte weitergeben.

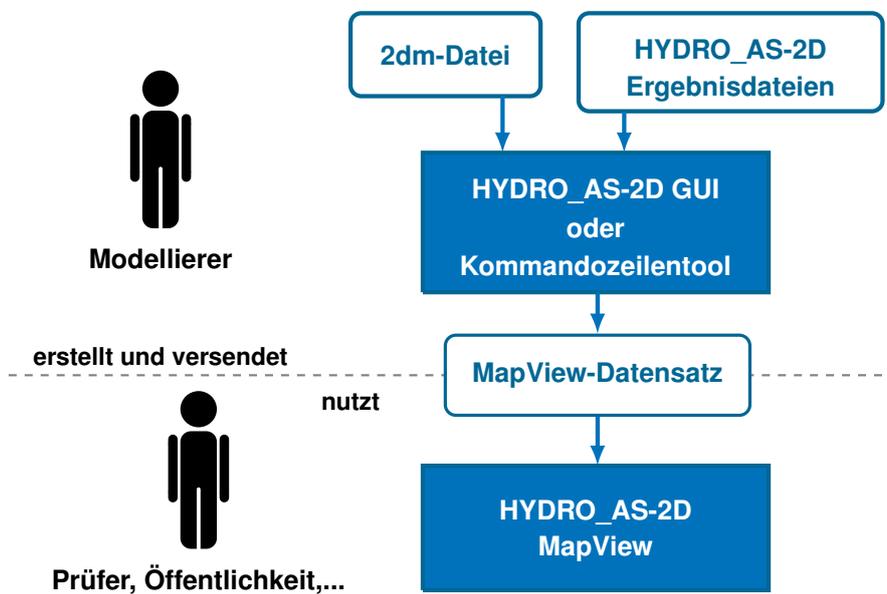


Abbildung 7.6: MapView-Datensatz als Schnittstelle

8 Steuerung mit Scripting

8.1 Einführung: Möglichkeiten der Scripting-Steuerung

Die Steuerung mit Skripten setzen Sie ein, um während der Simulation Daten des Modells abzufragen oder zu verändern. Sie entwickeln dazu Programm-Skripte in der Programmiersprache Lua, die von HYDRO_AS-2D in jedem internen Zeitschritt aufgerufen und abgearbeitet werden.

Einige Beispiele für konkrete Anwendungsfälle sind:

- **Bauwerkssteuerung: Schütze, Wehre, Brücken**

Mit HYDRO_AS-2D-Scripting können Sie Wasserstände oder Durchflüsse im Modell während der Simulation abfragen und in Abhängigkeit davon die Öffnung eines Durchlasses verändern.

Sie können mit HYDRO_AS-2D-Scripting auch die Kronenhöhe von 1D-Wehre verändern oder die Höhe von Konstruktionsunterkanten.

- **Zufluss-Zeitreihen von Dateien**

Zufluss-Zeitreihen zu einem HYDRO_AS-2D-Modell werden üblicherweise in SMS komfortabel mit den entsprechenden Editoren eingegeben.

Diese Technik kann aber ungünstig sein, wenn – beispielsweise in Umgebungen von Online- Vorhersagen - die Zufluss-Zeitreihen automatisiert aus anderen Datenquellen übernommen werden sollen. Der Weg über eine graphische Benutzeroberfläche ist in diesen Fällen hinderlich.

Mit HYDRO_AS-2D-Scripting können Sie externe Zufluss-Zeitreihen-Dateien einlesen, vorbereiten und während der Simulation an das Modell weiterreichen.

Evtl. erfordert ein Anwendungsfall sogar, die Zuflüsse in Abhängigkeit von internen Modellzuständen (Pegel, Durchflüsse) zu variieren. Auch dies kann mit dem Scripting erfolgen.

- **Bidirektionale Interaktion mit anderen Modellen**

HYDRO_AS-2D-Scripting ermöglicht die bidirektionale Interaktion mit anderen Modellen (zum Beispiel Grundwasser, 1D-Kanalnetzmodelle).

„Bidirektional“ bedeutet, dass die Eingaben und Ergebnisse der Modelle sich gegenseitig beeinflussen und die Modelle daher zeitlich verschränkt berechnet werden müssen.

Beispielsweise kann der von HYDRO_AS-2D berechnete Wasserstand in einem städtischen Gebiet als Grundlage für den Zufluss in ein Kanalnetz genutzt werden und andersherum der Überstau aus einem Kanalnetz als Zufluss in ein HYDRO_AS-2D-Modell.

Mit HYDRO_AS-2D-Scripting können Sie das Kanalnetzmodell z.B. in 5-Minuten-Schritten rechnen lassen, dabei die aktuellen Werte des 2D-Modells nutzen und die Ergebnisse des Kanalnetzmodells wieder in HYDRO_AS-2D weiterverwerten.

- **Deich-/Dammbrüche**

HYDRO_AS-2D ist aus numerischer Sicht geeignet Deichbrüche zu rechnen.

Die technische Umsetzung kann auf Basis von HYDRO_AS-2D-Scripting deutlich einfacher als ohne Scripting erfolgen: Entwickeln Sie hierzu ein HYDRO_AS-2D-Skript, in dem während der Simulation die Höhen des Deiches an der Bruchstelle herabgesetzt werden.

Sie können diesen Zeitpunkt fest vorgeben oder ihn anhand des Modellzustandes im Skript selbst bestimmen. Der Deichbruch soll beispielsweise simuliert werden, wenn an einer bestimmten Stelle ein Wasserstand überschritten wird oder wenn er an dieser Stelle seit wenigstens einem Tag überschritten wurde.

- **Spezielle Auswertungen zu Modellergebnissen**

Mit HYDRO_AS-2D-Scripting können statistische Auswertungen zu Modellergebnissen effizient und sehr genau während der Simulation berechnet werden.

Beispielweise können für einzelne Modellknoten während der Simulation Einstau-Dauerlinien berechnet werden: Wie lange war der Modellknoten während der Simulation wie hoch überflutet?

In bestimmten Fällen ist nicht nur die Überflutungshöhe relevant, sondern zur Berechnung der Gefährdung auch die Fließgeschwindigkeit an dem Knoten zu berücksichtigen. Auch diese Information kann das Skript benutzen.

Die Auswertung muss nicht auf den Ausgabezeitschritten von HYDRO_AS-2D basieren sondern kann die internen Berechnungszeitschritte nutzen.

Solche zusätzlichen Auswertungen können Sie in einem Datenformat ausgeben, das in Ihrem konkreten Anwendungsfall am besten weiter verwendet werden kann.

- **Varianten berechnen**

Üblicherweise berechnen Sie ausgehend von einem Ist-Zustand mit SMS/HYDRO_AS-2D Varianten, indem Sie den Ausgangsdatensatz in SMS einladen, eine Veränderung mit den Mitteln von SMS vornehmen, den veränderten Datensatz wieder abspeichern und die Simulation starten. Für den Fall, dass Sie sehr viele Varianten rechnen wollen, ist dieses Vorgehen fehleranfällig, weil von den vielen erforderlichen manuellen Arbeitsschritten evtl. einzelne nicht korrekt ausgeführt werden.

In ungünstigen Fällen stellen Sie nach der Erstellung der Varianten sogar fest, dass der Ausgangsdatensatz noch zu verändern ist und müssen dann alle Arbeitsschritte zur Erstellung der Variante wiederholen oder die gleichen Anpassungen in der bereits erstellten Variante durchführen.

HYDRO_AS-2D-Scripting kann in vielen Fällen die systematische Erstellung von Varianten erleichtern: Die Änderung des Modells wird dazu nicht manuell vorgenommen, sondern durch das Skript. Sie legen vom Ausgangsmodell keine Kopie mehr an, sondern arbeiten immer nur mit dem Ausgangsmodell und dem Skript, das die Variante erzeugt.

Diese Technik ist vor allem in solchen Fällen sinnvoll, in denen sehr viele Varianten zu rechnen sind und diese Varianten einfach systematisch zu erzeugen sind (viele unterschiedliche Deichbruchstellen, viele unterschiedliche Zufluss-Zeitreihen, unterschiedliche Wehrhöhen etc.).

Zur Nutzung von HYDRO_AS-2D-Scripting benötigen Sie grundlegende Programmierkenntnisse in der Sprache Lua. Lua ist eine „kleine“ Sprache: Falls Sie bereits in irgendeiner anderen Programmiersprache wie Python, Java, JavaScript, Visual-Basic, C# oder C++ Erfahrungen gesammelt haben, wird es Ihnen leicht fallen, die erforderlichen technischen Kenntnisse zu erlernen.

8.2 Technische Einführung

Um das HYDRO_AS-2D-Scripting zu aktivieren, muss es im HYDRO_AS-2D-Arbeitsverzeichnis eine Datei Data-in/control.lua geben. Der Name und die Verzeichnisstruktur müssen dabei wie folgt aussehen:

- > <Arbeitsverzeichnis>
 - > hydro_as-2d.2dm
 - > Data-in
 - > **control.lua**
 - > AS

Im Folgenden werden drei Skripte vorgestellt.

8.2.1 Grundprinzip des Scripting

Das folgende Skript führt noch keine sinnvollen Aktionen aus, sondern demonstriert lediglich grundlegende Konzepte.

```
-- Establish access to HYDRO_AS-2D-Modell
local hydroas = require("hydroas")

-- Reference on a gauge-node with name "Gauge1"
local gauge1 = hydroas.Node.new("Gauge1")

-- Return to HYDRO_AS-2D the functions which shall be called by HYDRO_AS-2D
return {
  -- Function 'step' will be called by HYDRO_AS-2D in each timestep
  step = function()
    print("Time:", hydroas.Global.timestepStart() , "h: ",gauge1:h())
  end
}
```

Das Skript wird beim Start der Simulation ausgeführt. Es enthält zwei Zeilen Code zur Initialisierung und es enthält eine Funktion, die von HYDRO_AS-2D während der Simulation mehrfach aufgerufen wird („Callback“).

```
-- Establish access to HYDRO_AS-2D-Modell
local hydroas = require("hydroas")
```

Die Lua-Scripting-Umgebung stellt den Zugang zum HYDRO_AS-2D-Modell her. Jede Kommunikation zwischen dem HYDRO_AS-2D-Modell und dem Skript erfolgt über die Variable `hydroas`.

```
-- Reference on a gauge-node with name "Gauge1"
local gauge1 = hydroas.Node.new("Gauge1")
```

Im HYDRO_AS-2D-Modell wurde als Knoten-Randbedingung ein Pegelpunkt mit der Bezeichnung „Gauge1“ vereinbart. Die Lua-Variable `gauge1` enthält nun eine Referenz auf diesen Knoten. Diese Referenz kann später dazu benutzt werden Daten des Knotens abzufragen oder zu verändern.

Beachten Sie, dass dieser Befehl keinen neuen Knoten anlegt, sondern lediglich eine Referenz auf einen bestehenden Knoten erzeugt.

```
-- Return to HYDRO_AS-2D the functions which shall be called by HYDRO_AS-2D
return {
  -- Function 'step' will be called by HYDRO_AS-2D in each timestep
  step = function()
    print("Time:", hydroas.Global.timestepStart() , "h: ",gauge1:h())
  end
}
```

Das Steuerungs-Skript liefert ein Objekt an HYDRO_AS-2D zurück. Dieses Objekt enthält Funktionen,

die später von HYDRO_AS-2D aufgerufen werden.

In diesem ersten Beispiel wird nur eine einzige Funktion mit den Namen „step“ zurückgeliefert.

Diese step-Funktion wird von HYDRO_AS-2D in jedem internen Berechnungszeitschritt ausgeführt.

Da HYDRO_AS-2D ein explizites Berechnungsverfahren benutzt, sind diese Berechnungszeitschritte häufig sehr klein. Sie kennen die Ausgabe von HYDRO_AS-2D auf der Kommandozeile, die die Länge des internen Berechnungszeitschrittes anzeigt:

```
SIMULATIONSZEIT =      300 Sekunden, ( DELT: 0.176701 2018-01-03 15:49:48)
SIMULATIONSZEIT =      600 Sekunden, ( DELT: 0.168347 2018-01-03 15:49:49)
...
ncycle, CPU, CPU / ncycle      530367      9346.238      1.7622210E-02
```

Insgesamt hat HYDRO_AS-2D in diesem Beispiel 530.367 Zeitschritte durchgeführt.

Die step-Funktion würde also 530.367-mal aufgerufen werden und 530.367 Zeilen auf die Kommandozeile ausgeben:

```
step = function()
    print("Time:", hydroas.Global.timestepStart(), "h: ", gauge1:h())
end
```

Die Ausgabe gibt für jeden internen Berechnungszeitschritt den Start des aktuellen Simulationszeitpunktes sowie die Einstautiefe am Pegel an.

Beachten Sie die Doppelpunktnotation beim Abruf der Wassertiefe von gauge1

```
print("Time:", hydroas.Global.timestepStart(), "h: ", gauge1:h())
```

Beim Abruf der Methoden von Objekten aus dem hydroas-Modul ist die Doppelpunktnotation zu verwenden, vgl. gauge1:h().

8.2.2 Beispiel: Wehrsteuerung

Im Beispiel <Installationsverzeichnis>/Beispiele/HAScripting ist das Skript Data-in/control.lua enthalten. Durch die Verwendung des Skripts wird das HYDRO_AS-2D-Modell um eine Polderbewirtschaftung erweitert.

Im Beispiel fließt das Gewässer von Süden nach Norden. Östlich des Gewässers befindet sich ein durchgehender Deich. Hinter Teilen des Deiches wurde ein Polder eingerichtet.

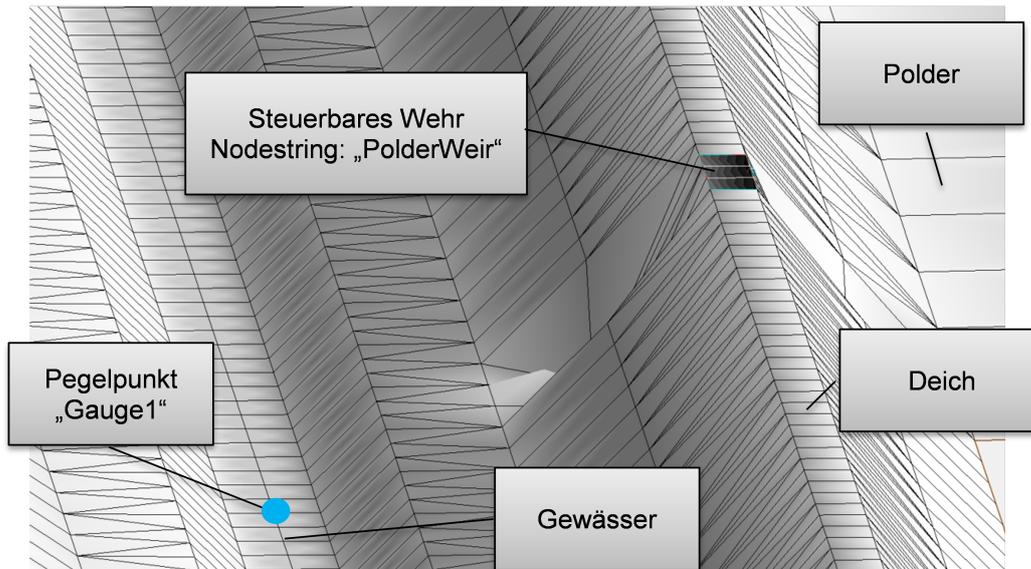


Abbildung 8.1: Ausschnitt des Beispielmodells HAScripting

Im südlichen Bereich des Deiches zwischen Polder und Gewässer befindet sich ein Wehr. Im nördlichen Bereich des Polders befindet sich ein Siel (nicht in der Grafik): Durch dieses Siel kann Wasser aus dem Polder zurück in das Gewässer fließen.

Wenn der Wasserstand im Gewässer sehr hoch ist, soll der Polder geflutet werden. Dazu soll das Wehr im Deich heruntergefahren werden.

Wenn das Hochwasser abgelaufen ist und sich im Gewässer wieder ein niedriger Wasserstand eingestellt hat, soll das Wehr wieder geschlossen werden.

Im HYDRO_AS-2D-Modell werden Randbedingungen ergänzt, auf denen das Skript aufbaut:

- Ein Kontrollquerschnitt „PolderWeir“ enthält die Modellknoten, die auf dem Wehr liegen.
- Ein Pegelpunkt „Gauge1“ markiert den Knoten, dessen Wasserstand für die Kontrolle des Wehres ausschlaggebend ist. In der Abbildung ist der Pegelpunkt blau markiert.

Im Skript werden mit `local` lokale Variablen angelegt, die zur Steuerung benötigt werden.

```
-- "PolderWeir" is a named controlsection in the HYDRO_AS-2D-model.
-- The set of nodes ("NodeSet") will contain all nodes in this nodestring.
local weir = hydroas.NodeSet.newByNodeString ("PolderWeir")
--Status-Variable indicates, whether the weir is closed or open
local weirStatus = 0; -- 0: Weir is closed; 1: Weir is open

-- "Outlet" is a culvert (Deutsch: "Siel")
local outlet = hydroas.Culvert.new ("Outlet")
-- save the original maximal capacity (qMax) of the outlet. When the outlet will be
opened,)
-- its original capacity will be reset)
local outletQMax = outlet.qMax ()
outlet:setQMax(0) -- Close the culvert

-- System is controlled by three gauges: 2 gauges in river and one gauge in polder
local gauge1 = hydroas.NodeSet.new("Gauge1") -- in river; near weir
local gauge2 = hydroas.Node.new("Gauge2") -- in river; near culvert
local gaugePolder = hydroas.Node.new("GaugePolder") -- in polder
```

In der Variable `textttweir` sind die vier Wehrknoten als ein `NodeSet` mithilfe des Nodestrings „PolderWeir“ zusammengefasst.

Die Verwendung von `NodeSet` macht das Skript übersichtlicher und weniger fehleranfällig:

- Mit `NodeSet` sind viele Knoten des Modells auf einmal ansprechbar. Theoretisch könnte man für jeden Knoten auf dem Wehr ein `hydroas.Node` anlegen und diese einzeln steuern. Der Code würde dadurch aber sehr viel unleserlicher.
- Sie können das `NodeSet` aus einem - oder auch mehreren - benannten Kontrollquerschnitten bilden. Auf diese Weise legen Sie graphisch in SMS fest, welche Knoten Sie ansprechen wollen.

Das Skript benutzt die folgenden zusätzlichen Variablen:

- `weirStatus` ist der aktuelle Zustand des Wehres (offen/geschlossen).
- `gauge1` referenziert den Pegelpunkt oberhalb des Wehres.
- `outlet` referenziert das Siel am Auslauf des Polders. Der ursprüngliche Wert des maximalen Abflusses wird in `outletQMax` gespeichert. In der folgenden Zeile wird der maximale Abfluss auf null gesetzt. Dadurch ist das Siel nun geschlossen.
- `gauge2` referenziert den Pegelpunkt auf der Flusseite des Siels
- `gaugePolder` referenziert den Pegelpunkt auf der Polderseite des Siels.

Die `step`-Funktion besteht aus zwei Blöcken. Im ersten Block wird das Wehr gesteuert, im zweiten der Abfluss aus dem Polder.

```
return {
  step = function()
    -- I: weir
    .
    .
    .
    -- II: polder
    .
    .
    .
    end
  end
}
```

Wehrsteuerung:

```
-- if the water level is very high and the weir is still closed
-- the weir shall be opened to flood the polder
if weirStatus == 0 and gauge1:w() > 104 then
  -- lower the weir
  weir:setZ(100)
  weirStatus = 1
  print("***** Open weir:", hydroas.Global.timestepStart())
-- if flood has finished weir shall be closed
elseif weirStatus == 1 and gauge1:w() < 101 then
  -- raise the weir to its original height
  weir:setZ(109.96)
  weirStatus = 0
  print("***** Close weir:", hydroas.Global.timestepStart())
end
```

Mit `gauge1.w()` wird der Wasserspiegel am Pegelpunkt „Gauge1“ abgefragt.

Wenn der Wasserspiegel am Pegelpunkt „Gauge1“ die Marke 104 (abs. Höhe in m) übersteigt und das Wehr noch geschlossen ist, wird das Wehr geöffnet, indem die Sohlhöhe (Z) der vier Punkte, die das Wehr `weir` bilden mit `weir:setZ(100)` auf 100 (abs. Höhe in m) abgesenkt wird. Der Status `weirStatus` wird auf 1, d.h. „offen“, geändert.

Mit `print` wird eine Meldung auf die Konsole geschrieben. `hydroas.Global.timestepStart()` liefert den Anfangszeitpunkt des aktuellen internen Zeitschritts.

Wenn der Wasserspiegel am Pegelpunkt „Gauge1“ unter 101 (abs. Höhe in m) fällt und das Wehr offen ist, wird es wieder geschlossen, indem die Wehr-Knoten wieder die ursprüngliche Höhe von 109,96 (abs. Höhe in m) zugewiesen bekommen. Der Status wird wieder auf 0, also „geschlossen“, gesetzt.

Poldersteuerung:

Am nördlichen Ende hat der Polder ein Siel, durch das der Inhalt des Polders wieder in das Gewässer abfließen kann.

Im HYDRO_AS-2D-Datensatz ist dieses Siel als ein Durchlass abgebildet, der nur in einer Richtung durchflossen werden kann.

```
--Outlet is closed: free capacity in river and Polder is filled?
-- Open outlet
if outlet:qMax() == 0 and gauge2:w() < 96 and gaugePolder:h() > 1 then
  outlet:setQMax(outletQMax)
  print("***** Open outlet:", hydroas.Global.timestepStart())
-- Outlet is open: water in river is too high: Close it
elseif outlet:qMax() > 0 and gauge2:w() > 99 then
  outlet:setQMax(0)
  print("***** Close outlet:", hydroas.Global.timestepStart())
end
```

Zur Steuerung des Auslasses wird die Kapazität des Durchlasses verändert:

- Der maximale Abfluss wird auf 0 (m^3/s) gesetzt, um den Durchlass zu schließen.
- Er wird auf den ursprünglichen Wert, der in `outletQMax` zu Beginn gespeichert wurde, zurückgesetzt, um den Durchlass zu öffnen.

Steuerkriterien für das Auslass-Siel sind:

- Das Auslass-Siel ist standardmäßig geschlossen.
- Wenn der Polder gefüllt ist und an der Einleitungsstelle des Siels im Gewässer der Wasserstand gering ist, soll der Polder geöffnet werden. Das im Polder zwischengespeicherte Wasser wird damit zur ablaufenden Welle hinzugefügt.
- Wenn das Siel geöffnet ist und der Wasserspiegel an der Einleitungsstelle des Siels hoch ist - d.h. im Gewässer die Marke 99 (abs. Höhe in m) überschritten hat, soll das Siel wieder geschlossen werden.

Mit `gaugePolder:h()` wird die Wassertiefe am Pegelpunkt im Polder abgefragt.

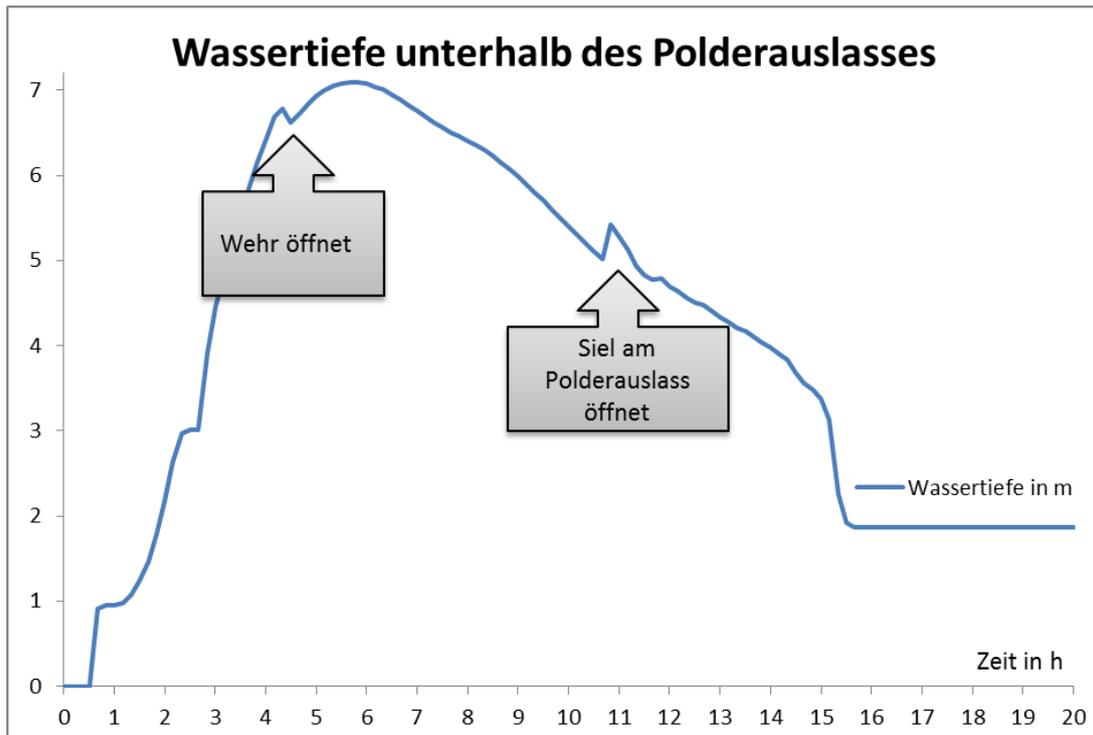


Abbildung 8.2: Plot der Wassertiefe am Polderauslass

8.2.3 Statistische Auswertung

In diesem Abschnitt wird ein Skript zur Berechnung einer Dauerlinie für die Einstautiefe von einzelnen Knoten vorgestellt.

Das Skript schreibt für einen ausgewählten Knoten eine Datei, die angibt, wie lange die jeweilige Wassertiefe am Knoten überschritten wurde. Die Wassertiefe variiert dabei in Zentimeterschritten. Die folgende Datei ist eine Beispielausgabe. Sie zeigt an, dass der Knoten 50 Sekunden lang wenigstens 2,13 m eingestaut war, 530 Sekunden lang wenigstens 2,12 m eingestaut war usw.

h [m]	duration [sec]
2.13	50
2.12	530
2.10	1661
...	...
0.01	7334
0.00	7600

Der Lua-Code zur Erzeugung dieser Ausgabe deklariert eine wiederverwendbare Funktion `DurationCurve` und wendet diese Funktion auf die beiden Pegel „Gauge1“ und „Gauge2“ an:

```
-- Establish access to HYDRO_AS-2D-Modell
local hydroas = require(" hydroas")

-- Calculates duration-curves of a node.
-- Resolution of the DurationCurve is 1cm.
-- nodeName describes the node to be watched
```

```

local function DurationCurve(nodeName)

-- table with durations
-- durations[0] contains the duration for height 0 cm .. 1 cm
-- durations[1] contains the duration for height 1 cm .. 2 cm
-- durations[2] contains the duration for height 2 cm .. 3 cm
-- durations[n] contains the duration for height n cm .. n+1 cm
local durations = {}
-- Maximal index n in the durations-table
local maxIndex = 0
local node = hydroas.Node.new(nodeName)

return {

-- Watch the node in the current timestep and update the durations-table
-- This function must be called once in each timestep.
step = function()
    local index = math.floor(node:h() * 100 )
    -- add the length of the current timestep to the duration of the class
    index.
    -- Lua-trick: (durations[index] or 0) is 0 if durations[index] is unset.
    durations[index] = (durations[ index] or 0) + hydroas.Global.timestepLength()
    maxIndex = math.max(maxIndex ,index)
end,

-- write the duration-curve to file
-- Filename is the name of the node. Extension is ".dc"
writeToFile = function()
    local file = io.open(nodeName .. ".dc","w")
    file:write("h[m]          duration[sec]\n ")
    local sumDuration = 0 -- Sum-up the duration of each height
    -- Loop from maxIndex down to 0
    for index = maxIndex,0,-1 do
        sumDuration = sumDuration + (durations[index] or 0)
        file:write(string.format("%10.2f %10.0f\n",index / 100, sumDuration))
    end
    file:close()
end
}
end

-- Instanciates two DurationCurves.
local d1 = DurationCurve("Gauge1")
local d2 = DurationCurve("Gauge2")

-- Return to HYDRO_AS-2D the functions which shall be called by HYDRO_AS-2D
return {
    step = function()
        -- Update both duration-curves
        d1.step(); --
        d2.step();
    end,
    -- Function 'close' will be called before HYDRO_AS-2D end simulation
    close = function()
        -- Output of the duration-curves
        d1.writeToFile()
        d2.writeToFile()
    end
}

```

```
end
}
```

Der Code ist bereits recht komplex und zeigt viele interessante Details:

Implementierung Du-Die Funktion `DurationCurve` implementiert die Dauerlinie als Lua-Klasse. Jede `DurationCurve` Instanz enthält eine Table (`duration`), die für jedes Wassertiefenintervall abspeichert, wie lange die Wassertiefe am Knoten in diesem Intervall lag. Die Intervalle sind durch Zentimeterschritte definiert.

„DurationCurve“ liefert ein Objekt zurück, das zwei Methoden (`step` und `writeToFile`) unterstützt:

```
local function DurationCurve(nodeName)
...
local durations = {}
...
return {
...
step = function()
...
durations[index] = ...
...
end,
writeToFile = function()
...
end
}
end
```

Die beiden Methoden arbeiten auf den lokalen Variablen der Funktion `DurationCurve`. Diese Technik heißt Closure.

Schreiben einer Datei Lua kann Dateien zum Schreiben öffnen, Text in diese Datei ausgeben und die Datei wieder schließen:

```
local file = io.open(nodeName .. ".dc", "w")
file:write("h[m]          duration[sec]\n")
...
file:close()
```

Das Zeilenumbruchzeichen „\n“ kennen Sie eventuell von JavaScript, Java, Python oder C.

Ausgaben formatieren Diese Zeile erzeugt einen formatierten Text und schreibt ihn in die zuvor geöffnete Datei:

```
file:write(string.format("%10.2f %10.0f\n", index / 100,
sumDuration))
```

Die Platzhalter „%10.2f“ geben an, dass eine Zahl mit insgesamt 10 Stellen und 2 Nachkommastellen ausgegeben werden soll.

close-Methode für HYDRO_AS-2D Die Datei soll einmal am Ende der Simulation geschrieben werden. Dazu wird außer der schon bekannten `step`-Funktion die Funktion `close` benutzt:

```
return {
```

```

step = function()
    ...
end,
-- Function 'close' will be called before HYDRO_AS-2D
terminates the simulation
close = function()
    -- Output of the duration-curves
    d1.writeToFile()
    d2.writeToFile()
end
}

```

HYDRO_AS-2D-Scripting kennt zwar eine close-Funktion, aber keine init-Funktion. Die init-Funktion wird nicht benötigt, da das gesamte Skript am Anfang einmal initial ausgeführt wird.

Beispielsweise sind die Anweisungen

```

local d1 = DurationCurve("Gauge1")
local d2 = DurationCurve("Gauge2")

```

Teil der Initialisierung.

Länge des aktuellen Zeitschrittes Die Länge des aktuellen HYDRO_AS-2D-Zeitschrittes ist abfragbar:

```

durations[index] = ... + hydroas.Global.timestepLength()

```

8.3 Verfügbare Callbacks

Das control.lua-Skript liefert eine „Table“ mit Callback-Funktionen aus. Die folgenden Callback-Funktionen werden von HYDRO_AS-2D-Scripting unterstützt:

- **step:** Wird in jedem internen Berechnungszeitschritt aufgerufen. Genaugenommen wird diese Funktion am Ende eines Berechnungszeitschrittes aufgerufen: Die von HYDRO_AS-2D berechneten Werte für die Wassertiefen und Geschwindigkeiten, auf die das Scripting Zugriff hat, sind die Werte, die für das Ende des Zeitschrittes gelten. Wenn beispielsweise ein Zeitschritt eine Sekunde dauert und er die Simulationszeit zwischen „3.600 Sekunden“ und „3.601 Sekunden“ abbildet, dann liefert HYDRO_AS-2D für den `step`- Aufruf dieses Zeitschrittes Wassertiefen zurück, die zum Zeitpunkt „3.601 Sekunden“ gültig sind. Veränderungen an diesen Wassertiefen werden im Zeitschritt ab „3.601 Sekunden“ wirksam. Die Funktion `hydroas.Global.timestepstart()` gibt den Anfang des Zeitschrittes zurück. Der Aufruf in der `step`-Funktion liefert also in diesem Beispiel „3.600 Sekunden“.
- Wenn Sie Anfangsbedingungen der Wassertiefen und Geschwindigkeiten abfragen und verändern wollen, führen Sie dies nicht in `step`, sondern im Initialisierungsteil des `control.lua`- Skripts aus.
- **writeResult:** Diese Funktion wird in den Q_Strg-Zeitschritten von HYDRO_AS-2D ausgeführt. Modellierer geben diese Zeitspanne in SMS ein:

Name	Value
Rechenlauf fortsetzen?	0 - bei T=0 starten
Simulationszeit [s]	27600.0
Zeitintervall SMS [s]	900.0
Zeitintervall Q_Strg [s]	300.0
Hmin [m]	0.01
VELMAX [m/s]	15.0
Amin	0.1
CMUVISC	0.6
CFL	0.8

Abbildung 8.3: Global Parameters: Zeitintervall Q_Strg [s]

In diesem Beispiel wird die `writeResult`-Funktion also nach jeweils 300 Sekunden Simulationszeit aufgerufen.

Diese Funktion nutzen Sie beispielsweise, um eigene Zwischenergebnisse in Dateien zu schreiben.

Intern wird `writeResult` nach dem Aktualisieren der in den internen Zeitschritten berechneten Werte aufgerufen. Dadurch liefert die Funktion `hydroas.Global.timestepstart()` den Startzeitpunkt des nächsten anstehenden Zeitschritts, bzw. das Ende des aktuellen Zeitschritts. Im Beispiel von oben wird der Aufruf von `hydroas.Global.timestepstart()` in `writeResult` also „3.601 Sekunden“ liefern.

- `close`: Diese Funktion wird von HYDRO_AS-2D genau einmal aufgerufen, bevor HYDRO_AS-2D die Simulation beendet. Diese Funktion benutzen Sie beispielsweise um Dateien zu schreiben oder zu schließen oder um abschließende Auswertungen vorzunehmen.

Die Liste der Callback-Funktionen enthält keine `init`-Funktion. Diese Callback-Funktion ist nicht erforderlich, da das Hauptskript vor dem `return`-Statement sowohl die Initialisierung der Skripte als auch die Initialisierung des Modells – beispielsweise mit Anfangsbedingungen oder Zeitreihen – enthält.

8.4 Allgemeine Hinweise zur Lua-Programmierung

Info: In Lua werden Objekte „Table“ genannt.

8.4.1 Methodentypen

Die folgende Referenz unterscheidet zwischen **statischen Klassenfunktionen** und **Objektmethoden**:

- Statische Klassenfunktionen werden meist verwendet, um Objekte anzulegen. Die meisten statischen Klassenfunktionen beginnen daher mit „new“.

Beispiele für statische Klassenfunktionen:

- `Culvert.new`
- `Node.new`

Einige statische Klassenfunktionen haben auch die Eigenschaften von Konstanten. Da Lua das Konzept der Konstanten nicht unterstützt, vereinbart die HYDRO_AS-2D-Scripting-Schnittstelle parameterfreie Funktionen an Stelle von Konstanten:

Beispiel:

```
- Culvert.Direction_Both
```

- Objektmethoden fragen Objekte ab oder verändern sie. In HYDRO_AS-2D-Scripting müssen Sie Objektfunktionen mit der „.“-Notation aufrufen. Beispiele sind:

```
- aNode:h(); // Fragt Wasserstand am Knoten aNode ab
- aNode:setZ(100); Setzt Z-Wert (absolute Höhe) in aNode
```

Die „.“-Notation bei Objektmethoden wird in den meisten anderen objektorientierten Sprachen nicht verwendet, daher ist die Verwechslung von „.“ und „.“ eine häufige Fehlerquelle.

Wenn Sie statt des „.“ einen „.“ verwenden, antwortet das Programm mit der Fehlermeldung „attempt to index a nil value (local 'self')“ beim Aufruf einer Funktion.

8.4.2 Ganze Zahlen vs. Fließkommazahlen

Lua unterscheidet intern nicht zwischen Fließkommazahlen und ganzen Zahlen: Alle Zahlen werden in Lua als 8-Byte Fließkommazahlen abgebildet. Damit sind zwar ganze Zahlen bis zur Größe 10^{15} exakt abbildbar, trotzdem ist in vielen Fällen konzeptuell zwischen Fließkommazahlen und ganzen Zahlen zu differenzieren: Beispielsweise wird in manchen Fällen ein Index erwartet/ zurückgeliefert, so dass der Kontext nur ganze Zahlen sinnvoll unterstützt.

Die Dokumentation unterscheidet daher zwischen Number (Integer) und Number (Float), obwohl rein technisch gesehen alle Zahlen in Lua Fließkommazahlen und damit „Float“ sind.

8.4.3 Einheiten

HYDRO_AS-2D arbeitet stets mit SI-Einheiten. Die Einheiten werden daher nicht bei jeder Funktion explizit genannt:

SI-Einheiten in HYDRO_AS-2D

Höhen/Längen	Meter	m
Flächen	Quadratmeter	m ²
Geschwindigkeiten	Meter pro Sekunde	m/s
Zeiten	Sekunden	s
Abfluss	Kubikmeter pro Sekunde	m ³ /s

8.4.4 Anlegen von Objekten

Der Aufruf von statischen „newXY“-Klassenmethoden legt stets nur Objekte an, die den Zugriff auf HYDRO_AS-2D-Elemente erlauben. Die Methode legt keine neuen Elemente in HYDRO_AS-2D selbst an.

Beispiel: Die Anweisung `local aNode = Node.new(4711)` legt keinen neuen Knoten im HYDRO_AS-2D-Berechnungsnetz an, sondern erlaubt nur über die Variable `aNode` den Zugriff auf den bereits existierenden Knoten 4711. Falls der Knoten 4711 nicht existiert, führt dies zu einem Scripting-Fehler.

Das Anlegen dieser Zugriffsinstanzen ist in vielen Fällen relativ aufwendig. Für HYDRO_AS-2DScripting wird dringend empfohlen, diese Instanzen stets während der Initialisierungsphase anzulegen und während der einzelnen Zeitschritte diese einmalig angelegten Objekte nur noch zu nutzen.

control.lua: Schlechter Stil	control.lua: Guter Stil
<pre>return { step = function() { local n = Node.new("GaugeNode"); n.setZ(n.z() + .01); } }</pre>	<pre>local n = Node.new("GaugeNode"); return { step = function() { n.setZ(n.z() + .01); } }</pre>

Sie können die Lua-Funktion `require` nutzen, um Instanzen nur einmal anzulegen und diese einmalig angelegten Instanzen überall zu nutzen. Legen Sie beispielsweise eine Datei `Model.lua` an, in der der Zugriff auf alle Elemente eingerichtet wird, die Sie für eine Steuerung benötigen:

```
local hydroas = require("hydroas")
return {
  gauge1 = Node.new("Gauge1"),
  gauge2 = Node.new("Gauge2"),
  inflow = InflowBC.new("Inflow1")
}
```

Danach können Sie auf diese Elemente mit `require` zugreifen:

```
local model = require("Model")
return {
  step = function() {
    if model.gauge1.h() > 3 then
      model.gauge1.setZ(model.gauge1.z() + .01);
    end
  }
}
```

8.5 Überblick über alle Scripting-Klassen

Die folgende Tabelle gibt einen kurzen Überblick über alle Lua-Klassen der Scripting-Umgebung. Für Details schlagen Sie in [Kapitel 8.4](#) nach. Diese Tabelle nutzen Sie lediglich als einen ersten Einstieg:

<code>hydroas.Culvert</code>	Zugriff auf die Daten von Durchlässen: Die für die Simulation wesentlichen Beschreibungen des Durchlasses können abgefragt und gesetzt werden. Der aktuelle Abfluss kann abgefragt werden. Beispielsweise kann die Höhe eines Durchlasses verändert werden, um ein steuerbares Schütz zu modellieren.
<code>hydroas.CulvertSet</code>	Zugriff und Änderung einer Gruppe von 1D-Durchlasselementen: Wenn 1D-Durchlasselemente dazu benutzt werden Durchlässe abzubilden, ist es oft erforderlich mehrere 1D-Durchlasselemente nebeneinander einzusetzen. Diese Gruppe von 1DDurchlasselemente kann mit dieser Klasse einheitlich angesprochen werden, um beispielsweise diesen Durchlass zu steuern.
<code>hydroas.Element</code>	Zufriff auf einen Teil der Elementdaten.
<code>hydroas.Global</code>	Abfrage und ggf. Veränderung einiger globaler Daten, wie den Informationen zum aktuellen Zeitschritt.
<code>hydroas.GrainFraction</code>	Abfrage und ggf. Veränderung einiger Daten von Kornfraktionen. Nur für Nutzung mit den Stofftransportmodulen.

<code>hydroas.InflowBC</code>	Zugriff auf die Eigenschaften einer Zufluss-Randbedingung: Diese Klasse können Sie beispielsweise benutzen, um während der Simulation Zuflusswerte aus externen Datenquellen zu lesen oder auf Basis von internen Zuständen im Modell zu manipulieren.
<code>hydroas.Node</code>	Zugriff auf die Daten eines Modellknotens: Neben den geometrischen Eigenschaften können auch der aktuelle Wasserstand sowie die aktuelle Geschwindigkeit abgefragt und verändert werden.
<code>hydroas.NodeSet</code>	Diese Klasse benutzen Sie um Knotenmengen zu manipulieren. Wenn Sie beispielsweise ein Wehr über Knotenhöhen im 2D-Modell abbilden und diese Wehrknoten sollen gemeinsam abgesenkt bzw. erhöht werden, benutzen Sie ein <code>NodeSet</code> statt einer Liste von <code>Nodes</code> .
<code>hydroas.NodeString</code>	Zugriff auf die Daten eines <code>NodeStrings</code> : Die einzelnen Knoten und der Durchfluss können abgefragt werden.
<code>hydroas.OutflowBC</code>	Zugriff auf die Eigenschaften einer Abfluss-Randbedingung.
<code>hydroas.Types</code>	Typen von Durchlässen, Wehren etc. werden in <code>HYDROAS-2DScripting</code> mit ganzen Zahlen kodiert. Diese ganzen Zahlen sind im Modul <code>hydroas.Types</code> als Konstanten hinterlegt.
<code>hydroas.Weir1D</code>	Zugriff und Änderung von 1D-Wehrelementen: Einsatzbereich kann die Wehrsteuerung sein.
<code>hydroas.Weir1DSet</code>	Zugriff und Änderung einer Gruppe von 1D-Wehrelementen: Wenn 1D-Wehrelemente dazu benutzt werden Wehre abzubilden, ist es oft erforderlich mehrere 1D-Wehrelemente nebeneinander einzusetzen. Diese Gruppe von 1D-Wehrelemente kann mit dieser Klasse einheitlich angesprochen werden, um beispielsweise dieses Wehr zu steuern.

8.6 Referenz Scripting

8.6.1 Klasse `hydroas.Culvert`

Diese Klasse erlaubt den Zugriff auf Durchlässe.

8.6.1.1 Statische Klassenfunktionen

`Culvert.Direction_1_2()`

Parameter: -

Rückgabe: Number (Integer)

`Culvert.Direction_1_2()` dient zum Setzen bzw. Auswerten der Funktionen `aCulvert:setDirection` und `aCulvert:direction`.

`Culvert.Direction_1_2()` zeigt an, dass der Durchlass nur vom ersten Knoten (vgl. `aCulvert:node1`) in Richtung des zweiten Knotens (vgl. `aCulvert:node2`) durchflossen werden kann.

Codebeispiel:

```
aCulvert1:setDirection(Culvert.Direction_1_2());
if (aCulvert2:direction() == Culvert.Direction_1_2()) then ...
```

`Culvert.Direction_2_1()`

Parameter: -

Rückgabe: Number (Integer)

`Culvert.Direction_2_1()` dient zum Setzen bzw. Auswerten der Funktionen `aCulvert:setDirection` und `aCulvert:direction`.

`Culvert.Direction_2_1()` zeigt an, dass der Durchlass nur vom zweiten Knoten (vgl. `aCulvert:node2`) in Richtung des ersten Knotens (vgl. `aCulvert:node1`) durchflossen werden kann.

Codebeispiel:

```
aCulvert1:setDirection(Culvert.Direction_2_1());
if (aCulvert2:direction() = Culvert.Direction_2_1()) then ...
```

`Culvert.Direction_Both()`

Parameter: -

Rückgabe: Number (Integer)

`Culvert.Direction_Both()` dient zum Setzen bzw. Auswerten der Funktionen `aCulvert:setDirection` und `aCulvert:direction`.

`Culvert.Direction_Both()` zeigt an, dass der Durchlass in beide Richtungen durchflossen werden kann.

Codebeispiel:

```
aCulvert1:setDirection(Culvert.Direction_Both());
if (aCulvert2:direction() == Culvert.Direction_Both()) then ...
```

`Culvert.isCulvertType(aType)`

Parameter:

aType: hydroas.Types - Typ

Rückgabe: Bool

Liefert true, wenn aType ein Durchlasstyp ist, sonst false.

`Culvert.new(name)`

Parameter:

name: String - Bezeichnung des Durchlasses.

Die Bezeichnung des Durchlasses wird im Eingabefeld „Name“ in SMS angegeben.

Rückgabe: Culvert-Instanz

8.6.1.2 Objektmethoden

```
aCulvert:coefficient()
```

Parameter: -

Rückgabe: Number (Float)

Liefert den Abflusskoeffizienten des Durchlasses.

```
aCulvert:direction()
```

Parameter: -

Rückgabe: Number (Integer)

Gibt an, in welche Richtungen der Durchlass durchflossen werden kann. Die Rückgabewerte dieser Funktion entsprechen den Konstanten `Culvert.Direction_1_2()`, `Culvert.Direction_2_1()` und `Culvert.Direction_Both()`.

```
aCulvert:height()
```

Parameter: -

Rückgabe: Number (Float)

Höhe des Durchlasses. Beachten Sie, dass sowohl Höhe als auch Breite bei einem kreisförmigen Durchlass dem Durchmesser entsprechen.

```
aCulvert:name()
```

Parameter: -

Rückgabe: String

Liefert den Name, der in SMS angegeben wurde, oder, wenn der Name nicht gesetzt ist, die Nummer des ersten Knotens als String.

```
aCulvert:node1Id()
```

Parameter: -

Rückgabe: Number (Integer)

Nummer des Anfangsknotens. Um Daten des Knotens zu verändern oder abzufragen, kann ein Node-Objekt angelegt werden:

```
local aNode = Node.new(aCulvert:node1Id())
```

```
aCulvert:setZ1(aNode:getZ() + 1)
```

```
aCulvert:node2Id()
```

Parameter: -

Rückgabe: Number (Integer)

Nummer des Endknotens. Um Daten des Knotens zu verändern oder abzufragen, kann ein Node-Objekt angelegt werden:

```
local aNode = Node.new(aCulvert:node2Id())
aCulvert:setZ2(aNode:getZ() + 1)
```

```
aCulvert:q()
```

Parameter: -

Rückgabe: Number (Float)

Abfluss durch diesen Durchlass im zurückliegenden Zeitschritt.

```
aCulvert:qMax()
```

Parameter: -

Rückgabe: Number (Float)

aCulvert:qMax() liefert den maximal möglichen Abfluss dieses Durchlasses zurück. Dieser Wert limitiert den aus der Durchlassgeometrie und den Abflussverhältnissen im Einlass- und Auslass-Bereich berechneten Abfluss.

```
CaCulvert:setCoefficient(coeff)
```

Parameter:

coeff: Number (Float) - Abflusskoeffizient

Rückgabe: -

Setzt den Abflusskoeffizienten des Durchlasses auf den vorgegebenen Wert.

```
aCulvert:setDirection(dir)
```

Parameter:

dir: Number (Integer) - gewünschte Richtung

Benutzen Sie Culvert.Direction_1_2(), Culvert.Direction_2_1() oder Culvert.Direction_Both() um den Parameter zu setzen.

Rückgabe: -

Setzt die möglichen Durchflussrichtungen.

```
aCulvert:setHeight(h)
```

Parameter:

h: Number (Float) - Durchlasshöhe in m

Rückgabe: -

Setzt die Höhe des Durchlasses: Wenn Sie steuerbare Schütze mit HYDRO_AS-2D abbilden wollen, können Sie das Schütz als Durchlass modellieren und mit dieser Funktion die Höhe des Schützes setzen.

Beachten Sie, dass sowohl Höhe als auch Breite bei einem kreisförmigen Durchlass dem Durchmesser entsprechen. Daher werden durch den Aufruf dieser Funktion für kreisförmige Durchlässe beide Werte neu gesetzt.

```
aCulvert:setQMax(q)
```

Parameter:

q: Number (Float) - Abflusswert in m³/s

Rückgabe: -

Setzt den maximal möglichen Abfluss dieses Durchlasses.

Dieser Wert limitiert den aus der Durchlassgeometrie und den Abflussverhältnissen im Einlass- und Auslassbereich berechneten Abfluss.

Mit `aCulvert:setQMax(0)` kann der Durchlass auch vollständig geschlossen werden.

```
aCulvert:setWidth(w)
```

Parameter:

w: Number (Float) - Breite in m

Rückgabe: -

Setzt die Breite des Durchlasses. Beachten Sie, dass sowohl Höhe als auch Breite bei einem kreisförmigen Durchlass dem Durchmesser entsprechen. Daher werden durch den Aufruf dieser Funktion für kreisförmige Durchlässe beide Werte neu gesetzt.

```
aCulvert:setZ1(z)
```

Parameter:

z: Number (Float) - absolute Höhe

Rückgabe: -

Setzt die absolute Höhe der unteren Kante des Anfangsknotens.

```
aCulvert:setZ2(z)
```

Parameter:

z: Number (Float) - absolute Höhe

Rückgabe: -

Setzt die absolute Höhe der unteren Kante des Endknotens.

aCulvert:type()

Parameter: -

Rückgabe: Number (Integer)

Liefert den Typ des Durchlasses: Nutzen Sie die Funktionen in `Types.Culvert` um den Typ auszuwerten:

```
if aCulvert:type() == hydroas.Types.Culvert.Rectangle()
then
  ...
end
```

Werten Sie die Typen nicht anders aus – es ist nicht sichergestellt, dass die konkreten Werte mit denen ein Typ kodiert ist sich künftig nicht ändern.

aCulvert:width()

Parameter: -

Rückgabe: Number (Float)

Liefert die Breite des Durchlasses. Beachten Sie, dass sowohl Höhe als auch Breite bei einem kreisförmigen Durchlass dem Durchmesser entsprechen.

aCulvert:z1()

Parameter: -

Rückgabe: Number (Float)

Liefert die absolute Höhe der unteren Kante des Anfangsknotens.

aCulvert:z2()

Parameter: -

Rückgabe: Number (Float)

Liefert die absolute Höhe der unteren Kante des Endknotens.

8.6.2 Klasse `hydroas.CulvertSet`

Diese Klasse erlaubt den Zugriff auf mehrere Durchlässe gleichzeitig. Sie wird für größere Durchlässe, die durch parallele Nodestrings abgebildet werden, benutzt.

8.6.2.1 Statische Klassenfunktionen

```
CulvertSet.new(namePattern)
```

Parameter:

namePattern: regEx - Muster für Namen

Rückgabe: CulvertSet

Das `namePattern` ist ein regulärer Lua-Ausdruck. Beachten Sie die Lua-Dokumentation zu regulären Ausdrücken.

In vielen Fällen wird der reguläre Ausdruck das Anfangsstück eines Textes beschreiben. Um Namen mit vorgegebenem Anfangsstück auszuwählen, besteht der Ausdruck aus dem Anfangsstück, einem vorangestelltem „^“ und nachgestelltem „*\$“ (^DN1200_*\$ oder ^Anlage_*\$).

Um sicher zu gehen, dass keine falschen Elemente ausgewählt werden, kann der Ausdruck ggf. auch restriktiver gestaltet werden: Die Ausdrücke ^DN1200_%d+\$ oder ^Anlage_%d+\$ wählen nur Elemente aus, die mit dem Anfangsstück beginnen und dann mit einer Ziffernfolge enden.

Sie sollten darauf achten, die Anfangs- und Endkennner (^ und \$) zu benutzen, um nicht versehentlich falsche Objekte mit auszuwählen. Beispielsweise würde das Pattern `Anlage_%d+` auch Elemente mit dem Namen `WehrAnlage_2` auswählen. Die Funktion wirft einen Fehler, wenn keine 1D-Durchlasselemente durch den Ausdruck beschrieben werden.

Beispiel:

```
local culvset = hydroas.CulvertSet.new("MyDN_%d+$")
```

8.6.2.2 Objektmethoden

```
aCulvertSet:anyCoefficient()
```

Parameter: -

Rückgabe: Number (Float)

Liefert den Abflusskoeffizienten eines der 1D-Durchlasselemente zurück, die in dem `CulvertSet` enthalten sind.

Diese Funktion ist sinnvoll einsetzbar, wenn das Skript davon ausgehen kann, dass alle 1D-Durchlasselemente denselben Abflusskoeffizienten haben.

Dies ist insbesondere dann der Fall, wenn zuvor `aCulvertSet:setCoefficient(x)` aufgerufen wurde.

```
aCulvertSet:anyHeight()
```

Parameter: -

Rückgabe: Number (Integer)

Liefert die relative Höhe eines der 1D-Durchlasselemente zurück, die in dem `CulvertSet` enthalten sind.

Diese Funktion ist sinnvoll einsetzbar, wenn das Skript davon ausgehen kann, dass alle 1D-Durchlasselemente dieselbe Höhe haben. Dies ist insbesondere dann der Fall, wenn zuvor `aCulvertSet:setHeight(x)` aufgerufen wurde. Beachten Sie, dass sowohl Höhe als auch Breite bei einem kreisförmigen Durchlass dem Durchmesser entsprechen.

```
aCulvertSet:countElements()
```

Parameter: -

Rückgabe: Number (Integer)

Liefert die Anzahl der 1D-Durchlasselemente zurück, die das `CulvertSet` enthält.

Um sicherzustellen, dass keine falschen oder zu wenige 1D-Durchlasselemente erfasst wurden, kann diese Funktion beispielsweise genutzt werden, um die erwartete Anzahl der Elemente zu prüfen:

```
local culvset = hydroas.CulvertSet.new("^MyCulv_%d+$")
assert(culvset:countElements() == 8)
```

```
aCulvertSet:forallCulverts(func)
```

Parameter: Lua-Funktion

Rückgabe: -

Die Lua-Funktion `func` wird für alle Durchlässe des `CulvertSet`s durchgeführt.

Beispiel: Sie möchten die Durchlasshöhe für alle Durchlässe im `CulvertSet` um 5 cm erhöhen. Dabei sind die Höhen nicht identisch, sondern für jeden Durchlass individuell eingestellt. Es gibt also keine gemeinsamen Höhen, auf die die Durchlässe eingestellt werden können. Schreiben Sie dazu eine (lokale) Lua-Funktion, die die Höhe für einen einzelnen Durchlass 5 cm höher setzt. Rufen Sie dann `forallCulverts` mit dieser Funktion auf.

```
local culvset = hydroas.CulvertSet.new("^MyCulv_%d+$")
local function increasHeight(aCulvert)
    local oldH = aCulvert:height()
    aCulvert:setHeight(oldH + 0.05)
end
culvset:forallCulverts(increasHeight)
```

```
aCulvertSet:fullWidth()
```

Parameter: -

Rückgabe: Number (Float)

Liefert die Breite aller enthaltenen Durchlasselemente zurück. Die Gesamtbreite eines `CulvertSet` ist die Summe aller Breiten der enthaltenen Durchlasselemente. Beachten Sie, dass sowohl Höhe als auch Breite bei einem kreisförmigen Durchlass dem Durchmesser entsprechen.

`aCulvertSet:namePattern()`

Parameter: -

Rückgabe: regEx - Muster für Namen

Liefert das Namens-Schema mit dem die Nodestrings gefunden wurden.

`aCulvertSet:q()`

Parameter: -

Rückgabe: Number (Float)

Liefert den Abfluss über den Gesamtdurchlass im zurückliegenden Zeitschritt.

Dazu werden die Abflüsse aller enthaltenen Durchlasselemente aufaddiert.

`aCulvertSet:qMax()`

Parameter: -

Rückgabe: Number (Float)

Liefert den Maximalabfluss über den Gesamtdurchlass im zurückliegenden Zeitschritt.

Dazu werden die Maximalabflüsse aller enthaltenen Durchlasselemente aufaddiert.

`CaCulvertSet:setCoefficient(coeff)`

Parameter:

coeff: Number (Float) - Abflussbeiwert

Rückgabe: -

Setzt für alle 1D-Durchlasselemente im `CulvertSet` den Abflusskoeffizienten auf den vorgegebenen Wert.

`aCulvertSet:setFullWidth(w)`

Parameter:

w: Number (Float) - Breite

Rückgabe: -

Benutzen Sie diese Funktion, wenn die Einzelbreiten der 1D-Durchlässe ungefähr korrekt sind, Sie aber sicherstellen wollen, dass die Gesamtbreite – unabhängig von der Korrektheit der exakten Lage der 2D-Elemente – einer Vermessung entspricht.

Wenn das `CulvertSet` aus n Elementen besteht, und jeder Einzeldurchlass die Breite

w_i^{alt} hat, wird diese Breite nach folgender Formel korrigiert:

$$w_i^{neu} = w_i^{alt} \frac{w}{\sum_k w_k^{alt}}$$

Beachten Sie, dass sowohl Höhe als auch Breite bei einem kreisförmigen Durchlass dem Durchmesser entsprechen. Daher werden durch den Aufruf dieser Funktion für kreisförmige Durchlässe beide Werte neu gesetzt.

```
aCulvertSet:setHeight(h)
```

Parameter:

h: `Number (Float)` - Höhe

Rückgabe: -

Setzt für alle 1D-Durchlasselemente im `CulvertSet` die Höhe auf den vorgegebenen Wert. Beachten Sie, dass sowohl Höhe als auch Breite bei einem kreisförmigen Durchlass dem Durchmesser entsprechen. Daher werden durch den Aufruf dieser Funktion für kreisförmige Durchlässe beide Werte neu gesetzt.

```
aCulvertSet:setQMax(qmax)
```

Parameter:

qmax: `Number (Float)` - neuer Maximalabfluss

Rückgabe: -

Setzt für alle 1D-Durchlasselemente im `CulvertSet` den Maximalabfluss, so dass der Gesamtmaximalabfluss auf dem vorgegebenen Wert entspricht. Dabei werden die einzelnen Maximalabflüsse mit dem Anteil, den der jeweilige Durchlass zur Gesamtbreite beiträgt, gewichtet.

8.6.3 Klasse `hydroas.Element`

Diese Klasse erlaubt den Zugriff auf Elemente.

8.6.3.1 Statische Klassenfunktionen

```
Element.count()
```

Parameter: -

Rückgabe: `Number (Integer)`

Anzahl der Elemente im Modell.

```
Element.new(idx)
```

Parameter:

idx: `Number (Integer)` - Nummer des Elements.

Rückgabe: `Element`-Instanz

8.6.3.2 Objektmethoden

`aElement.id()`

Parameter: -

Rückgabe: Number (Integer)

Nummer des Elements.

`aElement.nuconst()`

Parameter: -

Rückgabe: Number (Float)

Liefert den konstanten Anteil der Viskosität ν_0 , siehe [Kapitel 2.1.2](#).

`aElement.setNuconst(nu)`

Parameter:

nu: Number (Float) - Wert

Rückgabe: -

Setzt den konstanten Anteil der Viskosität ν_0 auf den vorgegebenen Wert `nu`, siehe [Kapitel 2.1.2](#).

8.6.4 Klasse `hydroas.Global`

Diese Klasse erlaubt den Zugriff auf einige globale Daten im System. Von dieser Klasse werden keine Instanzen angelegt.

8.6.4.1 Statische Klassenfunktionen

`Global.cfl()`

Parameter: -

Rückgabe: Number (Float)

Faktor für die CFL-Bedingung.

`Global.cmuvisc()`

Parameter: -

Rückgabe: Number (Float)

Faktor für die Wirbelviskosität. Siehe [Kapitel 2.1.2](#).

`Global.hMin()`

Parameter: -

Rückgabe: Number (Float)

H_{min} gibt den minimalen Wasserstand an, ab dem ein Knoten als nass definiert wird. Wasser kann nur aus nassen Knoten abfließen.

Knoten mit einem Wasserstand kleiner `Global.hMin()` werden wie trockene Knoten behandelt.

`Global.maxLenConstructionName()`

Parameter: -

Rückgabe: Number (Integer)

Maximale Zeichenlänge für Bauwerke.

`Global.maxLenGaugeName()`

Parameter: -

Rückgabe: Number (Integer)

Maximale Zeichenlänge für Pegelpunkte.

`Global.maxLenNodestringName()`

Parameter: -

Rückgabe: Number (Integer)

Maximale Zeichenlänge für Nodestrings.

`Global.qstrgTimestep()`

Parameter: -

Rückgabe: Number (Float)

Wert des Q_Strg-Ausgabezeitschritts.

`Global.setCfl(cfl)`

Parameter:

cfl: Number (Float) - Wert für die CFL-Zahl

Rückgabe: -

Setzt den Faktor für die CFL-Bedingung. Die Veränderung dieses Wertes kann sich auf die numerische Stabilität bzw. auf die Rechengeschwindigkeit auswirken.

Sie können diese Funktion im Scripting beispielsweise nutzen, um unterschiedliche Phasen der Simulation mit verschiedenen CFL-Werten zu berechnen und damit Genauigkeiten und Rechengeschwindigkeiten zu beeinflussen.

Die Auswirkung der Manipulation dieses Wertes in Skripten muss sorgfältig beobachtet werden.

`Global.setCmuvisc(cmu)`

Parameter:

`cmu: Number (Float)` - Wert für die Skalierung der Wirbelviskosität

Rückgabe: -

Setzt den Faktor für die Wirbelviskosität. Siehe [Kapitel 2.1.2](#).

`Global.setQstrgTimestep(timestep)`

Parameter:

`timestep Number (Float)` - Wert für den Q_Strg-Ausgabezeitschritt

Rückgabe: -

Setzt den Q_Strg-Ausgabezeitschritt auf den vorgegebenen Wert.

`Global.setSimEnd(simEnd)`

Parameter:

`simEnd: Number (Float)` - Ende der Simulationszeit in Sekunden

Rückgabe: -

Setzt das Ende der Simulationszeit in Sekunden.

Sie können diese Funktion insbesondere auch dann nutzen, wenn Sie die Simulation beenden wollen. Sie könnten beispielsweise überprüfen, ob an einer bestimmten Stelle in den letzten 5 Simulationsminuten keine Veränderungen mehr auftraten, und aufgrund dieser Bedingung die Simulation beenden:

```
local finish = false
... Insert code which determines
  whether the simulation shall finish ...
if finish then
  local t = hydroas.Global.timestepStart()
  hydroas.Global.setSimEnd(t)
end
```

Diese Sequenz beendet die Simulation kontrolliert: Der aktuelle Zeitschritt wird durchgerechnet, alle Ergebnisse werden rausgeschrieben und der close-Callback wird aufgerufen.

Im Gegensatz dazu kann ein unmittelbarer Abbruch mit Fehlermeldung beispielsweise mittels Lua error-Funktion erzwungen werden:

```
local finish = false
... Insert code which determines
  whether the simulation shall finish ...
if finish then
```

```
error("Cancel simulation in panic! Error xy");  
end
```

```
Global.setSmsTimestep(timestep)
```

Parameter:

timestep: Number (Float) - Wert für den SMS-Ausgabezeitschritt

Rückgabe: -

Setzt den SMS-Ausgabezeitschritt auf den vorgegebenen Wert.

```
Global.simEnd()
```

Parameter: -

Rückgabe: Number (Float)

Ende der Simulationszeit in Sekunden.

Wenn eine Simulation fortgesetzt wird, beinhaltet diese Zeit die simulierte Zeit des vorangegangenen Rechenlaufs. Daher kann diese Zeit von der Simulationsdauer des konkreten Rechenlaufes abweichen.

Beispiel: Ein Simulation liest Anfangsbedingungen zum Zeitpunkt 3.600 Sekunden ein, `Global.simEnd()` ist gleich 10.800, dann werden $10.800-3600=7.200$ Sekunden simuliert.

```
Global.smsTimestep()
```

Parameter: -

Rückgabe: Number (Float)

Wert des SMS-Ausgabezeitschritts.

```
Global.stopSimulation(now)
```

Parameter:

now: Boolean - Flag, ob die Simulation direkt beendet werden soll

Rückgabe: Number (Float)

Beendet die Simulation sofort, wenn `now` auf `true` gesetzt ist. Wenn `now` auf `false` gesetzt ist, wird die Simulation nach dem nächsten SMS-Ausgabezeitschritt beendet. Beachaten Sie, dass `now` automatisch auf `true` gesetzt wird, wenn es nicht angegeben wird. D.h. der Aufruf `hydroas.Global.stopSimulation()` ist identisch zu `hydroas.Global.stopSimulation(now)`

```
Global.timestepIndex()
```

Parameter: -

Rückgabe: Number (Integer)

Nummer des aktuellen Zeitschritts. Bei jedem neuen Zeitschritt wird diese Nummer um 1 heraufgezählt.

Der erste Zeitschritt hat die Nummer 1 (und nicht die Nummer 0).

Auch wenn mit der Option „Anfangsbedingungen einlesen“ gerechnet wird, fängt der `timestepIndex()` mit 1 an.

`Global.timestepLength()`

Parameter: -

Rückgabe: Number (Float)

Länge des aktuellen Zeitschritts in Sekunden.

`Global.timestepStart()`

Parameter: -

Rückgabe: Number (Float)

Beginn des aktuellen Zeitschritts in Sekunden.

Der Aufruf dieser Funktion liefert in den verschiedenen Callbacks unterschiedliche Zeitpunkte. In der `step`-Funktion sind die internen Werte schon berechnet aber noch nicht aktualisiert. In der `writeResult`-Funktion wurden die berechneten Werte übernommen und die Simuliertezeit ist um die interne Zeitschrittweite weitergeschritten. Damit ist der Beginn des aktuellen Zeitschritts in `writeResult` im Vergleich zu `step` um die interne Zeitschrittweite verschoben.

`Global.writeLog(text)`

Parameter:

text: String - gewünschter Text

Rückgabe: -

Schreibt den übergebenen Text in die HYDRO_AS-2D-Datei `hydro_2dm.mel`.

Mit dem „normalen“ `print` von Lua kann auch auf die Standard-Ausgabe geschrieben werden. Diese Ausgaben sind i.Allg. aber flüchtig. Benutzen Sie `Global.writeLog (text)`, um eine spätere Auswertungen des Rechenlaufes zu unterstützen.

`Global.writeResults(sms,qstrg)`

Parameter:

sms: Boolean- - Kennzeichen

qstrg: Boolean- - Kennzeichen

Rückgabe: -

Schreibt zusätzliche Ergebnisdateien. Die Kennzeichnungen geben an, welche Dateien geschrieben werden sollen.

`sms` bezieht sich auf die Ergebnisdateien, die in SMS geladen werden können. Dies sind die Dateien, die Folgendes enthalten Ergebnisse für jeden Mesh-Knoten und werden - ohne Aufruf dieser Funktion - in die SMS geschriebenen Zeitschritte ausgegeben, z.B. Tiefe, Geschwindigkeit.

`qstrg` bezieht sich auf die Ergebnisdateien, die Ergebnisse für Kontrollquerschnitte oder Spurweiten enthalten Punkte und werden - ohne Aufruf dieser Funktion - in die `Q_Strg`-Ausgabe-Zeitschritte geschrieben, z.B. `q_control.dat`, `gauge.dat`.

Wenn die beiden Kennzeichnungen nicht gesetzt sind, werden sie als `true` interpretiert. Das bedeutet, dass die Aufrufe `Global.writeResults()` und `Global.writeResults(true,true)` sind identisch.

```
Global.xMin()
```

Parameter: -

Rückgabe: Number (Float)

Liefert den Wert der westlichsten Koordinate des Modells, i.e. die minimale x-Koordinate.

```
Global.yMin()
```

Parameter: -

Rückgabe: Number (Float)

Liefert den Wert der südlichsten Koordinate des Modells, i.e. die minimale y-Koordinate.

8.6.5 Klasse `hydroas.GrainFraction`

Diese Klasse erlaubt den Zugriff auf Kornfraktionen.

8.6.5.1 Statische Klassenfunktionen

```
GrainFraction.count()
```

Parameter: -

Rückgabe: Number (Integer)

Anzahl der Kornfraktionen im Modell.

```
GrainFraction.new(idx)
```

Parameter:

idx: Number (Integer) - Nummer der Kornfraktion.

Rückgabe: `GrainFraction`-Instanz

8.6.5.2 Objektmethoden

```
aGrainFraction:id()
```

Parameter: -

Rückgabe: Number (Integer)

Nummer der Kornfraktion.

```
aGrainFraction:vSink()
```

Parameter: -

Rückgabe: Number (Float)

Liefert die Sinkgeschwindigkeit zurück.

```
aGrainFraction:setVSink(v)
```

Parameter:

v: Number (Float) - Geschwindigkeit

Rückgabe: -

Setzt die Sinkgeschwindigkeit auf den vorgegebenen Wert v.

8.6.6 Klasse hydroas.InflowBC

Diese Klasse erlaubt den Zugriff auf Zufluss-Randbedingungen.

8.6.6.1 Statische Klassenfunktionen

```
InflowBC.count()
```

Parameter: -

Rückgabe: Number (Integer)

Anzahl der Zufluss-Randbedingungen

```
InflowBC.countInflowOutflow()
```

Parameter: -

Rückgabe: Number (Integer)

Anzahl der Zufluss-Randbedingungen vom Typ *Zulauf* gebunden and *Auslauf* (`hydroas.Types.InflowBC.BoundedToOutflow()`)

```
InflowBC.countInflowSeries()
```

Parameter: -

Rückgabe: Number (Integer)

Anzahl der Zufluss-Randbedingungen vom Typ *Zulauf* mit *Zeitreihe* (`hydroas.Types.InflowBC.QSeries()`)

```
InflowBC.disableAll()
```

Parameter: -

Rückgabe: -

Setzt alle Zuflüsse im System auf 0

```
InflowBC.new(name)
```

Parameter:

name: `String` - Name der Zufluss-Randbedingung in SMS

oder

name: `Number (Integer)` - Nummer der Zufluss-Randbedingung.

Rückgabe: `InflowBC`

Erzeugt eine neue Instanz zum Zugriff auf eine Zufluss-Randbedingung.

Die Nummer der Zufluss-Randbedingung muss eine ganze Zahl zwischen 1 und `InflowBC.count()` sein.

8.6.6.2 Objektmethoden

```
aInflow.appendValue(t,q)
```

Parameter:

t: `Number (Float)` - Zu ergänzender Zeitpunkt der Zufluss-Zeitreihe.

q: `Number (Float)` - Abflusswert zu dem angegebenen Zeitpunkt.

Rückgabe: -

Ergänzt eine neue Stützstelle in die Zufluss-Zeitreihe. HYDRO_AS-2D ergänzt nur neue Werte an das Ende der Zeitreihe. Der neue Zeitpunkt muss also echt größer sein, als der bisher größte Zeitpunkt. Die Verletzung dieser Bedingung löst einen Lua-Fehler aus.

Im Zusammenhang mit `aInflow:reset(q)` können Sie die Reihe auf diese Weise auch vollständig neu setzen.

Die folgende Sequenz erzeugt beispielsweise eine vollständig neue Reihe mit vier Stützstellen:

```
aInflow:reset(0)
aInflow:appendValue(3600,2.7)
aInflow:appendValue(7200,2.7)
aInflow:appendValue(10800,1.8)
```

```
aInflow:disable()
```

Parameter: -

Rückgabe: -

Entfernt alle vorhandenen Zeitreihenstützstellen. Setzt Zufluss auf 0.

```
aInflow:loadSeriesFromXYFile(filename)
```

Parameter:

`filename`: String - Dateiname

Rückgabe: Number (Integer), Number (Float) - Tupel Anzahl verarbeiteten Wertezeilen, X-Wert des letzten Wertepaares

Diese Funktion liest eine Zuflusszeitreihe aus einer Datei ein und weist sie der Zuflussrandbedingung zu.

Die Datei muss das folgende Format haben:

```
XY "Value curve"
0 0.0299095
300 0.029819
600 0.0297265
```

Eigenschaften der Datei im Detail:

- Die erste Zeile wird nicht ausgewertet. Ihr Inhalt kann beliebig sein.
- Jede Wertezeile enthält einen Zeitstempel (in Sekunden) und einen Zuflusswert (in m₃/s).
- Die beiden Werte müssen durch wenigstens ein Leerzeichen oder einen Tabulator getrennt sein.
- Nach der ersten Zeile darf die Datei beliebig viele Leerzeilen beinhalten, die ignoriert werden. Leerzeilen sind Zeilen, die keinen Zeichen außer Leerzeichen und Tabulatoren enthalten.
- Dezimaltrenner für die Zahlen ist der Punkt („.“).
- Die Exponentialschreibweise (1.543E2) wird unterstützt.
- Jede Zeile darf nach einem Leerzeichen nach dem zweiten Wert beliebigen zusätzlichen Text enthalten, der ignoriert wird.
- Die Werte der ersten Spalte müssen aufsteigend angeordnet sein.

Wenn eine Datei nicht formal korrekt ist, wirft die Funktion einen Lua-error mit einer (englischen) Fehlermeldung, dem Dateinamen und einer Beschreibung des Fehlers. Wenn die Datei keine Werte enthält, wird die Zuflussreihe nicht verändert und ebenfalls ein Fehler gemeldet.

Wenn diese Funktion bei Fehlerfällen nicht zum Abbruch führen soll, müssen Sie diesen Fehler mit dem Lua-pcall fangen und verarbeiten.

Im folgenden Beispiel wird das Simulationensende so gesetzt, dass das Modell mit einer Stunde Nachlaufzeit simuliert wird:

```
local cnt,lastx=aInflow:loadSeriesFromXYFile("infile.xy")
hydroas.Global.setSimEnd(lastx+3600)
```

```
aInflow:name()
```

Parameter: -

Rückgabe: String

Liefert den Name, der in SMS angegeben wurde, oder, wenn der Name nicht gesetzt ist, den Default-Namen.

```
aInflow:nodeString()
```

Parameter: -

Rückgabe: `hydroas.NodeString`

Liefert Zugriff auf den der Randbedingung zugrundeliegenden Nodestring.

```
aInflow:q()
```

Parameter: -

Rückgabe: Number (Float)

Liefert den Abfluss aus der Zufluss-Randbedingung im letzten Zeitschritt. Dieser Wert ist identisch zu `aInflow:nodeString():q()`.

```
aInflow:reset(q)
```

Parameter:

q: Number (Float) - Zu setzender Zufluss der Zeitreihe

Rückgabe: -

Setzt die Zufluss-Zeitreihe vollständig neu auf einen einzigen konstanten Wert.

Die Zuflussreihe besteht anschließend aus genau einer Stützstelle mit dem Zeitpunkt 0 und dem vorgegebenem Zuflusswert.

Einsatzbereiche für diesen Befehl:

- Da HYDRO_AS-2D während der Simulation die Reihe extrapoliert, können Sie mit diesem Befehl HYDRO_AS-2D veranlassen mit einer konstanten Reihe zu rechnen.
- Sie können auch in jedem Zeitschritt den Zufluss (beispielsweise aus einer externen Datenquelle) neu ermitteln und setzen. Aus HYDRO_AS-2D-Sicht scheint die Reihe dann konstant.
- Zusammen mit `aInflow:appendValue()` können Sie die Reihe vollständig neu einlesen und setzen.

Die folgende Sequenz erzeugt eine vollständig neue Reihe mit vier Stützstellen:

```
aInflow:reset(1)
aInflow:appendValue(3600,2.7)
aInflow:appendValue(7200,2.7)
aInflow:appendValue(10800,1)
```

Diese Sequenz würden Sie wahrscheinlich nicht in jedem Zeitschritt, sondern beim Simulationsstart ausführen. Die Werte könnten Sie auch mit Lua aus einer Datei einlesen.

```
aInflow:seriesQ(indx)
```

Parameter:

indx: Number (Integer) - Index in der Zufluss-Zeitreihe

Rückgabe: Number (Float)

Liefert den Abfluss (Zufluss) an einer Stützstelle der Zufluss-Zeitreihe zurück.

Der übergebene Index muss eine ganze Zahl zwischen 1 und `aInflow:sizeSeries()` sein.

```
aInflow:seriesTime(indx)
```

Parameter:

indx: Number (Integer) - Index in der Zufluss-Zeitreihe

Rückgabe: Number (Float)

Liefert den Zeitpunkt (in Sekunden) an einer Stützstelle der Zufluss-Zeitreihe zurück.

Der übergebene Index muss eine ganze Zahl zwischen 1 und `aInflow:sizeSeries()` sein.

```
aInflow:setAllQ(q)
```

Parameter:

q: Number (Float) - zu setzender Zufluss der Zeitreihe

Rückgabe: -

Setzt den Zufluss der Zufluss-Zeitreihe auf einen konstanten Wert. Durch diesen Befehl werden alle Werte der Zufluss-Zeitreihe auf den übergebenen Wert gesetzt.

Die Zufluss-Randbedingung ist danach praktisch nicht mehr zeitabhängig.

Insbesondere können Sie diesen Befehl benutzen, wenn Sie in jedem Zeitschritt den Zufluss (beispielsweise aus einer externen Datenquelle) neu ermitteln.

```
aInflow:setSeriesQ(indx,q)
```

Parameter:

indx: Number (Integer) - Index in der Zufluss-Zeitreihe

q: Number (Float) - zu setzender Zufluss der Zeitreihe am angegebenen Index

Rückgabe: -

Verändert die Zufluss-Zeitreihe und setzt den Zufluss an einer Stützstelle auf einen bestimmter Wert.

Der übergebene Index muss eine ganze Zahl zwischen 1 und `aInflow:sizeSeries()` sein.

```
aInflow:sizeSeries()
```

Parameter: -

Rückgabe: Number (Integer) - Anzahl der Stützstellen der Zufluss-Zeitreihe

Falls die Zufluss-Randbedingung keine Zeitreihe („Zulauf gebunden an Auslauf“) hat, ist dieser Wert 0.

```
aInflow:type()
```

Parameter: -

Rückgabe: Number (Integer)

Der Typ der Zufluss-Randbedingung entspricht einem der Typen aus `hydroas.Types.InflowBC`.

8.6.7 Klasse `hydroas.Node`

Diese Klasse erlaubt den Zugriff auf einzelne Knoten.

8.6.7.1 Statische Klassenfunktionen

```
Node.count()
```

Parameter: -

Rückgabe: Number (Integer)

Anzahl der Knoten im Modell

```
Node.nearest(x,y)
```

Parameter:

x: Number (Integer) - absolute x-Koordinate

y: Number (Integer) - absolute y-Koordinate

Rückgabe: `Node`

Erzeugt eine neue Instanz zum Zugriff auf einen Knoten.

Dabei wird der Knoten, der den angegebenen Koordinaten am nächsten liegt zurückgeliefert. Gibt es mehrere Knoten, die minimalen Abstand zu (x,y) haben, wird einer dieser Knoten zurückgeliefert. Welcher Knoten dies ist, ist nicht definiert. Das Ergebnis dieser Funktion kann sich in diesem Fall z.B. durch Umnummerierung des Netzes ändern.

`Node.new(name)`

Parameter:

name: `String` - Name eines Pegelpunktes in SMS

oder

name: `Number (Integer)` - Nummer des Nodes

Rückgabe: `Node`

Erzeugt eine neue Instanz zum Zugriff auf einen Knoten.

Falls der Parameter `name` eine Zahl ist, muss dies eine ganze Zahl zwischen 1 und `Node.count()` sein.

Für den Parameter `name` sollte i. Allg. die Bezeichnung eines Pegelpunktes angegeben werden. Es ist schlechter Stil, die SMS-ID's der Knoten in den Skripten direkt zu verwenden, da diese sich leicht verändern können.

Die Knotennummen können aber aus anderen Funktionen des Scripting abgeleitet werden. Beispielsweise wird so der Zugriff auf den ersten Knoten einer Zufluss- Randbedingung vereinbart

```
local ns = hydroas.NodeString.new("Inflow1")
```

```
local node = hydroas.Node.new(ns:nodeId(1))
```

In diesem Beispiel wird zwar die SMS-Knoten-ID zum Anlegen der `Node`-Instanz genutzt. Diese Knoten-ID wurde aber nicht in das Skript fest einprogrammiert, sondern wird über die Bezeichnung „Inflow1“ angesprochen.

`Node.newWriter(filename, func1, func2)`

Parameter:

filename: `String` - Name der Ausgabedatei

func1: `Function` - Funktion der auszugebenden Werte

func2: `Function` - Optionaler Parameter: Weitere Funktion der auszugebenden Werte

Rückgabe: `NodeWriter`

Erzeugt einen `NodeWriter`, der Zustände von Knoten auf eine mit SMS lesbare Dat- Datei ausgeben kann.

Die Funktionen `func1` und (wenn gesetzt) `func2` müssen als Parameter jeweils ein `Node`-Instanz akzeptieren und als Rückgabewert eine `Number` liefern. Viele der im Folgenden beschriebenen `Node`-Methoden erfüllen diese Bedingung und sind einsetzbar. Beispielsweise könnten Sie folgende Funktionen übergeben: `Node.constructionLowerEdge`, `Node.z`, `Node.h`, `Node.v`, `Node.vx`, ...

Sie können aber auch eigene Funktionen implementieren und an den `NodeWriter` übergeben.

Anwendungsbeispiele:

Nach einem Dambruch können Sie die Knotenhöhen rausschreiben, um in SMS die veränderten Höhen zu visualisieren und zu kontrollieren. In diesem Fall wählen Sie als Funktion einfach die in Node angebotene z-Methode:

```
local zWriter =
  hydroas.Node.newWriter("DamBreakZ.dat", hydroas.Node.z)
local damm=hydroas.NodeSet.new("Dam")
...
return {
  step=function()
    if dambreakCondition() then
      -- Lower all nodes of the dam
      damm.setZ(65.7)
      -- write node-heights after dambreak
      zWriter.write()
    end
  end
}
```

Sie können auch eigene Funktionen entwickeln. Beispielsweise könnten Sie für alle Knoten die Druckhöhe ausgeben:

```
function pressureAltitude(aNode)
  return math.max(aNode:w()-aNode:constructionLowerEdge(),0)
end
local pressAltWriter =
  hydroas.Node.newWriter("PressAlt.dat",pressureAltitude)
return {
  writeResult=function()
    pressAltWriter.write()
  end
}
```

8.6.7.2 Objektmethoden

```
aNode:constantSource()
```

Parameter: -

Rückgabe: Number (Float)

Liefert den Wert des konstanten Quellterms in m/s für den Knoten zurück.

Der konstante Quellterm entspricht, dem in der Datei feflow-in.dat definierten oder dem durch `aNode:setConstantSource(source)` gesetzten Wert in m/s.

```
aNode:constructionLowerEdge()
```

Parameter: -

Rückgabe: Number (Float)

Liefert den Wert einer eventuellen KUK-Randbedingung zurück.

Intern haben im HYDRO_AS-2D-Rechenkern alle Knoten eine KUK-Höhe. Dieser Wert ist für Knoten ohne gesetzte KUK-Randbedingung aber so hoch (9999) gewählt, dass er nie limitierend wirkt. Ob eine KUK Randbedingung gesetzt ist oder nicht, können Sie mit

```
if aNode:constructionLowerEdge() > 9998 then
  -- Keine KUK gesetzt
  ...
end
abfragen.
```

```
aNode:countNeighbours()
```

Parameter: -

Rückgabe: Number (Integer)

Liefert die Anzahl der Nachbarknoten zu einem Knoten zurück.

```
aNode:h()
```

Parameter: -

Rückgabe: Number (Float)

Liefert den aktuellen Wert für die Wassertiefe am Knoten zurück. Dieser Wert entspricht im Wesentlichen den Ausgaben in der depth-Datei.

Dieser Befehl liefert im Gegensatz zur Ausgabe der depth-Datei aber die exakten internen Berechnungswerte zurück:

- Diese Werte können größer als die relativ zur Sohle gemessene Konstruktionsunterkante sein. Die von dieser Funktion zurück gelieferte Höhe entspricht dann der Druckhöhe an dem Knoten.
- Die zurückgelieferten Werte können kleiner als 0 sein. Für die HYDRO_AS-2D-Ausgabe werden diese Werte auf 0 gesetzt. Diese „fiktiven“ Werte werden für eine korrekte Wasserbilanz benötigt.

```
aNode:id()
```

Parameter: -

Rückgabe: Number (Integer) - ID/Nummer des Knoten

Diese ID entspricht der SMS-ID.

```
aNode:kst()
```

Parameter: -

Rückgabe: Number (Float)

Liefert die Manning- Strickler-Rauigkeit im Kontrollvolumen des Knotens in $m^{1/3}/s$.

Beachten Sie, dass Sie in der „normalen“ Nutzung von HYDRO_AS-2D den Rauigkeitswert nicht einem Knoten, sondern einem Element zuweisen: HYDRO_AS-2D rechnet die Rauigkeiten der einen Knoten umgebenden Elemente anteilig in die Rauigkeit des Kontrollvolumens an den Knoten um. Diese Funktion liefert diesen gemittelten Wert.

In der Literatur wird die Rauigkeit häufig mit der Inversen $s/m^{1/3}$ angegeben. Beachten die korrekte Einheit. Der hier zurückgelieferte Wert wird als Faktor auf die Fließformel

$$v = k_{st} \cdot R^{\frac{2}{3}} \cdot \sqrt{I}$$

benutzt. (v: Geschwindigkeit; R: Hydraulischer Radius; I: Energieliniengefälle)

`aNode:name()`

Parameter: -

Rückgabe: String

Liefert zu einem Node den Pegelnamen zurück. Falls der Name in dieser Pegel-Randbedingung ungesetzt ist, wird die Knotennummer als String zurückgeliefert. Falls für diesen Node keine Pegel-Randbedingung definiert wurde, wird ein leerer String zurückgeliefert.

`aNode:neighbour(indx)`

Parameter:

indx: Number (Integer) - Index eines Nachbarknotens

Rückgabe: Number (Integer)

Der Index-Parameter muss zwischen 1 und `aNode:countNeighbours()` sein.

Der Rückgabewert ist die ID/Nummer des entsprechenden Nachbarknotens. Mit

```
local ngbNode = Node.new(aNode:neighbour(i))
```

```
local ngbHeight = ngbNode:h()
```

kann eine Node-Instanz zum Nachbarn angelegt werden, um die Eigenschaften des Nachbarn abzufragen.

`aNode:relativeX()`

Parameter: -

Rückgabe: Number (Float)

Liefert den X-Wert der Koordinate des Knotens.

Der HYDRO_AS-2D-Rechenkern arbeitet nur mit relativen Koordinaten. Diese Koordinate ist relativ zu einer absoluten Koordinate im Modellgebiet.

Die Werte für die X-Koordinate sind im Vergleich zu absoluten Koordinaten (mit fünf bis acht Dezimalstellen) klein.

`aNode:relativeY()`

Parameter: -

Rückgabe: Number (Float)

Liefert den Y-Wert der Koordinate des Knotens.

Der HYDRO_AS-2D-Rechenkern arbeitet nur mit relativen Koordinaten. Diese Koordinate ist relativ zu einer absoluten Koordinate im Modellgebiet.

Die Werte für die Y-Koordinate sind im Vergleich zu absoluten Koordinaten (mit fünf bis acht Dezimalstellen) klein.

```
aNode:setConstantSource(source)
```

Parameter:

source: Number (Float) - Quellterm in m/s

Rückgabe: -

Setzt den Wert des konstanten Quellterms in m/s für den Knoten. Dieser Wert wird außerhalb des Scripting über die Datei feflow-in.dat gesetzt.

In der Datei feflow-in.dat/h5 können Quellterme auch als Abflusswerte in m³/Tag angegeben werden. Im Scripting ist nur die Angabe als Spende in der Einheit m/s möglich.

D.h. wenn Sie einen Abfluss als Quellterm setzen wollen, müssen Sie den Wert in eine Spende umrechnen:

```
q_ms = q_m3s/aNode:volume()
aNode:setConstantSource(q_ms)
```

Der hier angegebene Quellterm kann auch negativ sein und entspricht in dem Fall einer Senke.

In HYDRO_AS-2D gibt es neben dem konstanten Quellterm weiter zeitvariante Quellterme, z. B. die Niederschlagszeitreihen.

Diese anderen Quellterme werden durch die Funktion `setConstantSource` nicht verändert.

```
aNode:setConstructionLowerEdge(lowerEdge)
```

Parameter:

lowerEdge: Number (Float) - Höhe der Konstruktionsunterkante

Rückgabe: -

Setzt den Wert einer eventuellen KUK-Randbedingung.

Dieser Wert kann für alle Knoten gesetzt werden - unabhängig davon, ob in SMS für den Knoten eine KUK-Randbedingung vereinbart wurde. Für Knoten, die in der 2dm-Datei keine KUK gesetzt haben, kann eine KUK daher auch noch nachträglich angelegt werden.

Um eine bestehende KUK zu „entfernen“, ist der Wert so groß zu setzen, dass die KUK größer als die größte mögliche Wassertiefe ist:

```
aNode:setConstructionLowerEdge(9999)
```

```
aNode:setH(h)
```

Parameter:

h: Number (Float) - zu setzende Wassertiefe

Rückgabe: -

Setzt die Wassertiefe auf den angegeben Wert.

In der Praxis ist darauf zu achten, dass Änderungen an den Zustandsvariablen (aNode:h, aNode:vx, aNode:vh) die numerische Stabilität der Berechnungen beeinträchtigen können.

Wenn – wie im Falle einer Wasserstands-Randbedingung – diese Werte in jedem Zeitschritt manipuliert werden, ist der Entwickler des Steuerungsskripts selbst für die Wahrung der numerischen Stabilität verantwortlich. Dies kann in der Praxis ggf. schwer umzusetzen sein.

Ferner wird durch die Veränderung der Wassertiefe die Wassermengenbilanz verfälscht.

```
aNode:setKst(kst)
```

Parameter:

kst: Number (Float) - Manning-Strickler-Rauigkeit in $m^{1/3}/s$

Rückgabe: -

Setzt die Manning- Strickler-Rauigkeit für das Kontrollvolumen des Knotens.

Beachten Sie, dass Sie in der „normalen“ Nutzung von HYDRO_AS-2D den Rauigkeitswert nicht einem Knoten, sondern einem Element zuweisen: HYDRO_AS-2D rechnet die Rauigkeiten der einen Knoten umgebenden Elemente anteilig in die Rauigkeit des Kontrollvolumens am Knoten um. Der Wert den Sie hier setzen entspricht diesem gemittelten Wert.

In der Literatur wird die Rauigkeit häufig mit der Inversen $s/m^{1/3}$ angegeben. Beachten Sie die korrekte Einheit. Der hier übergebene Wert wird als Faktor auf die Fließformel benutzt.

$$v = k_{st} \cdot R^{\frac{2}{3}} \cdot \sqrt{I}$$

v: Geschwindigkeit;

R: Hydraulischer Radius;

I: Energieliniengefälle

Diese Funktion ist nicht sinnvoll einsetzbar, wenn Sie mit fließtiefenabhängigen Rauigkeiten rechnen: Am Anfang eines jeden internen Zeitschrittes berechnet HYDRO_AS-2D die tiefenabhängigen Rauigkeiten und überschreibt den mit dieser Funktion gesetzten Wert. Wenn Sie mit tiefenabhängigen Rauigkeiten rechnen, wird die hier gesetzte Rauigkeit daher nicht in der Simulation genutzt.

```
aNode:setVx(v)
```

Parameter:

v: Number (Float) - Geschwindigkeit

Rückgabe: -

Setzt die Geschwindigkeit in Ost-Richtung auf den angegeben Wert.

Der vx-Wert wird von HYDRO_AS-2D in jedem Zeitschritt neu berechnet. Bei Änderungen/Manipulationen an diesem Wert ist der Entwickler des Skripts für die Wahrung der numerischen Stabilität verantwortlich. Dies kann in der Praxis ggf. schwer umzusetzen sein.

Ferner wird durch die Veränderung der Geschwindigkeit die Energiebilanz verfälscht.

```
aNode:setVy(v)
```

Parameter:

v: Number (Float) - Geschwindigkeit

Rückgabe: -

Setzt die Geschwindigkeit in Nord-Richtung auf den angegeben Wert.

Auch hier sind die Hinweise zur numerischen Stabilität (vgl. `aNode:h()`, `aNode:vx()`) und zur Energiebilanz zu beachten.

```
aNode:setZ(v)
```

Parameter:

z: Number (Float) - Höhe

Rückgabe: -

Setzt die absolute Höhe der Gewässersohle am angegebenen Knoten.

Da die Konstruktionsunterkante relativ zu diesem z-Wert angegeben wird, wird durch Änderung der absoluten Höhe der Gewässersohle auch die absolute Höhe der Konstruktionsunterkante verändert. Falls dies nicht gewünscht ist, muss `aNode:setConstructionLowerEdge(kuk)` zusätzlich aufgerufen werden.

Mit der Änderung der Knotenhöhe wird auch das Reibungsgefälle in der Umgebung des Knotens neu bestimmt. Außerdem wird der Wert für den trockenen Wasserspiegel (verg. [Kapitel 5.7.1](#)) ggf. angepasst.

Änderungen an der Höhe der Gewässersohle wirken sich entsprechend auch auf den absoluten Wasserstand (`aNode:w()`) aus.

```
print(aNode:z()) - 100
print(aNode:h()) - 5
print(aNode:w()) - 105
aNode:setZ(102)
print(aNode:h()) - 5
print(aNode:h()) - 5
```

`aNode:source(source)`

Parameter: -

Rückgabe: Number (Float) - Quellterm für den Knoten in m/s

Dieser Wert ist die Summe aller Quellterme am Knoten: Konstanter Quellterm, Werte aus `sources-in.(dat/h5)` und Terme aus Niederschlagsdaten und aus Schachtzeitreihen sind hier addiert.

Da Quellterme aus `sources-in.(dat/h5)`, Niederschlagsdaten und Schachtzeitreihen zeitlich variabel sind, kann sich dieser Wert für jeden Zeitschritt ändern.

Sie können diesen Wert nicht unmittelbar mit dem Scripting verändern. Sie können mit dem Scripting nur den konstanten Quelltermanteil verändern (`aNode:setConstantSource`)

`aNode:v()`

Parameter: -

Rückgabe: Number (Float)

Liefert den aktuellen Wert für den Betrag der Geschwindigkeit. Dies ist die Wurzel der Quadratsumme von v_x und v_y .

`aNode:volume()`

Parameter: -

Rückgabe: Number (Float) - Finite-Kontrollvolumen in m^2 .

Liefert das Finite-Kontrollvolumen um einen Knoten herum zurück. Da HYDRO_AS-2D ein 2D-Modell ist, sind Kontrollvolumen Flächen.

Im Finite-Volumen-Modell HYDRO_AS-2D werden die Kontrollvolumina um die Knoten herum berechnet (`aNode:volumeOrig()`). Falls diese Berechnung einen kleineren Wert als A_{min} zurückliefert, wird das Finite-Kontrollvolumen auf A_{min} erhöht. Es gilt also:

```
aNode:volume() == max(aNode:volumeOrig(), aMin)
```

Benutzen Sie diesen Wert beispielsweise um Wassertiefen in Wasservolumina umzurechnen:

```
vol_m3 = aNode:volume()*aNode:h()
```

oder um Abflüsse für die Angabe von Quelltermen/Niederschlägen in Spenden umzurechnen.

Die Summe aller `vol_m3` an allen Knoten ergibt die im Modell insgesamt gespeicherte Fülle.

`aNode:volumeOrig()`

Parameter: -

Rückgabe: Number (Float) - Basiswert für das Finite-Kontrollvolumen in m^2 .

Liefert die Fläche um einen Knoten herum zurück, die die Basis für das Kontrollvolumen `aNode:volume()` ist. Der von dieser Funktion zurückgelieferte Wert kann aber auch kleiner als A_{min} sein.

Der zurückgelieferte Wert entspricht der 2D-Fläche und berücksichtigt daher kein Gefälle.

`aNode:vx()`

Parameter: -

Rückgabe: Number (Float)

Liefert den aktuellen Wert für die Geschwindigkeit in Ost-Richtung zurück.

`aNode:vy()`

Parameter: -

Rückgabe: Number (Float)

Liefert den aktuellen Wert für die Geschwindigkeit in Nord-Richtung zurück.

`aNode:w()`

Parameter: -

Rückgabe: Number (Float)

Liefert den absoluten Wasserspiegel am Knoten zurück.

Für das Scripting gilt stets

```
aNode:w() == aNode:h() + aNode:z()
```

Damit kann dieser Wert von den Werten für die Ausgabe des Wasserspiegels in der `wspl-`Ausgabedatei abweichen:

- `aNode:w()` nimmt ähnlich wie `aNode:h()` keine Korrekturen vor.
- `aNode:w()` nimmt keine Extrapolation der Wasserspiegellage auf benachbarte Knoten vor.

`aNodeWriter:write()`

Parameter: -

Rückgabe: -

Schreibt in die Datei der `NodeWriter`-Instanz den aktuellen Zustand der Knoten als SMS-dat-Datei raus.

Der aktuelle Zeitschritt von HYDRO_AS-2D wird als TS-Angabe übernommen.

Zur Nutzung der Funktion vergleiche `Node.newWriter`.

Beim ersten Aufruf der `write`-Methode wird die Ausgabedatei immer neu angelegt. Alte Inhalte der Datei werden somit überschrieben.

Bei nachfolgenden Aufrufen der `write`-Methode wird die Ausgabedatei erweitert.

Am einfachsten Nutzen Sie einen `NodeWriter`, indem Sie ihn in der Initialisierungsphase anlegen und danach nur noch die `write`-Methode aufrufen.

`aNode.x()`

Parameter: -

Rückgabe: Number (Float)

Liefert die absolute x-Koordinate des Knotens.

Es gilt

```
aNode.x() == aNode.relativeX() + hydroas.Global.xMin()
```

`aNode.y()`

Parameter: -

Rückgabe: Number (Float)

Liefert die absolute y-Koordinate des Knotens.

Es gilt

```
aNode.y() == aNode.relativeY() + hydroas.Global.yMin()
```

`aNode.z()`

Parameter: -

Rückgabe: Number (Float)

Liefert die absolute Höhe der Gewässersohle an dem angegebenen Knoten.

8.6.8 Klasse `hydroas.NodeSet`

Diese Klasse erlaubt den Zugriff auf eine Menge von Knoten. Benutzen Sie diese Klasse beispielsweise, wenn Sie eine Gruppe von Knoten simultan absenken, anheben oder für diese Knoten einen bestimmten Wert setzen wollen.

Ein Anwendungsfall könnte ein Wehr sein, in dem die Wehrkrone als Knoten abgebildet ist und das über die Veränderung der Z-Werte der Knoten abgesenkt/aufgerichtet wird.

Die Menge der Knoten kann theoretisch aus einer Liste einzelner Knotennummern erzeugt werden.

Ein `NodeSet` kann aus beliebigen Knotenmengen bestehen. Diese Knotenmengen brauchen nicht zusammenhängend zu sein. In der Praxis wird ein `NodeSet` meist aus den Knoten eines oder mehrerer

Nodestrings gebildet.

Beachten Sie, dass in HYDRO_AS-2D nur solche Nodestrings angesprochen werden können, für die eine entsprechende SMS-Randbedingung vereinbart wurde (vgl. [Kapitel 8.6.9](#)).

Ggf. werden Kontrollquerschnitte auch nur zu diesem Zweck vereinbart. Zu diesen Kontrollquerschnitten soll der entsprechende „Kontrollabfluss“ dann gar nicht abgefragt werden, da sie so geformt sein können, dass die Abflussberechnung keine sinnvollen Werte liefern kann.

8.6.8.1 Statische Klassenfunktionen

```
NodeSet.newByNodes(list)
```

Parameter:

Lua-Table mit Knoten-ID's oder Pegelpunktbezeichnungen

Rückgabe: NodeSet

Erzeugt ein NodeSet, das die Knoten in der angegebenen Table (Liste/Array) enthält.

Jedes Element von `list` muss vom Typ Number (Integer) oder String sein.

- Typ Number (Integer): Wert ist eine Node-ID
- Typ String: Wert ist Bezeichnung eines Pegelpunktes. Die Bezeichnung kann auch ein Pattern sein. In diesem Fall werden alle Knoten, deren Bezeichnung auf das Pattern zutreffen, zum NodeSet hinzugefügt.

Grundsätzlich können diese Typen auch in einer Liste gemischt werden. Beispielsweise kann ein NodeSet gebildet werden, das aus den Anfangs- und Endknoten eines Durchlasses und zusätzlich dem Knoten des Pegelpunktes „Gauge1“ besteht:

```
local hydroas = require("hydroas")
local culv = hydroas.Culvert("MyCulvert")
local ns = hydroas.NodeSet.newByNodes(
  {culv:node1Id(), culv:node2Id(), "Gauge1"})
```

```
NodeSet.newByNodeString(ns1,ns2,ns3,...)
```

Parameter:

ns1: String - Name eines NodeStrings

oder

ns1: Number (Integer) - Nummer eines NodeStrings

ns2 (optional): analog zu ns1

...

Rückgabe: NodeSet

Erzeugt ein NodeSet aus einem (oder mehreren Nodestrings).

Der Nodestring wird über den Namen eines Kontrollquerschnittes oder einer Zufluss-/ Abfluss-Randbedingung angegeben. Der Name kann auch als Pattern angegeben werden. In diesem Fall werden alle Nodestrings, deren Namen auf das Pattern zutreffen, berücksichtigt.

Die Nodestrings sind intern von 1 bis `NodeString.count()` durchnummeriert. Es kann auch diese Nummer als Beschreibung für einen Nodestring verwendet werden. Diese Technik wird aber normalerweise nicht benutzt.

Anwendungsbeispiel:

```
local ns = NodeSet.newByNodeString("Wehr")
ns:setZ(100)
```

Wenn das Wehr aus zwei nebeneinanderliegenden Reihen von Knoten besteht, kann der Nodestring auch so gebildet werden, dass er beide Reihen enthält. Er ist in diesem Fall allerdings nicht mehr als sinnvoller Kontrollquerschnitt einsetzbar.

Alternativ könnten die beiden Reihen auch als eigene Kontrollquerschnitte deklariert werden:

```
local ns = NodeSet.newByNodeString ("WehrRow1", "WehrRow2")
ns:setZ(100)
```

8.6.8.2 Objektmethoden

```
aNodeSet:addToZ(delta)
```

Parameter:

delta: Number (Float) - um wieviel soll die Höhe der Knoten im NodeSet verändert werden?

Die Z-Wert aller Knoten im NodeSet werden um den Wert delta erhöht bzw. (bei negativen delta) abgesenkt.

Diese Funktion respektiert dabei die Methode `aNodeSet:freezeLowerZ()`.

Knotenhöhen werden nicht unter den Wert verschoben, den der Knoten hatte als `aNodeSet:freezeLowerZ()` aufgerufen wurde.

```
aNodeSet:anyKst()
```

Parameter: -

Rückgabe: Number (Float)

Liefert den Kst-Wert eines beliebigen Knotens im NodeSet zurück. Es ist nicht festgelegt, welcher Knoten dies ist und in welcher Weise der Wert repräsentativ für das gesamte NodeSet ist.

Diese Methode ist insbesondere dann sinnvoll anwendbar, wenn der Skript-Entwickler weiß, dass alle Knoten im NodeSet denselben Kst-Wert haben, weil die Kst-Werte für die Knoten mit `aNodeSet:setKst(kst)` zuvor gesetzt wurden.

```
aNodeSet:anyZ()
```

Parameter: -

Rückgabe: Number (Float)

Liefert den Z-Wert eines beliebigen Knotens im NodeSet zurück. Es ist nicht festgelegt, welcher Knoten dies ist und in welcher Weise der Wert repräsentativ für das gesamte NodeSet ist.

Diese Methode ist insbesondere dann sinnvoll anwendbar, wenn der Skript-Entwickler weiß, dass alle Knoten im NodeSet dieselbe Höhe haben, weil die Knoten mit `aNodeSet:setZ(z)` zuvor auf dieselbe Höhe gesetzt wurden.

Sie könnten in dieser Situation auch mit `aNodeSet:maxZ()` oder `aNodeSet:minZ()` arbeiten. Diese Methoden müssen jedoch alle Nodes untersuchen und sind daher langsamer.

Beispiel für ein `control.lua`:

```
local hydroas = require("hydroas")
local ns = hydroas.NodeSet.newByNodeString("Weir")
ns:setZ(100)
return {
  step = function()
    -- All nodes have same altitude.
    if ns:anyZ() < 120 then
      ns:setZ(ns:anyZ() + 0.1)
    end
  end
end
}
```

`aNodeSet:freezeLowerZ()`

Parameter: -

Rückgabe: -

Beim Aufruf dieser Methode werden die aktuellen Höhen aller Knoten im NodeSet gespeichert. Spätere Aufrufe von `aNodeSet:setZ(z)` oder `aNodeSet:addToZ(z)` verschieben die Knotenhöhen eines einzelnen Knotens dann nicht mehr unter diese Höhe.

Mit dieser Methode werden daher untere Grenzen für die Knotenhöhen festgelegt.

Anwendungsbeispiel:

Ein Wehr soll gesteuert werden. Die Knoten des Wehres werden mittels eines Kontrollquerschnittes festgelegt. Um den Kontrollquerschnitt sinnvoll abfragen zu können, enthält er an den Grenzen auch die seitlichen – nicht variablen – Widerlager des Wehres.

`aNodeSet:forallNodes(func)`

Parameter: Lua-Funktion

Rückgabe: -

Die Lua-Funktion `func` wird für alle Knoten des NodeSets durchgeführt.

Beispiel: Sie möchten die KUK für alle Knoten im `NodeSet` um 20 cm absenken. Dabei kann der Wert nicht global angepasst werden, weil die KUKs ein Bogenbrücke bilden. Schreiben Sie dazu eine (lokale) Lua-Funktion, die die KUK für einen einzelnen Knoten entsprechend neu setzt. Rufen Sie dann `forallNodes` mit dieser Funktion auf.

```
local function lowerKUK(aNode)
  local oldVal = aNode:constructionLowerEdge()
  aNode:setConstructionLowerEdge(oldVal - 0.2)
end
aNodeSet:forallNodes(lowerKUK)
```

```
aNodeSet:maxConstantSource()
```

Parameter: -

Rückgabe: Number (Float)

Liefert den maximalen konstanten Quellterm aller Knoten im NodeSet zurück.

```
aNodeSet:maxConstructionLowerEdge()
```

Parameter: -

Rückgabe: Number (Float)

Liefert die maximale KUK aller Knoten im NodeSet zurück.

```
aNodeSet:maxH()
```

Parameter: -

Rückgabe: Number (Float)

Liefert die maximale Wassertiefe aller Knoten im NodeSet zurück.

```
aNodeSet:maxKst()
```

Parameter: -

Rückgabe: Number (Float)

Liefert den maximalen Rauheitsbeiwert aller Knoten im NodeSet zurück.

```
aNodeSet:maxSource()
```

Parameter: -

Rückgabe: Number (Float)

Liefert den maximalen Wert der Summe aller Quellterm aller Knoten im NodeSet zurück.

```
aNodeSet:maxW()
```

Parameter: -

Rückgabe: Number (Float)

Liefert den maximalen Wasserspiegel aller Knoten im NodeSet zurück.

`aNodeSet:maxZ()`

Parameter: -

Rückgabe: Number (Float)

Liefert den maximalen Z-Wert aller Knoten im NodeSet zurück.

`aNodeSet:minConstantSource()`

Parameter: -

Rückgabe: Number (Float)

Liefert den minimalen konstanten Quellterm aller Knoten im NodeSet zurück.

`aNodeSet:minConstructionLowerEdge()`

Parameter: -

Rückgabe: Number (Float)

Liefert die minimale KUK aller Knoten im NodeSet zurück.

`aNodeSet:minH()`

Parameter: -

Rückgabe: Number (Float)

Liefert die minimale Wassertiefe aller Knoten im NodeSet zurück.

`aNodeSet:minKst()`

Parameter: -

Rückgabe: Number (Float)

Liefert den minimalen Rauheitsbeiwert aller Knoten im NodeSet zurück.

`aNodeSet:minSource()`

Parameter: -

Rückgabe: Number (Float)

Liefert den minimalen Wert der Summe aller Quellterm aller Knoten im NodeSet zurück.

`aNodeSet:minW()`

Parameter: -

Rückgabe: Number (Float)

Liefert den minimalen Wasserspiegel aller Knoten im NodeSet zurück.

`aNodeSet:minZ()`

Parameter: -

Rückgabe: Number (Float)

Liefert den minimalen Z-Wert aller Knoten im NodeSet zurück.

```
aNodeSet:node(i)
```

Parameter:

i: Number (Integer) - Index eines Knotens im NodeSet

Rückgabe: Node

Liefert eine Node-Instanz zurück. Damit kann direkt auf den i-ten Knoten des NodeSets zugegriffen werden.

Hinweis: Intern wird hier die Node-Instanz mit `Node.new` angelegt.

Diese Funktion ist eine Kurzschreibweise für
`hydroas.Node.new(aNodeSet:nodeId(indx))`

```
aNodeSet:nodeId(i)
```

Parameter:

i: Number (Integer) - Index eines Knotens im NodeSet

Rückgabe: Number (Integer)

Liefert die Nummer/ID eines Knotens im NodeSet zurück.

Beispiel:

```
local hydroas = require("hydroas")
local ns = hydroas.NodeSet.newByNodeString("Weir")
for indx=1,ns:size() do
  print("'Weir' contains node ".. ns:nodeId(indx))
end
```

```
aNodeSet:setConstructionLowerEdge(kuk)
```

Parameter:

kuk: Number (Float) - Wert für die Konstruktionsunterkante (absolute Höhe in m)

Rückgabe: -

Setzt für alle Knoten in aNodeSet die KUK auf den angegebenen Wert.

```
aNodeSet:setKst(kst)
```

Parameter:

kst: Number (Float) - Neuer kst-Wert der Knoten im NodeSet

Rückgabe: -

Setzt den Kst-Wert aller Knoten im NodeSet auf diesen Wert.

aNodeSet:setZ(z)

Parameter:

z: Number (Float) - Neue Höhe der Knoten im NodeSet

Rückgabe: -

Setzt den Z-Wert aller Knoten im NodeSet auf diesen Wert.

Diese Funktion respektiert dabei die Methode `aNodeSet:freezeLowerZ()`: Knotenhöhen werden nicht unter den Wert verschoben, den der Knoten hatte als `aNodeSet:freezeLowerZ()` aufgerufen wurde.

aNodeSet:size()

Parameter: -

Rückgabe: Number (Integer)

Liefert die Anzahl der Knoten im NodeSet

aNodeSet:unfreezeLowerZ()

Parameter: -

Rückgabe: -

Hebt eine mit `aNodeSet:freezeLowerZ()` festgelegte untere Knotenhöhe wieder auf.

Nach dem Aufruf dieser Methode können Knoten wieder beliebig tief abgesenkt werden.

8.6.9 Klasse `hydroas.NodeString`

Die Klasse `NodeString` unterstützt den Zugriff auf die Nodestrings zu

- Kontrollquerschnitten
- Zufluss-Randbedingungen
- Abfluss-Randbedingungen
- 2D-Wehre
- Wasserstands-Randbedingungen

SMS-Nodestrings, zu denen keine dieser Randbedingungen vereinbart wurden, sind im HYDRO_AS-2D-Scripting nicht abrufbar.

8.6.9.1 Statische Klassenfunktionen

`NodeString.count()`

Parameter: -

Rückgabe: Number (Integer)

Anzahl der Nodestrings: Diese enthält alle Nodestrings zu den oben genannten Randbedingungen.

`NodeString.new(id)`

Parameter:

id: String - Name eines Nodestrings

oder

id: Number (Integer) - Nummer eines Nodestrings

Rückgabe: NodeString

`id` kann eine Bezeichnung einer der oben genannten Randbedingungen sein oder eine Zahl (i. e. Index) zwischen 1 und `NodeString.count()`.

Bei der Modellierung in SMS soll darauf geachtet werden, dass diese Bezeichnungen eindeutig sind. Falls die Bezeichnung nicht eindeutig sein sollte, ist nicht festgelegt, welcher der möglichen Nodestrings tatsächlich zurückgegeben wird. Dies könnte sich beispielsweise allein dadurch verändern, dass eine Datei mit SMS neu abgespeichert wird.

8.6.9.2 Objektmethoden

`aNodeString.id()`

Parameter: -

Rückgabe: Number (Integer)

Liefert die ID des Nodestrings.

`aNodeString.name()`

Parameter: -

Rückgabe: String

Liefert den Namen des Nodestrings, sofern dieser gesetzt ist, sonst die Nummer des ersten Knotens des Nodestrings als String.

`aNodeString.nodeId(i)`

Parameter:

i: Number (Integer) - Nummer des Knotens im Nodestring

Rückgabe: Number (Integer)

Diese Funktion liefert die ID des i-ten Knotens im Nodestring zurück.

i muss zwischen 1 und aNodeString.size() liegen. Die zurückgelieferte Knoten-ID können Sie beispielsweise nutzen um eine Node-Instanz anzulegen:

```
local ns = hydroas.NodeString.new("Inflow1")
local node = hydroas.Node.new(ns:nodeId(1))
```

aNodeString:q()

Parameter: -

Rückgabe: Number (Float)

Diese Funktion liefert den Abfluss durch den NodeString im letzten Zeitschritt zurück.

Für Zufluss- und Abfluss-Randbedingungen wird dieser Abfluss mit den entsprechenden Verfahren der Randbedingungen gebildet.

aNodeString:size()

Parameter: -

Rückgabe: Number (Integer)

Liefert die Anzahl der Knoten im NodeString

8.6.10 Klasse hydroas.OutflowBC

Diese Klasse erlaubt den Zugriff auf Abfluss-Randbedingungen.

8.6.10.1 Statische Klassenfunktionen

OutflowBC.count()

Parameter: -

Rückgabe: Number (Integer)

Anzahl der Abfluss-Randbedingungen

OutflowBC.new(name)

Parameter:

name: Number (Integer) - Name der Abfluss-Randbedingung in SMS

oder

name: Number (Integer) - Nummer der Abfluss-Randbedingung.

Rückgabe: OutflowBC

Erzeugt eine neue Instanz zum Zugriff auf eine Abfluss-Randbedingung.

Die Nummer der Abfluss-Randbedingung muss eine ganze Zahl zwischen 1 und `OutflowBC.count()` sein. Achtung diese Nummer ist nicht die SMS-ID des Nodestrings!

8.6.10.2 Objektmethoden

`aOutflow.appendValue(x,y)`

Parameter:

x: Number (Float) - zu ergänzender x-Wert der Reihe.

y: Number (Float) - zu ergänzender y-Wert der Reihe.

Rückgabe: -

Ergänzt eine neue Stützstelle in der Reihe der Auslaufrandbedingung.

`aOutflow.getIe()`

Parameter: -

Rückgabe: Number (Float)

Liefert den Wert des angegebenen Energieliniengefälles als dimensionslose, normierte Zahl, wenn es sich um eine Auslaufrandbedingung vom Typ `Types.OutflowBC.Ie()` handelt. Ein Rückgabewert von 0.001 bedeutet also ein Gefälle von 1Promille.

`aOutflow.loadSeriesFromXYFile(filename)`

Parameter:

filename: String - Dateiname

Rückgabe: Number (Integer), Number (Float) - Tupel Anzahl verarbeiteten Wertezellen, X-Wert des letzten Wertepaares

Diese Funktion liest eine Reihe von Datei ein und weist sie der Abflussrandbedingung zu.

`aOutflow.mu()`

Parameter: -

Rückgabe: Number (Float)

Liefert den Abflussbeiwert für Wehr-Randbedingungen. D.h. der Typ muss `Types.OutflowBC.WQ_GaugePoint()`, `Types.OutflowBC.WeirControl_States()` oder `Types.OutflowBC.WeirControl_WSeries()` sein.

`aOutflow:name()`

Parameter: -

Rückgabe: String

Liefert den Namen, der in SMS angegeben wurde, oder, wenn der Name nicht gesetzt ist, den die Nummer des ersten Knotens des Nodestrings als String.

`aOutflow:nodeString()`

Parameter: -

Rückgabe: `hydroas.NodeString`

Liefert Zugriff auf den der Randbedingung zugrundeliegenden Nodestring.

`aOutflow:q()`

Parameter: -

Rückgabe: Number (Float)

Liefert den Abfluss aus der Abfluss-Randbedingung im letzten Zeitschritt. Dieser Wert ist identisch zu `aOutflow:nodeString():q()`.

`aOutflow:qmax()`

Parameter: -

Rückgabe: Number (Float)

Liefert den maximal möglichen Abfluss aus der Abfluss-Randbedingung im vorherigen Zeitschritt.

`aOutflow:relax()`

Parameter: -

Rückgabe: Number (Float)

Liefert den Wert des Relaxationsparameters für die Auslaufrandbedingung. Dieser Parameter dämpft starke Änderungen im Abfluss über den Auslaufquerschnitt. Je größer der Wert ist, desto stärker wird der Abfluss gedämpft.

`aOutflow:reset(x,y)`

Parameter:

x: Number (Float) - zu setzender x-Wert der Reihe

y: Number (Float) - zu setzender y-Wert der Reihe

Rückgabe: -

Setzt die Reihe vollständig neu auf einen einzigen Stützstelle (x,y). Beachten Sie auch, dass bei den W/Q-Beziehungen der x-Wert der Reihe den Wasserstand W und der y- Wert der Reihe den Abfluss Q repräsentiert. Dies gilt für die Randbedingungen vom Typ `Types.OutflowBC.WQ()` und `Types.OutflowBC.WQ_GaugePoint()`.

```
aOutflow:sizeSeries()
```

Parameter: -

Rückgabe: Number (Integer) - Anzahl der Stützstellen der Reihe

Falls die Randbedingung keine Reihe hat (z.B. `Types.OutflowBC.Ie()`), ist dieser Wert 0.

```
aOutflow:seriesX(i)
```

Parameter:

i: Number (Integer) - Index einer Stützstelle der Reihe

Rückgabe: Number (Float)

Liefert den x-Wert der i-ten Stützstelle der Reihe zurück.

Der übergebene Index muss eine ganze Zahl zwischen 1 und `aOutflow:sizeSeries()` sein. Beachten Sie auch, dass bei den W/Q-Beziehungen der x-Wert der Reihe den Wasserstand W und der y-Wert der Reihe den Abfluss Q repräsentiert. Dies gilt für die Randbedingungen vom Typ `Types.OutflowBC.WQ()` und `Types.OutflowBC.WQ_GaugePoint()`.

```
aOutflow:seriesY(i)
```

Parameter:

i: Number (Integer) - Index einer Stützstelle der Reihe

Rückgabe: Number (Float)

Liefert den y-Wert der i-ten Stützstelle der Reihe zurück.

Der übergebene Index muss eine ganze Zahl zwischen 1 und `aOutflow:sizeSeries()` sein. Beachten Sie auch, dass bei den W/Q-Beziehungen der x-Wert der Reihe den Wasserstand W und der y-Wert der Reihe den Abfluss Q repräsentiert. Dies gilt für die Randbedingungen vom Typ `Types.OutflowBC.WQ()` und `Types.OutflowBC.WQ_GaugePoint()`.

```
aOutflow:setIe(ie)
```

Parameter:

ie: Number (Float) - zu setzendes Energieliniengefälle

Rückgabe: -

Setzt das Energieliniengefälle auf den vorgegebenen Wert, falls der Auslauf vom Typ `Types.OutflowBC.Ie()` ist. Beachten Sie, dass der Wert `ie` als dimensionsloses, normiertes Gefälle angegeben wird. D.h. ein Gefälle von 1 Promille wird durch `setIe(0.001)` gesetzt.

```
aOutflow:setMu(mu)
```

Parameter:

mu: Number (Float) - zu setzender Abflussbeiwert

Rückgabe: -

Setzt den Abflussbeiwert für Wehr-Randbedingungen. D.h. der Typ muss `Types.OutflowBC.WQ_GaugePoint()`, `Types.OutflowBC.WeirControl_States()` oder `Types.OutflowBC.WeirControl_WSeries()` sein.

```
aOutflow:setRelax(relax)
```

Parameter:

relax: Number (Float) - zu setzender Relaxationswert

Rückgabe: -

Setzt den Wert des Relaxationsparameters auf den vorgegebenen Wert. Der kleinst mögliche Wert ist 1. Dieser Parameter dämpft starke Änderungen im Abfluss über den Auslaufquerschnitt. Je größer der Wert ist, desto stärker wird der Abfluss gedämpft. Bei sehr kleinen Werten können Oszillationen auftreten.

```
aOutflow:setSeriesX(i,x)
```

Parameter:

i: Number (Integer) - Index in der Reihe

x: Number (Float) - zu setzender x-Wert der Reihe am angegebenen Index

Rückgabe: -

Verändert die Reihe und setzt den x-Wert an einer Stützstelle auf den angegebenen Wert.

Der übergebene Index muss eine ganze Zahl zwischen 1 und `aOutflow:sizeSeries()` sein. Beachten Sie auch, dass bei den W/Q-Beziehungen der x-Wert der Reihe den Wasserstand `W` und der y-Wert der Reihe den Abfluss `Q` repräsentiert. Dies gilt für die Randbedingungen vom Typ `Types.OutflowBC.WQ()` und `Types.OutflowBC.WQ_GaugePoint()`.

```
aOutflow:setSeriesY(i,y)
```

Parameter:

i: Number (Integer) - Index in der Reihe

x: Number (Float) - zu setzender y-Wert der Reihe am angegebenen Index

Rückgabe: -

Verändert die Reihe und setzt den y-Wert an einer Stützstelle auf den angegebenen Wert.

Der übergebene Index muss eine ganze Zahl zwischen 1 und `aOutflow:sizeSeries()` sein. Beachten Sie auch, dass bei den W/Q-Beziehungen der x-Wert der Reihe den Wasserstand W und der y-Wert der Reihe den Abfluss Q repräsentiert. Dies gilt für die Randbedingungen vom Typ `Types.OutflowBC.WQ()` und `Types.OutflowBC.WQ_GaugePoint()`.

```
aOutflow:setUpstreamNode(aNode)
```

Parameter:

aNode: Node-Instanz - zu setzender Oberwasserpegel als Node-Instanz

Rückgabe: -

Verändert die Reihe und setzt den y-Wert an einer Stützstelle auf den angegebenen Wert.

Setzt den Oberwasserpegelpunkt für eine Wehr-Randbedingung auf `aNode`. D.h. der Typ muss `Types.OutflowBC.WQ_GaugePoint()`, `Types.OutflowBC.WeirControl_States()` oder `Types.OutflowBC.WeirControl_WSeries()` sein.

```
aOutflow:setUpstreamNodeID(index)
```

Parameter:

index: Number (Integer)

Rückgabe: -

Verändert die Reihe und setzt den y-Wert an einer Stützstelle auf den angegebenen Wert.

Setzt den Oberwasserpegelpunkt für eine Wehr-Randbedingung über die vorgegebenen ID `index` neu. D.h. der Typ muss `Types.OutflowBC.WQ_GaugePoint()`, `Types.OutflowBC.WeirControl_States()` oder `Types.OutflowBC.WeirControl_WSeries()` sein.

```
aOutflow:setWeirvelocity(veloc)
```

Parameter:

i: Number (Integer) - zu setzende Geschwindigkeit

Rückgabe: -

Setzt die Wehrgeschwindigkeit auf den vorgegebenen Wert, falls der Auslauf eine Wehr-Randbedingung ist. D.h. der Typ muss `Types.OutflowBC.WQ_GaugePoint()`, `Types.OutflowBC.WeirControl_States()` oder `Types.OutflowBC.WeirControl_WSeries()` sein.

`aOutflow:type()`

Parameter: -

Rückgabe: Number (Integer)

Der Typ der Abfluss-Randbedingung entspricht einem der Typen aus `hydroas.Types.OutflowBC`.

`aOutflow:upstreamNode()`

Parameter: -

Rückgabe: Number (Integer) - Oberwasserpegelpunkt als Node-Instanz

Liefert den Oberwasserpegelpunkt, der in der Wehr-Randbedingung referenziert ist, als Node-Instanz. D.h. der Typ muss `Types.OutflowBC.WQ_GaugePoint()`, `Types.OutflowBC.WeirControl_States()` oder `Types.OutflowBC.WeirControl_WSeries()` sein.

`aOutflow:upstreamNodeId()`

Parameter: -

Rückgabe: Number (Integer) - ID des Oberwasserpegelpunktes

Liefert die ID des Oberwasserpegelpunktes, der in der Wehr-Randbedingung referenziert ist. D.h. der Typ muss `Types.OutflowBC.WQ_GaugePoint()`, `Types.OutflowBC.WeirControl_States()` oder `Types.OutflowBC.WeirControl_WSeries()` sein.

`aOutflow:weirvelocity()`

Parameter: -

Rückgabe: Number (Float) - Wehrgeschwindigkeit

Liefert die Wehrgeschwindigkeit einer Wehr-Randbedingung.

D.h. der Typ muss `Types.OutflowBC.WQ_GaugePoint()`, `Types.OutflowBC.WeirControl_States()` oder `Types.OutflowBC.WeirControl_WSeries()` sein.

8.6.11 Klasse `hydroas.Types`

Dieses Modul enthält nur statische Funktionen, die Konstanten entsprechen. Einsatz erfolgt im Zusammenhang mit den types der einzelnen Randbedingungsklassen. Beispiel:

```
local Culvert = require("hydroas.Culvert")
local Types = require("hydroas.Types")
local culvert = Culvert.new("myCulvert")
if culvert:type() == Types.Culvert.Circle() then
```

```
... use Circle ...
end
```

Diese Funktionen liefern Zahlencodes für die einzelnen Typen zurück. Theoretisch könnte dieser Zahlencode auch direkt im Programm verwendet werden. (`if culvert:type() == 4 then...`). Sie sollten dies aber auf jeden Fall vermeiden, da diese Werte in künftigen Versionen anders sein könnten.

8.6.11.1 Statische Klassenfunktionen

Rückgabewerte von `Culvert:type()`

```
Types.Culvert.Circle()
```

Kreisförmiger Durchlass

```
Types.Culvert.Rectangle()
```

Rechteckiger Durchlass

Rückgabewerte von `aInflowBC:type()`

```
Types.InflowBC.QSeries()
```

Zufluss-Randbedingung mit Zufluss-Zeitreihe

```
Types.InflowBC.BoundedToOutflow()
```

Zufluss-Randbedingung, deren Zufluss aus einer „gebundenen“ Abfluss-Randbedingung übernommen wird.

Rückgabewerte von `aOutflowBC:type()`

```
Types.OutflowBC.Height()
```

H-Randbedingung: Wasserstandszeitreihe wird vorgegeben.

```
Types.OutflowBC.Ie()
```

Abfluss-Randbedingung, bei der der Abfluss aus einem fest vorgegebenen Energieliniengefälle abgeleitet wird.

```
Types.OutflowBC.QSeries()
```

Abfluss-Randbedingung, bei der der Abfluss als Zeitreihe vorgegeben wird.

```
Types.OutflowBC.WQ()
```

Abfluss-Randbedingung, bei der der Abfluss in funktionaler Abhängigkeit vom mittleren Wasserstand an der Randbedingung berechnet wird.

```
Types.OutflowBC.WQ_GaugePoint()
```

Abfluss-Randbedingung, bei der der Abfluss in funktionaler Abhängigkeit vom Wasserstand eines angegebenen Pegelpunktes berechnet wird.

```
Types.OutflowBC.WeirControl_States()
```

Diese Abfluss-Randbedingung wird zur Abbildung von Wehren benutzt: Das Wehr wird so gesteuert, dass zum Beispiel ein Rückhaltebecken oberhalb des Wehres vor einer Hochwasserwelle geleert wird, um die Wellenspitze zu kappen.

```
Types.OutflowBC.WeirControl_WSeries()
```

Diese Abfluss-Randbedingung wird zur Abbildung von Wehren benutzt: Eine Wasserstandszeitreihe wird vorgegeben.

8.6.12 Klasse hydroas.Weir1D

Diese Klasse gestattet den Zugriff auf 1D-Wehrelemente. Insbesondere können Sie mit dieser Klasse die Höhe des Wehres verändern und den Abfluss über das Wehr abfragen.

8.6.12.1 Statische Klassenfunktionen

```
Weir1D.isWeirType(aType)
```

Parameter:

aType: `hydroas.Types` - Typ

Rückgabe: `Bool`

Liefert `true`, wenn `aType` ein Wehrtyp ist, sonst `false`.

```
Weir1D.new(id)
```

Parameter:

id: `String` - Name eines NodeStrings

oder

id: `Number (Integer)` - Nummer eines NodeStrings

Rückgabe: `Weir1D`

Erzeugt eine `Weir1D`-Instanz zum Zugriff auf die Eigenschaften eines Wehrelementes.

8.6.12.2 Objektmethoden

```
aWeir1D.coefficient()
```

Parameter: -

Rückgabe: `Number (Float)`

Faktor zur Berechnung des Abflusses (Überfallbeiwert)

```
aWeir1D.height()
```

Parameter: -

Rückgabe: Number (Float)

Absolute Höhe der Wehrkrone

```
aWeir1D:name()
```

Parameter: -

Rückgabe: String

Liefert den Name, der in SMS angegeben wurde, oder, wenn der Name nicht gesetzt ist, den Default-Namen.

```
aWeir1D:node1Id()
```

Parameter: -

Rückgabe: Number (Integer)

ID/Nummer des ersten Knotens des 1D-Wehr-Elementes

```
aWeir1D:node2Id()
```

Parameter: -

Rückgabe: Number (Integer)

ID/Nummer des zweiten Knotens des 1D-Wehr-Elementes

```
aWeir1D:q()
```

Parameter: -

Rückgabe: Number (Float)

Abfluss über dieses Wehrelement im zurückliegenden Zeitschritt.

```
aWeir1D:setCoefficient(mu)
```

Parameter:

mu: Number (Float) - Überfallbeiwert

Rückgabe: -

Der Faktor (Überfallbeiwert) zur Berechnung des Abflusses sollte größer als 0 und kleiner oder gleich 1 sein.

Ein Überfallbeiwert von 1 bedeutet verlustfreien Abfluss.

Benutzen Sie diesen Wert, um die Leistungsfähigkeit des Abflusses über das Wehr zu verändern.

```
aWeir1D:setHeight(h)
```

Parameter:

h: Number (Float) - absolute Höhe der Wehrkrone

Rückgabe: -

Absolute Höhe der Wehrkrone setzen: Diesen Wert verändern Sie beispielsweise um eine Wehrsteuerung abzubilden, bei der das Wehr abgesenkt/erhöht wird.

```
aWeir1D:setWidth(w)
```

Parameter:

w: Number (Float) - Breite des Wehrfeldes

Rückgabe: -

Breite des Wehrfeldes setzen: In der Praxis wird dieser Wert wahrscheinlich nicht – oder nur in Ausnahmefälle - geändert werden.

```
aWeir1D:width()
```

Parameter: -

Rückgabe: Number (Float)

Breite des Wehrfeldes: Für das 1D-Wehrelement wird die Breite nicht aus der Netz-Geometrie berechnet, sondern explizit vorgegeben.

8.6.13 Klasse hydroas.WeirSet1D

1D-Wehrelemente werden häufig nicht einzeln, sondern in Gruppen eingesetzt. Die folgende Grafik skizziert ein Wehr und die acht 1D-Wehrüberfall-Elemente mit denen es in HYDRO_AS-2D abgebildet werden kann:

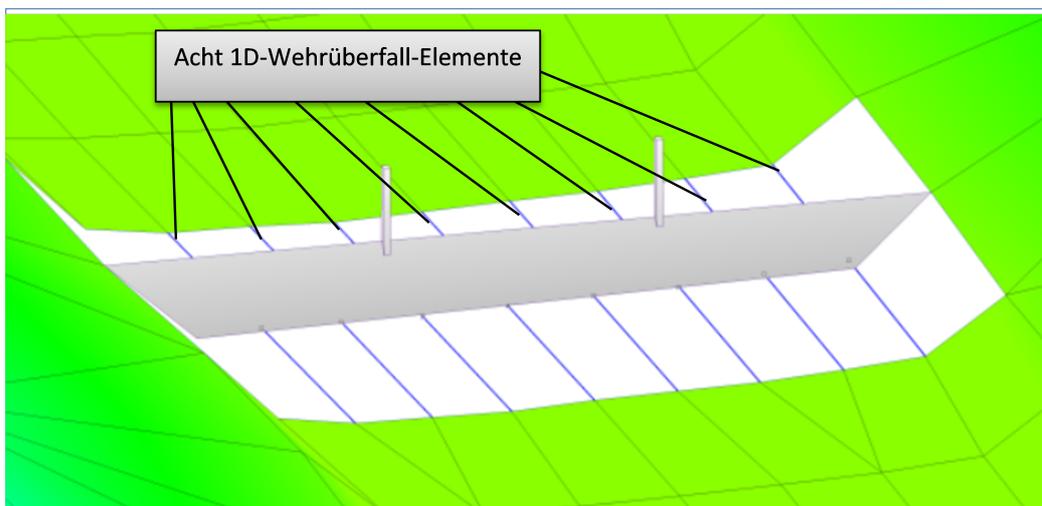


Abbildung 8.4: 1D-Wehrelemente zu Abbildung eines Wehres

Im Modell werden die 2D-Elemente des Wehres normalerweise deaktiviert/gelöscht.

Die Steuerung eines solchen Wehres ist auf Basis der Klasse `Weir1D` zwar grundsätzlich möglich, aber ein wenig umständlich zu programmieren, da jedes Element einzeln angesprochen werden müsste.

Die Klasse `WeirSet1D` gestattet den einfachen und einheitlichen Zugriff auf die Wehrfelder eines Wehres in einem Befehl. Zur Nutzung dieser Klasse muss der Modellierer das Modell aufbereiten, indem er die acht Wehrfelder mit einem Namensschema belegt:

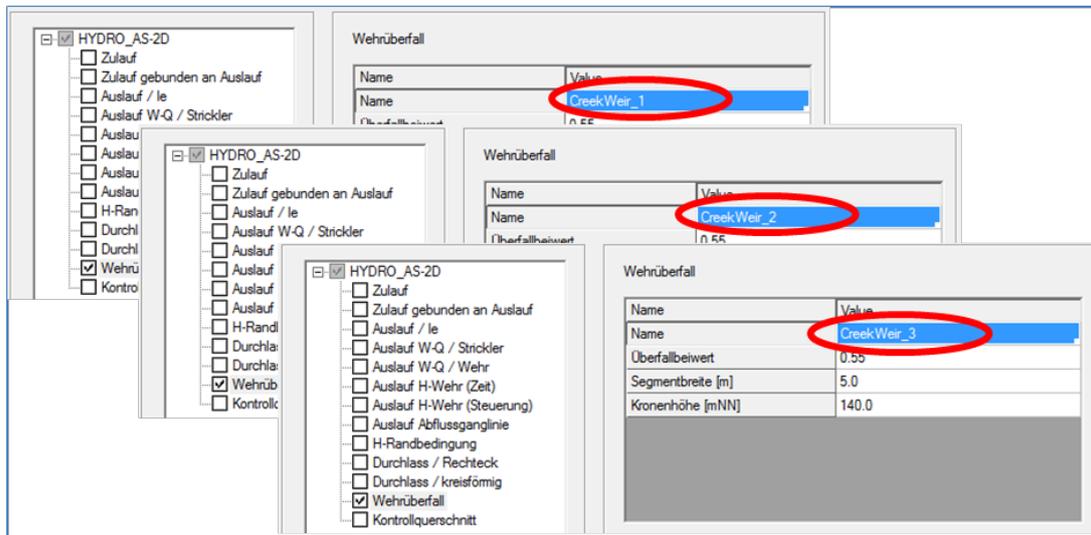


Abbildung 8.5: Namensschema: `CreekWeir_*`

Die Namen der Wehrelemente lauten im Beispiel `CreekWeir_1`, `CreekWeir_2`, ..., `CreekWeir_7`, `CreekWeir_8`. Diese Bezeichnungen sind über einen regulären Ausdruck leicht auszuwählen: `„CreekWeir_.*$“`.

HYDRO_AS-2D-Scripting wählt alle Wehre aus, deren Namen zu diesem Ausdruck passen. Diese Wehre können dann über die Klasse `WeirSet1D` mit einem einzigen einheitlichen Befehl erhöht oder abgesenkt werden.

Achten Sie beim Modellieren darauf, dass alle anderen Bauwerke diesem Namen nicht entsprechen.

8.6.13.1 Statische Klassenfunktionen

```
WeirSet1D.new(namePattern)
```

Parameter:

`namePattern`: `regex` - Muster für Namen

Rückgabe: `WeirSet1D`

Das `namePattern` ist ein regulärer Lua-Ausdruck. Beachten Sie die Lua- Dokumentation zu regulären Ausdrücken.

In vielen Fällen wird der reguläre Ausdruck das Anfangsstück eines Textes beschreiben. Um Namen mit vorgegebenem Anfangsstück auszuwählen, besteht der Ausdruck aus dem Anfangsstück, einem vorangestelltem „^“ und nachgestelltem „.*\$“ (^NiersWehr_.*\$ oder ^Anlage_.*\$).

Um sicher zu gehen, dass keine falschen Elemente ausgewählt werden, kann der Ausdruck ggf. auch restriktiver gestaltet werden: Die Ausdrücke ^NiersWehr_%d+\$ oder ^Anlage_%d+\$ wählen nur Elemente aus, die mit dem Anfangsstück beginnen und dann mit einer Ziffernfolge enden.

Sie sollten darauf achten, die Anfangs- und Endkennner (^und \$) zu benutzen, um nicht versehentlich falsche Objekte mit auszuwählen. Beispielsweise würde das Pattern Anlage_%d+ auch Elemente mit dem Namen WehrAnlage_2 auswählen.

Die Funktion wirft einen Fehler, wenn keine 1D-Wehrelemente durch den Ausdruck beschrieben werden.

Beispiel:

```
local ws = hydroas.WeirSet1D.new("^MyWeir_%d+$")
```

```
aWeirSet:anyCoefficient()
```

Parameter: -

Rückgabe: Number (Float)

Liefert den Abflusskoeffizienten eines der 1D-Wehrelemente zurück, die in dem WeirSet1D enthalten sind.

Diese Funktion ist sinnvoll einsetzbar, wenn das Skript davon ausgehen kann, dass alle 1D-Wehrelemente denselben Abflusskoeffizienten haben. Dies ist insbesondere dann der Fall, wenn zuvor aWeirSet:setCoefficient(x) aufgerufen wurde.

```
aWeirSet:anyHeight()
```

Parameter: -

Rückgabe: Number (Float)

Liefert die absolute Höhe eines der 1D-Wehrelemente zurück, die in dem WeirSet1D enthalten sind.

Diese Funktion ist sinnvoll einsetzbar, wenn das Skript davon ausgehen kann, dass alle 1D-Wehrelemente dieselbe Höhe haben. Dies ist insbesondere dann der Fall, wenn zuvor aWeirSet:setHeight(x) aufgerufen wurde.

```
aWeirSet:anyZ()
```

Parameter: -

Rückgabe: Number (Float)

Identisch zu `aWeirSet:anyHeight(h)`

```
aWeirSet:countElements()
```

Parameter: -

Rückgabe: Number (Integer)

Liefert die Anzahl der 1D-Wehrelemente zurück, die das `WeirSet1D` enthält.

Um sicherzustellen, dass keine falschen oder zu wenige 1D-Wehrelemente erfasst wurden, kann diese Funktion beispielsweise genutzt werden, um die erwartete Anzahl der Elemente zu prüfen:

```
local ws = hydroas.WeirSet1D.new("^MyWeir_%d+$")
assert(ws:countElements() == 8)
```

```
aWeirSet:forallWeirs(func)
```

Parameter: Lua-Funktion

Rückgabe: -

Die Lua-Funktion `func` wird für alle Wehre des `WeirSet1D` durchgeführt.

Beispiel: Sie möchten die Kronenhöhe zwar für alle Wehre im `WeirSet1D` aber nicht identisch, sondern um 5 Meter höher als die Sohlöhe am ersten Knoten des Wehr- Nodestings setzen. Schreiben Sie dazu eine (lokale) Lua-Funktion, die die Kronenhöhe für ein einzelnes Wehr entsprechend ermittelt und setzt. Rufen Sie dann `forallWeirs` mit dieser Funktion auf.

```
local weirset = hydroas.WeirSet1D.new("^MyWeir_%d+$")
local function setMyCrest(aWeir1D)
    local z1 = hydroas.Node.new(aWeir1D:node1Id())
    local w = z1:z() + 5.
    aWeir1D:setHeight(w)
end
weirset:forallWeirs(setMyCrest)
```

```
aWeirSet:fullWidth()
```

Parameter: -

Rückgabe: Number (Float)

Liefert die Breite aller enthaltenen Wehrelemente zurück. Die Gesamtbreite eines `WeirSet1D` ist die Summe aller Breiten der enthaltenen Wehrelemente.

```
aWeirSet:namePattern()
```

Parameter: -

Rückgabe: regEx - Muster für Namen

Liefert das Namens-Schema mit dem die Nodestings gefunden wurden.

```
aWeirSet:q()
```

Parameter: -

Rückgabe: Number (Float)

Liefert den Abfluss über dieses Wehr im zurückliegenden Zeitschritt.

Dazu werden die Abflüsse aller enthaltenen Wehrelemente aufaddiert.

```
aWeirSet:setCoefficient(coeff)
```

Parameter:

coeff: Number (Float) - Abflussbeiwert

Rückgabe: -

Setzt für alle 1D-Wehrelemente im WeirSet1D den Abflusskoeffizienten auf den vorgegebenen Wert.

```
aWeirSet:setFullWidth(w)
```

Parameter:

w: Number (Float) - Breite

Rückgabe: -

Setzt die Breite des Gesamtwehres, indem die Einzelbreiten proportional so angepasst werden, dass die Gesamtbreite dem übergebenen Wert entspricht.

Benutzen Sie diese Funktion, wenn die Einzelbreiten der 1D-Wehre ungefähr korrekt sind, Sie aber sicherstellen wollen, dass die Gesamtbreite – unabhängig von der Korrektheit der exakten Lage der 2D-Elemente – einer Vermessung entspricht.

Wenn das WeirSet1D aus n Elementen besteht, und jedes Wehrfeld die Breite

w_i^{alt} hat, wird diese Breite nach folgender Formel korrigiert:

$$w_i^{neu} = w_i^{alt} \frac{w}{\sum_k^n w_k^{alt}}$$

```
aWeirSet:setHeight(h)
```

Parameter:

h: Number (Float) - Höhe

Rückgabe: -

Setzt für alle 1D-Wehrelemente im WeirSet1D die Höhe auf den vorgegebenen Wert.

```
aWeirSet:setZ(h)
```

Parameter:

`h: Number (Float)` - Höhe

Rückgabe: -

Identisch zu `aWeirSet:setHeight(h)`

8.6.13.2 Objektmethoden

8.7 Lua

Die Standardlieferung von HYDRO_AS-2D-Scripting beinhaltet Zusatz-Module, die nicht Teil der regulären Lua-Distribution, aber auch nicht HYDRO_AS-2D-spezifisch sind. Die Lizenzen dieser Module erlauben die freie Nutzung.

8.7.1 Zusatzmodul Lua-Filesystem

„lfs“ steht für „Lua-Filesystem“. Dieses Modul enthält Basisoperationen für das Dateisystem. Beispielsweise sind enthalten:

- Inhalt von Verzeichnissen auflisten
- Attribute von Dateien abfragen
- Verzeichnisse anlegen
- Verzeichnisse und Dateien löschen

Autoren sind Roberto Ierusalimsky, André Carregal und Tomás Guisasola.

Um „lfs“ zu nutzen rufen Sie lediglich die Funktionen für die globale Variable „lfs“ auf. Die Variable „lfs“ ist automatisch vorhanden. Das Statement `require(lfs)` kann daher nicht benutzt werden. Dies führt zu einem Fehler.

Eine Liste der enthaltenen Funktionen gibt es - nur auf Englisch- unter <https://keplerproject.github.io/luafilesystem/manual.html>.

Im Folgenden finden Sie eine Kopie dieser Liste (aufgerufen am 04.09.2019):

```
lfs.attributes (filepath [, aname | atable])
```

Returns a table with the file attributes corresponding to `filepath` (or `nil` followed by an error message and a system-dependent error code in case of error). If the second optional argument is given and is a string, then only the value of the named attribute is returned (this use is equivalent to `fs.attributes(filepath)[aname]`, but the table is not created and only one attribute is retrieved from the O.S.). if a table is passed as the second argument, it is filled with attributes and returned instead of a new table. The attributes are described as follows; attribute `mode` is a string, all the others are numbers, and the time related attributes use the same time reference of `os.time`:

`dev`

on Unix systems, this represents the device that the inode resides on.

On Windows systems, represents the drive number of the disk containing the file

`ino`

on Unix systems, this represents the inode number.

On Windows systems this has no meaning

`mode`

string representing the associated protection mode

the values could be file, directory, link, socket, named pipe, char device, block device or other

`nlink`

number of hard links to the file

`uid`

user-id of owner (Unix only, always 0 on Windows)

`gid`

group-id of owner (Unix only, always 0 on Windows)

`rdev`

on Unix systems, represents the device type, for special file inodes.

On Windows systems represents the same as `dev`

`access`

time of last access

`modification`

time of last data modification

`change`

time of last file status change

`size`

file size, in bytes

`permissions`

file permissions string

`blocks`

block allocated for file; (Unix only)

`blksize`

optimal file system I/O blocksize; (Unix only)

This function uses `stat` internally thus if the given filepath is a symbolic link, it is followed (if it points to another link the chain is followed recursively) and the information is about the file it refers to. To obtain information about the link itself, see function `lfs.symlinkattributes`.

`lfs.chdir (path)`

Changes the current working directory to the given path.

Returns `true` in case of success or `nil` plus an error string.

`lfs.lock_dir(path, [seconds_stale])`

Creates a lockfile (called `lockfile.lfs`) in `path` if it does not exist and returns the lock. If the lock already exists checks if it's stale, using the second parameter (default for the second parameter is `INT_MAX`, which in practice means the lock will never be stale. To free the the lock call `lock:free()`.

In case of any errors it returns `nil` and the error message. In particular, if the lock exists and is not stale it returns the *File exists* message.

```
lfs.currentdir ()
```

Returns a string with the current working directory or `nil` plus an error string.

```
iter, dir_obj = lfs.dir (path)
```

Lua iterator over the entries of a given directory. Each time the iterator is called with `dir_obj` it returns a directory entry's name as a string, or `nil` if there are no more entries.

You can also iterate by calling `dir_obj:next()`, and explicitly close the directory before the iteration finished with `dir_obj:close()`. Raises an error if `path` is not a directory.

```
lfs.lock (filehandle, mode[, start[, length]])
```

Locks a file or a part of it. This function works on open files; the file handle should be specified as the first argument. The string `mode` could be either `r` (for a read/shared lock) or `w` (for a write/exclusive lock). The optional arguments `start` and `length` can be used to specify a starting point and its length; both should be numbers.

Returns `true` if the operation was successful; in case of error, it returns `nil` plus an error string.

```
lfs.link (old, new[, symlink])
```

Creates a link. The first argument is the object to link to and the second is the name of the link. If the optional third argument is `true`, the link will be a symbolic link (by default, a hard link is created).

```
lfs.mkdir (dirname)
```

Creates a new directory. The argument is the name of the new directory.

Returns `true` in case of success or `nil`, an error message and a system-dependent error code in case of error.

```
lfs.rmdir (dirname)
```

Removes an existing directory. The argument is the name of the directory.

Returns `true` in case of success or `nil`, an error message and a system-dependent error code in case of error.

```
lfs.setmode (file, mode)
```

Sets the writing mode for a file. The mode string can be either `"binary"` or `"text"`. Returns `true` followed the previous mode string for the file, or `nil` followed by an error string in case of errors. On non-Windows platforms, where the two modes are identical, setting the mode has no effect, and the mode is always returned as `binary`.

```
lfs.symlinkattributes (filepath [, aname])
```

Identical to `lfs.attributes` except that it obtains information about the link itself (not the file it refers to). It also adds a `target` field, containing the file name that the symlink points to. On Windows this function does not yet support links, and is identical to `lfs.attributes`.

```
lfs.touch (filepath [, atime [, mtime]])
```

Set access and modification times of a file. This function is a bind to `utime` function. The first argument is the filename, the second argument (`atime`) is the access time, and the third argument (`mtime`) is the modification time. Both times are provided in seconds (which should be generated with Lua standard function `os.time`). If the modification time is omitted, the access time provided is used; if both times are omitted, the current time is used.

Returns `true` in case of success or `nil`, an error message and a system-dependent error code in case of error.

```
lfs.unlock (filehandle[, start[, length]])
```

Unlocks a file or a part of it. This function works on open files; the file handle should be specified as the first argument. The optional arguments `start` and `length` can be used to specify a starting point and its length; both should be numbers.

Returns `true` if the operation was successful; in case of error, it returns `nil` plus an error string.

8.7.2 Zusatzmodul Lua-Date

„LuaDate“ ist ein Modul für Datums- und Zeit-Berechnungen sowie dem Umgang mit dem Gregorianischem Kalender.

Autor ist Thijs Schreijer.

Das Modul nutzen Sie, indem Sie das Statement `require(date)` in ihrem Skript aufrufen.

Eine Dokumentation zu diesem Modul finden Sie in der HYDRO_AS-2D Installation im Ordner `lua/ date`.

8.7.3 Literatur und Referenzen

HYDRO_AS-2D-Scripting basiert auf der Programmiersprache Lua 5.3.4. Der Sprachumfang von Lua ist vergleichsweise klein, sodass es möglich ist, mit wenig Lernaufwand erfolgreich sinnvolle Skripte zu entwickeln.

Für den Fall, dass Sie komplexere Skripte benötigen oder fortgeschrittene Programmieraufgaben umsetzen wollen, sollten Sie Lua systematischer erlernen. Das Internet wird dazu umfassende Informationen liefern.

Geeignete Quellen sind beispielsweise:

Quelle	Beschreibung
Lua-Web-Seite: https://www.lua.org/	Diese Web-Site enthält umfangreiche Dokumentationen in englischer (und teilweise portugiesischer) Sprache.
	Diese Web-Site ist die Hauptreferenz bzgl. aller Belange der Sprache Lua.

Quelle	Beschreibung
Buch "Programming in Lua" von R. Ierusalimschy	von "Programming in Lua" ist das bekannteste Einführungsbuch zu Lua. Der Autor Roberto Ierusalimschy ist Architekt der Sprache Lua.
Lua.org, fourth edition, August 2016	Das Buch ist das Standardwerk zum Thema. Es führt die Sprache Lua grundlegend anhand einfacher Beispiele ein - setzt aber voraus, dass der Leser ein grundsätzliches Verständnis über Programmierung mitbringt. Neben russischen und portugiesischen Übersetzungen liegt eine ältere Auflage auch in deutscher Übersetzung vor, wird aber aktuell nicht mehr verlegt. Für den Einsatz in HYDRO_AS-2D wird i. Allg. auch eine ältere Auflage des Buches reichen.
Lua 5.1 "Reference Manual" von R. Ierusalimschy, L. H. de Figueiredo, W. Celes,	Das Referenzhandbuch beschreibt die Sprache Lua vollständig und formal eindeutig. HYDRO_AS-2D Anwender werden diese Informationen in der Tiefe normalerweise nicht benötigen oder ggf. für einzelne Fragestellungen in der Internet-Version des Buches nachlesen.
Lua.org, August 2006 ISBN 8590379833	Im Internet finden Sie auch freie deutsche Übersetzungen.
Lua Unofficial FAQ (uFAQ) http://www.luafaq.org/	Diese Seite enthält viele nützliche Tipps und Tricks zum Umgang mit LUA
Package manager für LuaModule https://luarocks.org/	Website beschreibt und verwaltet LUA-Zusatzmodule.

8.7.4 Lua-Programm-Verzeichnisse

Im Installationsverzeichnis von HYDRO_AS-2D finden Sie ein Verzeichnis *lua*. Dieses Verzeichnis enthält die im [Kapitel 8.5](#) aufgeführten Klassen und wird automatisch gefunden, wenn die Datei *Data-in/control.lua* existiert.

Das zentrale Quellskript ist immer die Datei *Data-in/control.lua*. Sie können den Lua-require- Mechanismus auch verwenden, um Code in andere Dateien auszulagern.

Diese Dateien müssen ebenfalls in Verzeichnissen namens *lua* gespeichert werden. Wenn in dem Verzeichnis *working directory* die Modelldatei *hydro_as-2d.2dm* liegt, dann wird in allen Verzeichnissen ab dem Verzeichnis *working directory/Data-in* nach *lua*-Verzeichnissen gesucht.

Beispiel: Wenn die Modelldatei *C:/projekte/meinFluss/VarianteA/HQ100/hydro_as-2d.2dm* ist, dann wird an folgenden Stellen nach *lua*-Verzeichnissen gesucht:

- *C:/lua*
- *C:/projekte/lua*
- *C:/projekte/meinFluss/lua*
- *C:/projekte/meinFluss/VarianteA/lua*
- *C:/projekte/meinFluss/VarianteA/HQ100/lua*

- *C:/projekte/meinFluss/VarianteA/HQ100/Data-in/lua*

Das tiefste Verzeichnis (*Data-in/lua*) hat dabei die höchste Priorität - die anderen Verzeichnisse geringere. Zusätzlich kann die Umgebungsvariable `Lua_PATH` eine „;-“ getrennte Liste von Verzeichnisnamen enthalten, in denen Lua-Dateien gesucht werden.

Diese Dateien haben eine geringere Suchpriorität.

9 Validierung

9.1 Strömung in einer trapezförmigen Rinne

Die genaue Geschwindigkeits- / Schubspannungsermittlung im Flussquerschnitt ist eine wichtige Voraussetzung für zutreffende Geschiebetransportmodellierung. Nachfolgend werden Schubspannungen und tiefengemittelte Fließgeschwindigkeiten im Trapezquerschnitt berechnet und mit Ergebnissen aus der Literatur verglichen [16, 30]. Als Referenzwerte liegen Simulationsergebnisse des Strömungsmodells SIMK [16] und RISMO vor. RISMO ist ein Finite-Elemente Strömungsprogramm zur Lösung der tiefengemittelten Flachwassergleichungen. SIMK bildet die Strömung zweidimensional in einer Querschnittsebene ab. Das Geschwindigkeitsfeld wird dreidimensional gelöst. Es wird ein stationär-gleichförmiger Zustand berechnet, da alle Gradienten in Hauptströmungsrichtung vernachlässigt werden. Der Einsatz dieses Strömungsmodells für die Kalibrierung und Validierung des 2D-Modells hat sich bei den Untersuchungen von Schmutz [30] bewährt, sodass hier ebenfalls auf diesen Vergleich zurückgegriffen wird.

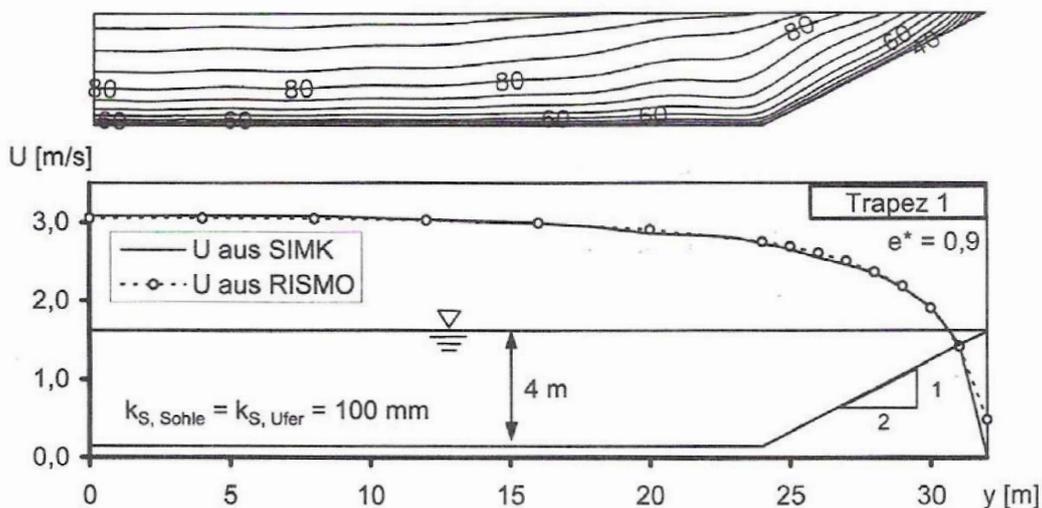


Abbildung 9.1: Ermittelte Fließgeschwindigkeit im Trapezquerschnitt [30]

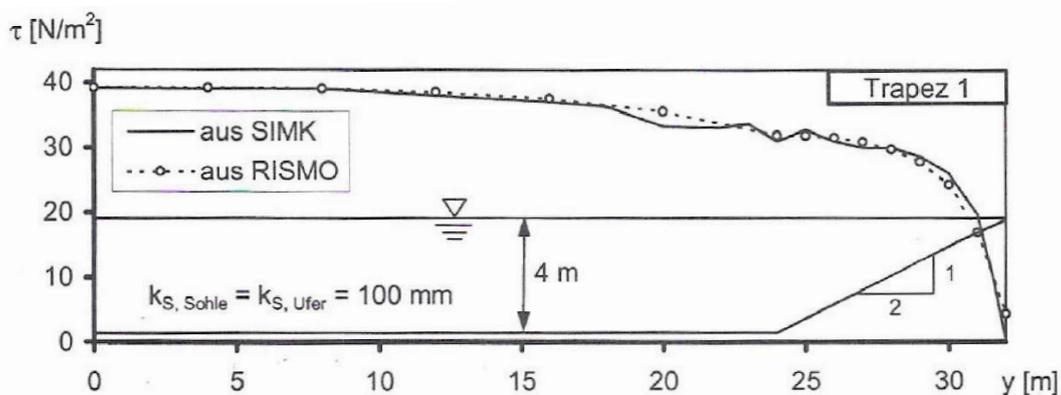


Abbildung 9.2: Ermittelte Schubspannung im Trapezquerschnitt [30]

Der Trapezquerschnitt hat eine Breite an der Sohle von $b_s = 48$ m und die Böschungsneigung beträgt 1:2. Die Rinne ist symmetrisch, sodass nur eine Hälfte der Rinne modelliert wird. Die Strömung erfolgt

bei bordvollem Abfluss mit einer Wassertiefe von 4 m. Das Sohlgefälle ist konstant und beträgt 1%. Die äquivalente Sandrauheit wurde mit $k_s = 100$ mm angegeben.

Die Schubspannung τ wird in HYDRO_AS-2D mit folgendem Ansatz berechnet:

$$\tau = \rho \cdot g \cdot h \cdot I_R \quad (22)$$

wobei

ρ = Dichte des Wassers

g = Erdbeschleunigung

h = Wassertiefe

Das Reibungsgefälle I_R wird entsprechend der Strickler-Formel berechnet:

$$I_R = \frac{v^2}{\left(K_{St}^2 \cdot h^{\frac{4}{3}}\right)} \quad (23)$$

mit

v = Größe der tiefengemittelten Fließgeschwindigkeit

Für die zweidimensionale Berechnung mit HYDRO_AS-2D wurde die äquivalente Sandrauheit wie folgt in Strickler-Werte umgerechnet:

$$K_{St} = 8,3 \cdot \sqrt{g} \cdot k_s^{-\frac{1}{6}} = 38,163 \quad (24)$$

Die folgende Abbildung stellt das für die 2D-Simulation mit HYDRO_AS-2D eingesetzte Berechnungsnetz sowie die berechnete Schubspannungsverteilung dar. Die Fließstrecke ist 235 m lang. Der Punktabstand variiert in Querrichtung zwischen 1 m im Böschungsbereich und 2 m in der Flussmitte. Für die Simulation wurden insgesamt 1.034 Elemente und 1.104 Punkte verwendet.

Die Auswertung erfolgte im Querschnitt, der 130 m vom Zulauf entfernt liegt. Sowohl die ermittelten tiefengemittelten Fließgeschwindigkeiten als auch die Schubspannungen zeigen eine gute Übereinstimmung mit den Angaben aus der Literatur.

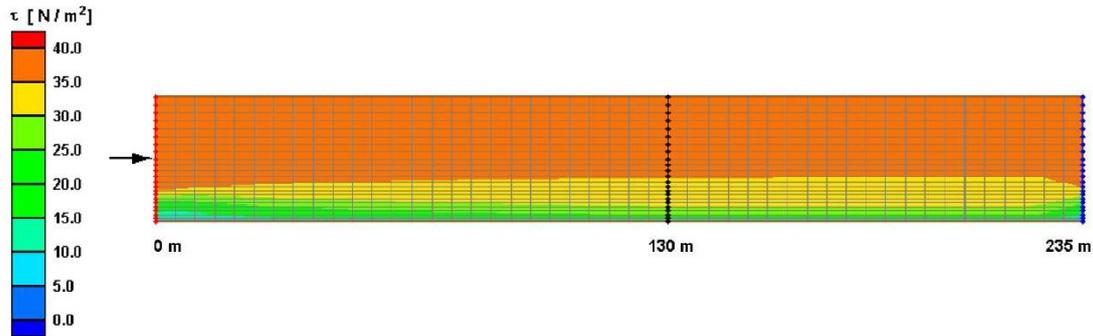


Abbildung 9.3: Berechnungsnetz und ermittelte Schubspannungsverteilung

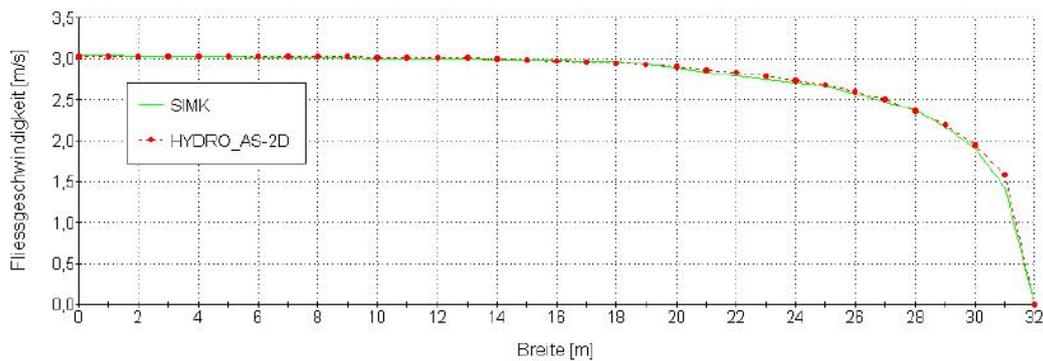


Abbildung 9.4: Ermittelte tiefengemittelte Fließgeschwindigkeiten im Trapezquerschnitt

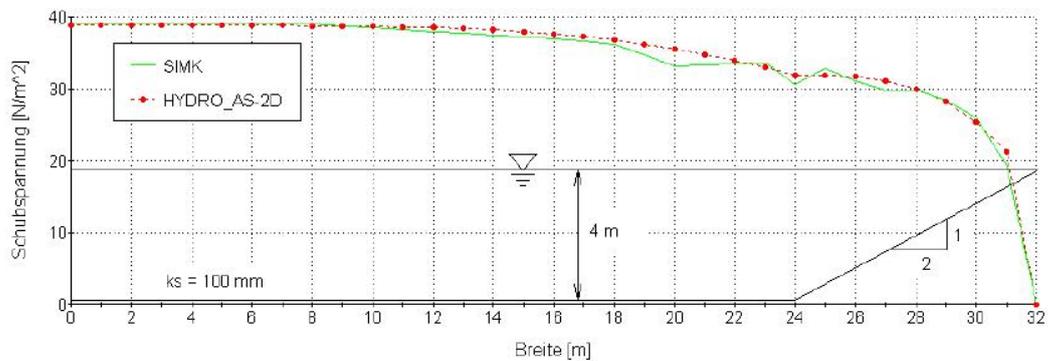


Abbildung 9.5: Ermittelte Schubspannungen im Trapezquerschnitt

9.2 Dambruch 1D

Ein weiteres Validierungsbeispiel stellt die Nachrechnung des Modellversuches von Lauber und Hager dar [18]:

Eine Dambruchwelle breitet sich auf dem Gelände mit einer Neigung von 10 % aus. Dieses Beispiel demonstriert, dass die Dambruchwellenausbreitung selbst bei einer derart großen Geländeneigung gut erfasst wird.

Anmerkung: Das numerische Modell wurde entsprechend des physikalischen Modellversuchs aufgebaut. Für einen Vergleich ist deswegen eine Umrechnung der Ergebnisdaten in eine dimensionslose Form erforderlich.

Die Fließstrecke ist 15 m lang und 0,5 m breit. Für die Simulation wurden insgesamt 3.000 Elemente und 3.311 Punkte verwendet. Es liegt nur ein Materialtyp mit einem Rauigkeitsbeiwert von $K_{St} = 100$ vor.

Die Berechnung wurde mit dem 2step-Verfahren HYDRO_AS durchgeführt, mit dem u. a. auch hoch instationäre Vorgänge, wie die Ausbreitung einer Dammbrechwelle, erfolgreich simuliert werden. Die gewählte Zeitdiskretisierung entspricht somit dem expliziten Runge-Kutta-Verfahren zweiter Ordnung.

Hinweis: Bei größeren Geländeneigungen ($Is > 3\%$), sollte ein geringerer Punktabstand in Strömungsrichtung ($dl < 5$ m in Realität) verwendet werden, damit zuverlässige Ergebnisse erzielt werden.

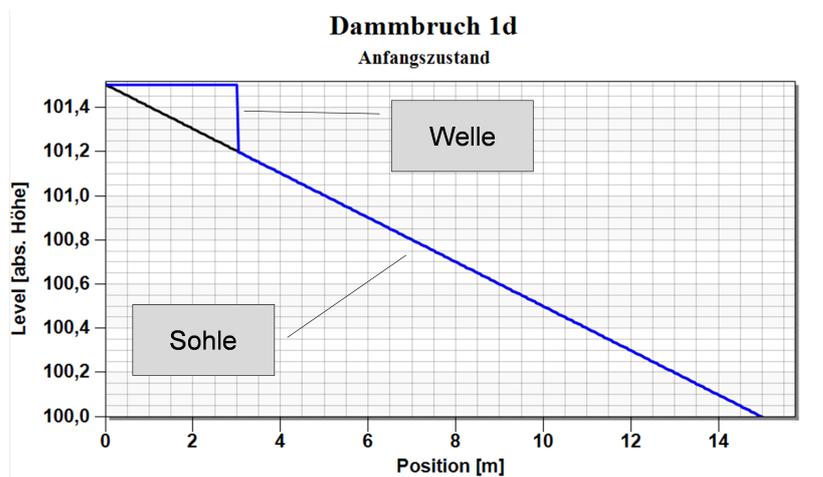


Abbildung 9.6: Validierungsbeispiel Dammbrech 1D

9.3 Dammbrech 2D

Das Validierungsbeispiel bezieht sich auf den Modellversuch der Catholic University of Louvain. Das Modell war ein Testfall für numerische Modelle im Rahmen des CADAM (= Concerted Action on Dam-break Modelling) Workshops über die Dammbrechmodellierung (siehe [24]).

Die Berechnung wurde ebenfalls mit dem 2step-Verfahren HYDRO_AS durchgeführt.

9.4 Trapezkanal

Zur Validierung der H-Randbedingung wurde das folgende Modell erstellt:

Dieses Beispiel hat folgende Geometrie und Parameter:

- trapezförmiger Kanal mit einer Sohlenbreite von 30 m und einer Böschungsneigung von 1:2
- Sohlgefälle = 0,001
- Rauheit: $K_{St} = 45 \text{ m}^{1/3}/\text{s}$
- Länge = 7 km

- vorgegebene Wassertiefe unten (Kilometer 0) = 5,63 m
- Zufluss oben (Kilometer 7): $Q = 139 \text{ m}^3/\text{s}$

Die Staulinie dieser stationären Situation kann mithilfe der Fließformel nach Manning-Strickler bestimmt werden. In [21] sind diese berechneten Werte alle 500 m tabellarisch angegeben. Ab Kilometer 5,280 flussaufwärts stellt sich ein gleichförmiger Abfluss mit der Wassertiefe 2 m ein.

In HYDRO_AS-2D wurde die vorgegebene untere Wassertiefe durch eine H-Randbedingung dargestellt. Die Simulation liefert die **gesuchte Staulinie**. Der Vergleich der mit HYDRO_AS-2D berechneten Wassertiefen und der Tabellenwerte aus [21] (untere Abbildung), zeigt vergleichbare Ergebnisse. Lediglich am oberen Kanalende sind die mit HYDRO_AS-2D berechneten Wassertiefen etwas niedriger als die theoretischen Werte. Im Wesentlichen sind die Ergebnisse vergleichbar.

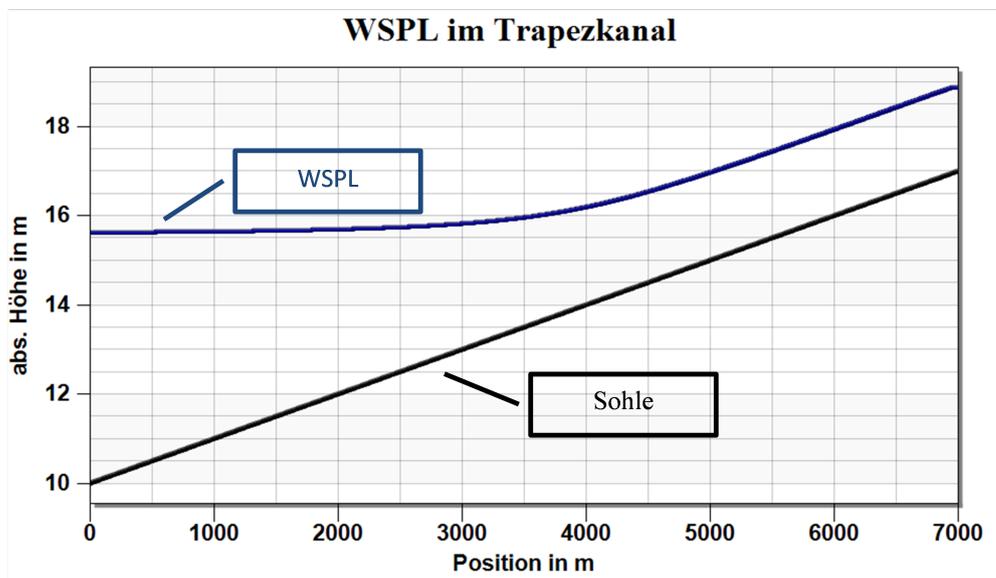


Abbildung 9.7: Stationärer Wasserspiegel im Trapezkanal mit HYDRO_AS-2D berechnet

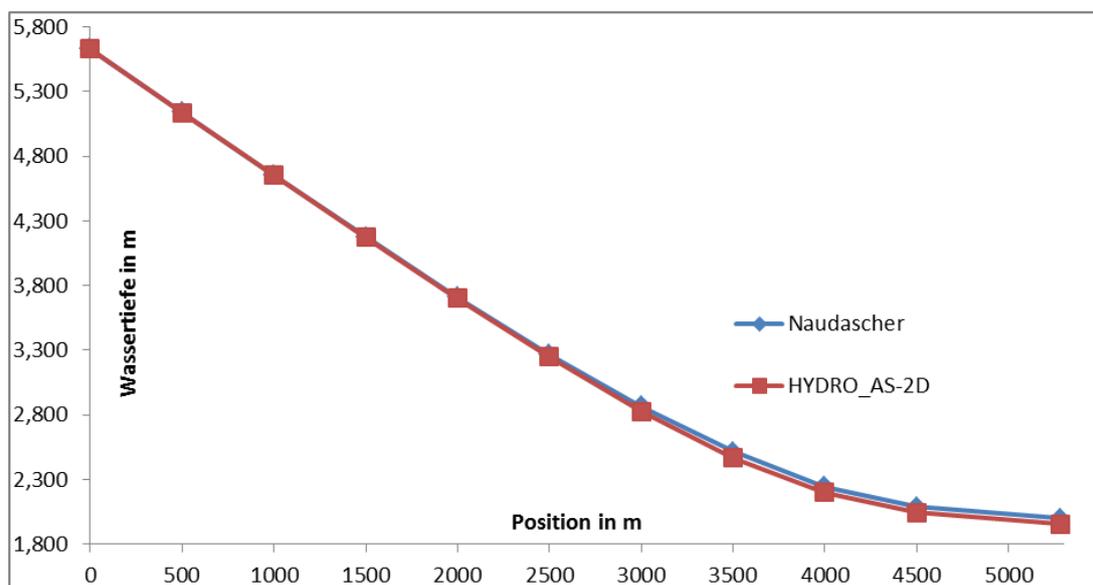


Abbildung 9.8: Vergleich der Wassertiefen im Trapezkanal

10 Release-Notes

10.1 Version 5.3.1 (12.10.2021)

- MapWork: In MapWork können nun auch MapView-Datensätze erzeugt werden.
- MapWork: Verbesserte Ausgabe der verwendeten Materialien.
- MapWork: Überprüfung der Landnutzungspolygone wird nun nach ESRI-Konvention durchgeführt.
- MapView: Es ist nun möglich MapView-Offline analog zu MapView-Online zu konfigurieren. Dies beinhaltet beispielsweise frei konfigurierbare Hintergrundkarten, Beschreibungstexte mit Hyperlinks, freie Farbgebungen für die Geschwindigkeitsanimation und anderes. Gehen Sie dazu wie folgt vor:
 - Starten Sie MapView-Online (<https://mapview.hydrotec.de>) und melden sich an,
 - öffnen für ein Modell den entsprechenden Konfigurationsdialog,
 - nehmen Sie die gewünschten Einstellungen vor,
 - kopieren Sie die Einstellungen im JSON-Format aus dem Reiter 'Texteingabe' in einen Texteditor und
 - speichern Sie diese Datei unter dem Namen 'config.hmvc' oder dem Namen '<Name HMV-Datei>.hmv' neben die HMV-Datei.

Die hmv-Datei versenden Sie dann zusammen mit der hmv-Datei.

- MapView-Offline: KML- und KMZ-Dateien können mittels 'Drag and Drop' im MapView-Offline-Viewer angezeigt werden.
- MapView: MapView unterstützt nun die XMDF(h5)- und ASCII(dat.)-Dateiformate, die beim Export aus SMS erzeugt werden.
- HYDRO_AS-2D: Die Simulationsprogramme lesen nun auch das ASCII(dat.)-Dateiformat von SMS ein, das zusätzlich zu den Knoten-Werten Elementwerte (activity) beinhaltet.
- dattoh5: Das Konvertierungstool dattoh5 konvertiert nun auch das ASCII(dat.)-Dateiformat von SMS, das zusätzlich zu den Knoten-Werten Elementwerte (activity) beinhaltet.

10.2 Version 5.3.0 (10.09.2021)

- MapWork wird mit ausgeliefert. Für die Nutzung von MapWork wird eine zusätzliche Lizenz benötigt. Weitere Informationen finden Sie im Handbuch zu MapWork.
- Pro HYDRO_AS-2D Lizenz können vier *Prozessorkerne* statt wie bisher zwei genutzt werden.
- Die Vorlagedatei enthält ein Element mit den Koordinaten (0,0), (0,1) und (1,0). Dieses Element ist nur ein Hilfsmittel, um einen Fehler in SMS zu umgehen, der dazu führt, dass die Parameter der Standard-Materialien nicht übernommen werden.
- Alle Knoten-basierten Eingabedateien können auch als Binärdatei (h5) eingelesen werden.
- Unter Hilfsprogramme finden Sie ein neues Tool *dattoh5* zur Konvertierung von Eingabedateien im ASCII-Format in Binärdateien (h5). Siehe auch [Kapitel 5.2](#).
- Die Batch-Jobs (nur Windows) zum Aufruf der Tools *h5todat* und *dattoh5* befinden sich jetzt im Installationsverzeichnis der Version unter Hilfsprogramme.

- Neue Ausgabedateien: Es gibt zwei neue Ausgabedateien. Die Datei *schacht.dat* enthält Informationen zu Abflüssen von Schächten (Kontenrandbedingung Zufluss/Abfluss). Die Datei *volumenbilanz.dat* enthält Informationen zu den Wasservolumen während der Simulation. Beide Dateien sind standardmäßig deaktiviert und müssen bei Bedarf manuell aktiviert werden.
- Von der GUI aus können zusätzliche Ausgaben analog zur Datei *hydro_as-sim.inp* angefordert werden.
- Verbesserung Rechenlauf fortsetzen: Die Datei *q_strg.dat* wurde in manchen Konstellationen von Ausgabeschritten (kleine Werte, z.B. 1 Sekunde, oder Änderung des Ausgabeschritts für den fortgesetzten Rechenlauf) nicht korrekt fortgesetzt.
- Scripting Verzeichnis *lua*: Das durch den Nutzer angelegte Verzeichnis *lua* kann auch im Data-in-Verzeichnis liegen. Es wird auch dort automatisch gefunden. Bisher war die erste mögliche Ebene das Verzeichnis des Modells.
- Scripting erweitert: Die Klasse *Global* liefert den Koordinatenursprung des Modells zur Umrechnung von relativen Koordinaten in absolute Koordinaten.
- Scripting erweitert: *Nodesets* können über ein Namensschema angelegt werden. Zusätzlich bietet die Klasse *Nodeset* eine Reihe weiterer Funktionen, siehe [Kapitel 8.6.8](#).
- Format der Ausgabedatei *bw_temp.dat*: Die Breite der ersten beiden Spalten für die Angabe der Knotennummern wurde auf 12 Stellen erweitert. So werden nun auch Knotennummern im Hundertmillionenbereich durch Leerzeichen getrennt.
- Neue Fehlermeldungen im Präprozessor zum Modellrand: Der Präprozessor gibt einen Fehler aus, wenn das Netz schlitzförmige Löcher oder Einbuchtungen aufweist.
- Neue Fehlermeldungen im Präprozessor zu Randbedingungen: Der Präprozessor gibt einen Fehler aus, wenn ein Knoten mehrfach mit einer Randbedingung vom selben Typ belegt ist.
- Neue Information im Präprozessor zu Materialien: Der Präprozessor liefert Informationen zu verwendeten und nicht verwendeten Materialien.
- Speichermanagement im Präprozessor verbessert: Das Speicherverbrauch zur Laufzeit des Präprozessors wurde verringert.
- Nutzereinstellungen in der GUI: Durch den Nutzer in der GUI gesetzte Parameter wurden nach dem Neuladen des Modells weiterhin angezeigt, aber nicht genutzt. Die Eingaben werden nun mit dem Neuladen zurückgesetzt.
- Die Codemeter-Version in der Auslieferung wurde auf die Version 7.21a aktualisiert.
- MapView: *Max-Zustand* wird auch bei nicht vorhandener *veloc_max*-Datei erzeugt, wenn die Geschwindigkeiten-Option ausgewählt wurde.
- MapView: Pegel haben nun auch Höheninformation.
- MapView: Abflusskurve wurde manchmal zu stark ausgedünnt.
- MapView: Speichernutzung von rechteckigen Elementen verringert.

10.3 Version 5.2.5 (03.03.2021)

- **Behebung eines Problems im Zusammenhang mit A_{min} :** Die in 5.2.0 vorgenommene Änderung (siehe [Kapitel 10.8](#)) hatte sich in manchen Modellen ungünstig ausgewirkt. Es kann nun stattdessen zu Gunsten der Stabilität zu etwas kleineren internen Zeitschritten kommen. Starke Auswirkungen auf die Rechenzeit sind nicht zu erwarten.
- Zulauf gebunden an Auslauf: Die Abflusswerte für Zuflussrandbedingungen vom Typ Zulauf gebunden an Auslauf werden nun direkt aus den Abflüssen der zugehörigen Auslaufrandbedingungen ermittelt. Bisher waren die Werte um einen internen Zeitschritt verschoben.
- Tool h5todat.exe: Eine neue Option zum Schreiben von ASCII-Dateien im freien Format wurde hinzugefügt. Siehe Konvertieren von Ausgabedateien.
- Die Ausgabedateien qc_strg.dat, qc_fa.dat, qg_strg.dat und qg_fa.dat zu den Feststofftransportmodulen berücksichtigen nun auch die Einstellung „Ausgabe q_strg 6 Nachkommastellen“.

10.4 Version 5.2.4 (01.12.2020)

- Rechenlauf fortsetzen: Ausgabedateien wurden beim Fortsetzen eines Rechenlauf nicht richtig weitergeschrieben, wenn die cpr-Dateien durch die entsprechende Option nur zum letzten Zeitschritt geschrieben wurden.
- Ausgabedatei *dauer(.dat/h5)*: Die Werte in der Ausgabedatei waren in manchen Fällen nicht korrekt. Es kann Unterschiede in der Datei im Vergleich zur vorherigen Version geben.
- Ausgabedatei *strickler.dat*: Die Namen in der Datei werden jetzt ohne Leerzeichen geschrieben (stattdessen „_“), damit sie in SMS richtig dargestellt werden.
- Scripting: Die Funktion `NodeSet.newByNodes` wurde korrigiert.
- MapView: Kommandozeilenargument zum identifizieren von Pegeln und Querprofilen über den Namen
- MapView: von der GUI erzeugte Argumente nutzen neues Argument für Namen, wenn Querprofil oder Pegel einen Namen hat
- Die Codemeter-Version in der Auslieferung wurde auf die Version 7.10a aktualisiert.

10.5 Version 5.2.3 (01.10.2020)

- Präprozessor brach ab, wenn in einer Kontenrandbedingung (KUK oder Pegelpunkt) in der Anweisung `BC_VAL N` eine nicht existierende Knotennummer angegeben war (z.B. -1). Dasselbe gilt für Nodestringrandbedingungen (`BC_VAL S`) mit ungültigen NodestringIDs.
- Präprozessor meldet einen Fehler bei ungültigen (negativen oder nicht definierten) Material-IDs. Weitere Fehlermeldungen im Zusammenhang mit Materialdefinitionen wurden verbessert.
- Präprozessor überprüft Nodestring- und Pegelpunktnamen auf Leerzeichen und kürzt Namen beim Leerzeichen und/oder auf 12 Stellen.
- Präprozessor: Prüfungen auf mehrfach vorkommende Namen werden für gekürzte Namen durchgeführt.
- Scripting: Abfrage der Anzahl der Zuflüsse vom Typ Zulauf gebunden an Auslauf und Zulauf (Zeitreihe) sind nun getrennt möglich. Siehe [Kapitel 8.6.6](#).

- Scripting: Zugriff auf Kornfraktionen, wenn ein Stoff- oder Geschiebetransport-Modul benutzt wird. Siehe [Kapitel 8.6.5](#).
- Scripting: Implementierung von WeirSet1D:forallWeirs und CulvertSet:forallCulverts wurde korrigiert.
- Feflow-in.dat: Daten in Exponentialschreibweise wurden nicht richtig eingelesen.
- MapView: Zeitschritteüberspringen bei Textergebnissen (.dat-Dateien) an Binärergebnisse angepasst. Zeitschritte werden nach dem Startzeitschritt übersprungen.

10.6 Version 5.2.2 (24.08.2020)

- Präprozessor: Die Datei hydro_as-2d.inp wurde in einem falschen Format geschrieben, wenn die Namen der Eingabedateien im Voraus in SMS in der Registerkarte EIN/AUSGABEDATEIEN (E/A: Dateiname.dat/.h5) gelöscht wurden.
- Scripting erweitert: In der Klasse Global gibt es zwei neue Funktionen zum Beenden der Simulation und das Schreiben von Ergebnissen. Siehe [Kapitel 8.6.4](#)
- Die Ausgabedatei der Strickler-Werte wurde für Knoten verbessert, die sich innerhalb eines deaktivierten Bereichs befinden.
- Im Verzeichnis SMS-2D-Template finden Sie eine Datei control.lua, die Sie als Vorlage für Skripte verwenden können.
- MapView-Berechnung der hmv-Datei: In einigen Modellen wurde die Berechnung des Umfangs nicht beendet.

10.7 Version 5.2.1 (10.06.2020)

- Randbedingungen H-Wehr (Zeit) und H-Wehr (Steuerung): Die Abflussabminderung durch unvollkommenen Überfall wurde korrigiert. Die Abminderung wirkt nur auf den maximal möglichen Abfluss.
- MapView: Umlaute in Modellnamen und Bezeichnungen (Pegelpunkte, Kontrollquerschnitte) werden in MapView nun richtig übernommen.

10.8 Version 5.2.0 (25.05.2020)

- MapView wird mit ausgeliefert. Für das Erstellen von MapView-Dateien wird eine zusätzliche Lizenz benötigt. Weitere Informationen finden Sie in [Kapitel 7.3](#).
- In der Auslieferung gibt es ein neues Tool zur Konvertierung von binären (.h5) Ergebnisdateien in das ASCII-Format (.dat). Es liegt im Ordner Hilfsprogramme und kann ohne weitere Lizenz benutzt werden. Siehe auch [Kapitel 5.2](#).
- Rechenlauf fortsetzen: Die Ausgabedateien q_strg.dat, pegel.dat, bw_temp.dat und wehr.dat sowie die SMS-Ausgabedateien im Binärformat (.h5) werden beim Fortsetzen eines Rechenlaufs vorab bereinigt, so dass keine doppelten Zeitschritte enthalten sind.
- neue Option für cpr-Dateien ergänzt: Das Schreiben der cpr-Dateien kann deaktiviert werden oder der Nutzer kann wählen, dass nur die Werte des letzten gültigen Zeitschritts in die Dateien geschrieben werden.

- Randbedingungen: In Randbedingungen, die eine andere Struktur (Pegelpunkt oder Auslauf oder Kontrollquerschnitt) per x- und y-Koordinaten referenzieren, gibt es jetzt die zusätzliche Option diese Struktur per Namen zu referenzieren. Dafür muss der entsprechende Name korrekt vergeben sein. Siehe auch [Kapitel 4.5](#).
- 1D-Bauwerke Erweiterung: Die Option, die Strömungsrichtung in der Abflussbestimmung von 1D-Bauwerken zu berücksichtigen, wurde ergänzt. Siehe [Kapitel 4.5.11](#) und [Kapitel 4.5.10](#).
- Das Einlesen der Dateien feflow-in und sources-in ist nun flexibler. Es ist unter anderem möglich diese Dateien im Binärformat (.h5) einzulesen. Siehe auch [Kapitel 4.6.4](#) und [Kapitel 4.6.6](#).
- Der interne Datenaustausch zwischen dem Präprozessor und den Simulationsprogrammen wurde so verbessert, dass Netze mit maximal 500 Millionen Kanten und entsprechende Anzahlen von Knoten und Elementen berechnet werden könnten. Voraussetzung ist ein Rechner mit hinreichend großem RAM.
- H-Randbedingung kann für Zulauf gebunden an Auslauf genutzt werden. Falls der Abfluss der H-Randbedingung negativ ist, d.h. der H-Rand wirkt als Zufluss, wird der Abfluss für den gebundenen Zulauf auf Null gesetzt. Ist der H-Rand ein Abfluss funktioniert die Kopplung wie gehabt.
- Meldungen des Präprozessors wurden verbessert. Insbesondere gibt es eine neue Fehlermeldung, wenn ein Auslauf mehrmals durch einen Zulauf gebunden an Auslauf referenziert wird.
- Die Module FT, GS1, GSm, ST und WT können nun auch mit dem 2step-Verfahren berechnet werden. Bisher war dies auf das 1step-Verfahren beschränkt.
- **In seltenen Fällen konnte es zu unplausiblen Ergebnisse kommen, wenn der A_{min} -Wert im Vergleich zu den Kontrollvolumen sehr groß gesetzt ist und viele benetzte Kontrollvolumen kleiner als A_{min} sind. Die Behebung dieses Problems hatte allerdings zu Ergebnisunterschieden in anderen Fällen, in denen A_{min} größer null gesetzt ist, geführt und wurde daher in Version 5.2.5 überarbeitet. Siehe Kapitel ??.**
- Beim upgrade von GS-Modellen in Versionen niedriger als 4.4.0 wurden die Einstellungen zum Modul (GS) und die Anzahl der Kornfraktionen nicht richtig übertragen.
- Scripting erweitert: Zugriff (lesend und schreibend) auf Parameter und Reihen der Auslaufrandbedingungen ist jetzt möglich. Siehe auch [Kapitel 8.6](#).
- Scripting verbessert: Wenn die Höhen von Knoten verändert werden, wird jetzt auch das Reibungsgefälle in der Umgebung der Knoten angepasst. Unterschiede in den Berechnungsergebnissen treten voraussichtlich nur für sehr starke Änderungen der Knotenhöhen von nassen Knoten auf. Außerdem wird der Wert für den trockenen Wasserspiegel ggf. angepasst.
- Scripting Fehler behoben: Wenn die Knotenhöhe eines Knotens, der zu einem Zuflussnodestring gehört, per Scripting geändert wurde, konnte es zu falschen Zuflusswerten kommen.
- Ausgabedatei `wspl_max`: Der maximale Wasserspiegel wurde ggf. nicht korrekt bestimmt, wenn sich die Knotenhöhe während der Simulation durch Geschiebeanlandung, Erosion oder durch per Scripting neu gesetzte Z-Werte verändert hat.
- Die Ausgabedateien im ASCII-Format (.dat) haben die zusätzliche Zeile `TIMEUNIT Seconds` im Header.
- Ausgabedatei `q_strg.dat`: Der Header der Datei enthält jetzt auch die SMS-IDs der Nodestrings.
- Ausgabedatei `wehr.dat`: In manchen Fällen wurden nicht die richtigen Nodestrings für die Ausgabe gefunden. Zusätzlich wurde im Header der Datei eine Zeile für die Namen der Nodestrings ergänzt.

- Der Präprozessor löscht beim Start alte pod- und inp-Dateien.
- Die Codemeter-Version in der Auslieferung wurde auf die Version 7.0.0 aktualisiert.
- Die GPU-Version unterstützt Tesla V100 GPUs. Die unterstützten compute capabilities sind damit 3.5 bis 7.0.

10.9 Version 5.1.9 (25.03.2020)

- Randbedingungen *H-Wehr (Steuerung)* und *H-Wehr (Zeit)*: Die Berechnung des unvollkommenen Überfalls aufgrund von Unterwassereinflüssen wurde korrigiert.
- Die Ausgabedatei wehr.dat wurde um eine Spalte erweitert und der Header wurde angepasst und verbessert.
- In Version 5.1.7 und 5.1.8 hatte das Rausschreiben von zusätzlichen SMS-Ausgaben und das Stoppen nach dem nächsten SMS-Zeitschritt mittels der Datei hydro_as-sim.inp nicht richtig funktioniert. Die Funktionen sind nun wieder hergestellt.

10.10 Version 5.1.8 (06.02.2020)

- Problembhebung in den Stofftransportmodulen GS1 und GSm. Siehe Releasenoten im FT-Handbuch.

10.11 Version 5.1.7 (19.12.2019)

- Präprozessor: neue Fehlermeldung, wenn Materialname mehrfach vergeben ist.
- Präprozessor: neue Meldung, wenn Name von Pegelpunkt mehrfach vergeben ist.
- Präprozessor: neue Meldung, wenn Name von Pegelpunkt zu lang ist.
- Präprozessor: Redundanten Meldungen zur Netzstruktur wurden zusammengefasst.
- Präprozessor: Meldungen zu dreiecksförmigem Gitterrand wurden verbessert.
- Präprozessor: Meldungen zu großen und kleinen Innenwinkeln werden nicht mehr für disabled Elemente ausgegeben.
- Präprozessor: In der Version 5.1.6 stürzt der Präprozessor ab, wenn das Netz nicht fortlaufend nummeriert ist (renumber in SMS vergessen). Nun erscheint wieder die Fehlermeldung.
- Präprozessor: Neue Fehlermeldung, wenn Knoten, Element oder Nodestring in der 2dm-Datei keine gültigen IDs (z.B. negative Zahl) besitzen.
- Benutzeroberfläche: Auswahl der Lizenzen um automatische Suche erweitert.
- Scripting erweitert: Zugriff (lesend und schreibend) auf Ausgabezeitschritte (Q_Strg und SMS) ermöglicht.
- Scripting verbessert: Fehlermeldung ergänzt, wenn ungültiger Wert durch ein Skript gesetzt wird.
- In der Ausgabedatei der Zeitschrittweiten waren die Werte in manchen Fällen an einzelnen Knoten zu groß. Die minimale Schrittweite wird nun korrekt ermittelt.

10.12 Version 5.1.6 (28.10.2019)

- Scripting: Handling von nutzerdefinierten Skripten verbessert, siehe [Kapitel 8.7.4](#).
- Scripting: Beim Einlesen von Zuflusszeitreihen durch ein Skript, kann die Zulauftrandbedingung im 2dm-Modell eine leere Reihe enthalten. Bisher führte das zum Programmabbruch.
- Scripting: Die Klasse CulvertSet wurde ergänzt.
- Scripting: Dokumentation verbessert

10.13 Version 5.1.5 (06.09.2019)

- Scripting: Das Abfragen und Setzen der Geschwindigkeitskomponenten V_x und V_y war vertauscht.
- GUI: Die GPU-Version konnten nicht aus der GUI gestartet werden.
- Scripting: Dokumentation - insbesondere zum LUA-Zusatzmodul *lfs* - verbessert.

10.14 Version 5.1.4 (08.08.2019)

- Die Abflussbestimmung für 1D-Durchlässe wurde verbessert: Wenn ein Durchlass entgegen seiner Neigung durchflossen wurden, wurde der Abfluss unterschätzt. In Extremfällen war der Abfluss Null.
Für Durchlässe, die in Richtung der Neigung durchflossen werden, gibt es keine Änderung.

10.15 Version 5.1.3 (19.07.2019)

- Fehler im Präprozessor behoben: Wenn ein Modell nicht renumbered wurde, kam es in der Version 5.1.2 zum Programmabbruch.

10.16 Version 5.1.2 (12.07.2019)

- Aus der Datei *wtiefe_0.dat* eingelesene negative Werte für Anfangswassertiefen werden nun programmintern auf Null gesetzt. Eine entsprechende Warnung wird ausgegeben. Bisher wurden diese Knoten intern mit negativen Werten belegt, so dass kein Abfluss aus ihnen statt fand.
- Eine unnötig restriktive Prüfung der Eingaben für die Auslauftrandbedingungen H-Wehr (Zeit) und H-Wehr (Steuerung) im Zusammenhang mit unvollkommenem Überfall wurde angepasst.
- Die Prüfung auf singuläre Knoten wurde verbessert. Ein Knoten wird jetzt auch als singulärer Knoten erkannt, wenn alle angrenzenden Elemente disabled sind.
- Die GPU-Version wurde stabilisiert. In seltenen Fällen konnte es zu Speicherzugriffsfehlern oder Berechnungsfehlern kommen. Im Fall der Berechnungsfehler waren die Ergebnisse aber offensichtlich unplausibel.
- Die Stofftransportberechnung wurde für Unix-Systeme verbessert. In seltenen Fällen konnte es zu Problemen beim Schreiben der Ergebnisdateien kommen.

10.17 Version 5.1.1 (24.06.2019)

- Einlesen der Datei nodesources.dat verbessert: Fehlermeldungen bei fehlerhafter Formatierung wurden ergänzt.
- Neue Warnung: Wenn die interne Zeitschrittweite kleiner als 10^{-6} Sekunden ist, wird eine Warnung ausgegeben und die SMS-Ergebnisse zu diesem Zeitpunkt werden in separaten Dateien ausgegeben. Insbesondere die Datei timesteps_<Zeit>.dat kann zur Analyse des Modells genutzt werden.

10.18 Version 5.1.0 (08.05.2019)

- Die Windows-Version ist nun als 64-bit-Version verfügbar. Für die Vollversionen (Windows und Linux) ist die Knotenanzahl nun unbegrenzt (nur noch abhängig vom Hauptspeicher).
- Die Unterstützung von GPUs wurde an neue Anforderungen angepasst. Hinweise finden Sie in der IT-Dokumentation.
- Prüfeinstellungen für den Präprozessor wurden erweitert. Die Auswahl der Parameter wird außerdem in der 2dm-Datei abgespeichert, siehe [Kapitel 6.5](#).
- Die Methode der wassertiefenabhängigen k_{St} -Werte wurde verbessert und die Performance gesteigert. Insbesondere ist die Anzahl der Stützstellen frei.
- Ausgabe in q_strg.dat bei mehr als 6000 Nodestrings wurde korrigiert.
- Scripting: konstante Quellterme können abgefragt und gesetzt werden.
- Der Modellname wird in den h5-Ausgabedateien angegeben.
- Performance verbessert.
- Weitere Felddimensionen wurden aufgehoben; geblieben ist: Maximale Anzahl der Stützstellen einer Niederschlagszeitreihe (1000).

10.19 Version 5.0.2 (14.12.2018)

In diesem Release wurden neben der Fehlerbeseitigung auch Anforderungen umgesetzt, die beim Anwendertreffen vom 13. November 2018 formuliert wurden.

- Fehler bei Geschiebeeintrag mit mehreren Stoffeintrags-Zeitreihen behoben.
- GUI des Präprozessors beim Umgang mit vielen Warnungen verbessert: Warnungen gleichen Typs werden nun gruppiert und sind einfach auszublenden. Die Liste der Hinweise ist nun auch bei sehr vielen Warnungen aussagekräftig.
- Die GUI des Präprozessors speichert nun die Fensterposition und Größen der einzelnen GUI-Elemente, so dass einmal getroffene Einstellungen erhalten bleiben und nicht immer neu vorgenommen werden müssen.
- Beim Upgrade der 2DM-Dateien auf Version 5.0.1 kam es zu Fehlern, wenn diese neuen 2dm-Dateien anschließend mit SMS gespeichert wurden: SMS speicherte Randbedingungsinformationen nicht mehr ab. (Problem resultierte aus einem SMS-Fehler. HYDRO_AS-2D 5.0.2 wurde so verändert, dass der SMS-Fehler nicht mehr auftritt. Zeitgleich wurde in SMS 12.3.5 der Fehler behoben.)
- Bei sehr (i.e. unplausibel) großen Durchlässen kam es zu Fehlern in der Berechnung. Diese äußerten sich in NaN-Ausgaben für den Abfluss am Durchlass.

- Bei der Deinstallation im „Silent-Modus“ erschienen Dialogabfragen. Der „Silent-Modus“ läuft nun ohne Rückfragen durch.
- Es kam zu einem Fehler beim Schreiben der binären Ausgabe für Geschwindigkeiten, wenn kein Dateiname angegeben wurde. Die Ausgabe für Geschwindigkeiten findet nun nicht mehr statt, wenn der Dateiname „-“ ist.
- Die Gesamtzeit der Simulation ist nun in der GUI einstellbar
- Beim Konvertieren alter Dateien wurde ein falscher Defaultwert bei Zufluss *Winkel vorgeben* gesetzt.
- Die mitgelieferte Batchdatei zum Aufruf des Programmes bricht nun ab, wenn der Präprozessor einen Fehler gefunden hat. Wenn kein Fehler gefunden wurde, läuft sie nun ohne erforderliche Tastatureingaben durch.
- Einige Fehlermeldungen wurden präziser formuliert.
- Wenn in der `hydro_as-2d.inp`-Datei *Rechenlauf fortsetzen* aktiviert wurde und keine Anfangsbedingungsdateien vorhanden sind, wird nun ohne Anfangsbedingungen simuliert. Dieses Merkmal ist nicht im Zusammenhang mit der GUI nutzbar.
- Bei unzuverlässigen Netzwerkverbindungen zum Lizenzserver wird nun vor einem Programmabbruch häufiger und in größerem zeitlichen Abstand geprüft, ob eine Verbindung wieder aufgebaut werden kann. (Grundsätzlich muss der Lizenzserver aber weiterhin verfügbar sein.)

10.20 Version 5.0.1 (08.11.2018)

- Übernehmen der Sohlhöhen aus dem Netz für 1D-Bauwerke funktioniert nun richtig.
- Datei `nodeniederschlag.dat` für die Zuordnung der Niederschlagszeitreihen zu den Knoten wurde nicht gelesen, wenn die Werte Nachkommastellen „0“ enthielten.
- Werte für den SMS-Zeitschritt und den `Q_strg`-Zeitschritt können in der GUI auf beliebige Werte gesetzt werden.
- Verbesserung und Korrekturen von Meldungen.
- In W-Q-Beziehungen von Abflussrandbedingungen wird nun überprüft, dass die WSP-Werte streng monoton steigend sind. Falls dies nicht der Fall ist, wird eine Fehlermeldung erzeugt und der Fehler muss behoben werden.
- Scripting: Korrektur der Funktion `Node.new`: Ein Node kann jetzt auch per ID angelegt werden.
- Scripting: Diverse neue Funktionen. Insbesondere zum Zugriff auf den `kst`-Wert und zum Schreiben von SMS-Dat-Dateien.
- Scripting: Wenn Zustandswerte von HYDRO_AS-2D in der `close`-Methode abfragt wurden, kam es zu einem Absturz.

10.21 Version 5.0.0 (24.10.2018)

- Der Präprozessor wurde in Bezug auf Geschwindigkeit, Fehler-Meldungen, Warnungen und Informationen zum Modell hin verbessert und beinhaltet zusätzliche Prüfungen.
- Maximale Anzahl von Nodestrings, Knoten in Nodestrings, Stützstellen von Zuflussganglinien, und W/Q-Beziehungen wurde aufgehoben. Die Größen sind nicht mehr limitiert.

- Neue Funktionalität „Steuerung mit Skripten“ wurde ergänzt. Skripte können genutzt werden um z. B. Bauwerke zu steuern. Informationen finden Sie in [Kapitel 8](#).
- Die Berechnungsmethoden für 1D-Wehre und 1D-Durchlässe wurden überarbeitet und verbessert. In einigen Fällen wird die Leistungsfähigkeit der Durchlässe nun höher eingestuft. Die Berechnungsmethoden sind in [Kapitel 2.2](#) beschrieben.
- Die Berechnung des Abflusses an Kontrollquerschnitten ist nun genauer und robuster. Informationen finden Sie in [Kapitel 4.5.12](#).
- Die Ausgaben in der Datei „bw_temp.dat“ werden nach den Nodestring-Namen sortiert.
- Die Zeitangaben in den binären Ausgabedateien (.h5), sowie die Zeitpunkte der maximalen Wassertiefe und maximalen Geschwindigkeit werden auf ganze Sekunden gerundet.

10.22 Version 4.4.7 (25.06.2018)

- Maximale Gesamtanzahl der Knoten aller Nodestring auf 3.000.000 erhöht.
- Weitere Release-Notes für die Zusatzmodule Stofftransport im FT-Handbuch.

10.23 Version 4.4.6 (13.06.2018)

- Warnungen zum Ablauf von Demolizenzen werden drei Tage vor Ablaufdatum aktiviert.
- Fehlermeldung ergänzt: Anfangswassertiefen dürfen nicht negativ sein.

10.24 Version 4.4.5 (17.05.2018)

- Anpassung in 4.4.4 korrigiert. Maximale Anzahl der Nodestrings ist nun auf 6.000 erhöht.

10.25 Version 4.4.4 (03.05.2018)

- Maximale Anzahl Nodestrings von 2.000 auf 6.000 erhöht. (fehlerhafte Änderung)

10.26 Version 4.4.3 (16.03.2018)

- Die Zeitschritte in den binären SMS- Ausgabedateien (.h5) sind nun an die Zeitschritte in den ASCII-Dateien (.dat) angepasst.
- Problem bei fehlender Ganglinie behoben. Fehlt die Definition der Ganglinie zu einer Nodestring-Randbedingung in SMS, wird der Abfluss Null angenommen. Eine entsprechende Warnung wird ausgegeben.

10.27 Version 4.4.2 (24.01.2018)

- Problem beim Anlegen binärer Ausgabedateien mit sehr langen Dateinamen behoben. Sehr lange Dateinamen (absoluter Pfad) konnten unter Linux zu Programmabbrüchen führen.

10.28 Version 4.4.1 (19.01.2018)

- In einzelnen Fällen konnten sehr schmale Elemente mit starkem Gefälle zu Fehlern (Wassertiefe = NaN) in der Berechnung führen. Die Berechnungsmethode wurde robuster gemacht und für Extremfälle eine explizite Fehlermeldung ergänzt.

10.29 Version 4.4.0 (06.12.2017)

- Mit zusätzlichen Lizenzen sind die Module FT, ST, GS1, GSm, und WT nun mit der aktuellen HYDRO_AS-2D-Version aktivierbar.
- Es gibt eine neue Auslaufrandbedingung vom Typ „Abflusszeitreihe“. Diese Randbedingung ist einem negativen Zufluss vorzuziehen, da sie höchstens das vorhandene Wasservolumen dem Modell entzieht, aber nie mehr, siehe [Kapitel 4.5.8](#).
- Für die 1D-Bauwerke können nun auch Namen vergeben werden.
- Für Pegelpunkte können nun auch Namen vergeben werden.
- Die Datei sources-in.dat kann vor dem Simulationsende enden. Die Werte des letzten angegebenen Zeitschritts können beliebig sein. Bisher war es notwendig, dass die letzten Werte Null sind, siehe [Kapitel 4.6.6](#).
- Bereiche, in denen sowohl die Fließgeschwindigkeiten als auch das Wasserspiegelgefälle sehr klein sind, werden nicht mehr zur internen Zeitschrittberechnung hinzugezogen. Dadurch können die Zeitschritte ggf. etwas größer und die Rechenzeiten entsprechend kürzer sein.
- Die Dateien q_strg.dat und pegel.dat wurden auch dann erstellt, wenn die Namen der Dateien in SMS gelöscht wurden. Die Dateien werden in diesem Fall nun nicht mehr angelegt.
- Für die Ausgabe der Abflüsse in der q_strg.dat können optional sechs Nachkommastellen gewählt werden.
- Im Verzeichnis SMS-2dm-Vorlage liegt ein Ordner „Data-in“, der Eingabedateien mit den vorgesehenen Dateinamen enthält. Diese Dateien können benutzt werden, um Eingaben mit den richtigen Dateinamen zu erstellen.
- Der Parameter SCF wird nicht mehr verwendet, da die Parallelisierung und verbesserte Berechnung der Zeitschrittweite Beschleunigungen der Rechenzeit bewirken.

10.30 Version 4.3.4 (05.09.2017)

- Weiteres Problem im Zusammenhang mit Konvertierung und Speichern in SMS behoben: Wenn Zufluss-Randbedingungen nach dem Konvertieren bearbeitet wurden, hat SMS den Fehler „Value (\"") cannot be set. It is not in the list of choices“ gemeldet und beim Speichern der 2dm-Datei wurden Randbedingungen gelöscht.
Der Fehler betrifft nicht Dateien, die für HYDRO_AS 4.2 erstellt wurden, sondern nur ältere Versionen.

10.31 Version 4.3.3 (28.08.2017)

- In manchen Fällen brach der Präprozessor mit einem Hinweis auf vorhandene Fehler ab („Es wurden 1 Fehler gefunden“), ohne dass eine Fehlerbeschreibung auf die Standardausgabe oder in der „hydro_2dm.mel“-Datei erzeugt wurde.

10.32 Version 4.3.2 (24.08.2017)

- Problem im Zusammenhang mit dem Softwareschutz gelöst: Ab der Version 4.3.0 wurde jeweils eine Lizenz zu viel ausgecheckt.

10.33 Version 4.3.1 (18.08.2017)

- Problem im Zusammenhang mit Konvertierung und Speichern in SMS behoben. Das Problem trat nur bei Speichern in SMS nach Konvertierung auf Version 4.3. von Modellen auf, die älter sind als Version 4.2.0 und 1D-Wehrüberfälle und/oder 1D-Durchlässe enthalten.

10.34 Version 4.3.0 (08.08.2017)

- Erweiterung um neuen Quellterm-Typ (Effektivniederschlag) pro Knoten und für beliebige Zeitschritte (vgl. [Kapitel 4.6.6](#)).
- In SMS können nicht benötigte Ausgabedateien durch Löschen des Namens in den „Global Parameters“ deaktiviert werden.
- Neue Ausgabedateien: Quellterme, maximale Geschwindigkeit, Zeitpunkt der maximalen Geschwindigkeit und maximale Wassertiefe (vgl. [Kapitel 5.7](#)).
- Anwendung des „2dm-upgrades“ wurde vereinfacht/verbessert. Hinweise dazu befinden sich in der IT-Dokumentation_HYDRO_AS-2D.pdf. Für die Version 4.3.0 ist das Upgrade erforderlich.
- Namen (max. 12 Zeichen) können für Zulauf- und Auslaufnodestrings sowie für Kontrollquerschnitte in SMS angegeben werden. Ausgabe in q_strg.dat wird nach Namen sortiert (vgl. [Kapitel 4.5.13](#)).
- Neue Option für „Rechenlauf fortsetzen?“ in SMS: Automatisch beim letzten gültigen Zeitschritt fortsetzen ohne Abfrage der Zeitschrittnummer.
- Neue Ausgabe des Präprozessors für Modellqualitätsprüfung: Statistik zu Amin, Winkeln und KUKs.
- Die Datei hydro_as-2d.cpr hat sich verändert und neue cpr-Dateien sind hinzugekommen. Zum Fortsetzen eines Rechenlaufs müssen die Dateien in der aktuellen Version vorhanden sein.
- Feature: wspl_max, schub_max etc. nur für einen zeitlichen Abschnitt der Simulation bestimmen (vgl. [Kapitel 5.7](#)).
- Konvention für Dateinamen wurde eingeführt, weil Linux-Systeme Case-sensitive sind: Alle Dateinamen (inclusive Endungen) werden mit kleinen Buchstaben geschrieben.
- Wassertiefen niedriger als hmin in der wtiefe_0.dat werden unverändert übernommen. Bisher wurden diese Werte nach dem Einlesen intern auf 0 gesetzt.
- Modelle ohne Zulauf- und Auslaufnodestrings sind nun möglich, wenn Quellterme definiert sind.
- Fehlermeldung und Simulationsstopp bei falscher Zuordnung der Knoten-Randbedingung Zufluss/Abfluss ergänzt.
- SSE4.1 ist keine Voraussetzung für die Prozessoren mehr.

10.35 Version 4.2.7 (26.06.2017)

- Behebung eines selten im Zusammenhang mit vielen 1D-Bauwerken auftretenden Programmabbruchs.
- Behebung eines selten unter Linux auftretenden Programmabbruchs im Präprozessor.

10.36 Version 4.2.6 (15.05.2017)

- Der Maximalabfluss bei 1D-Elementen wird in beiden Fließrichtungen berücksichtigt. Bisher wurde der Abfluss von z1 zu z2 begrenzt, d.h. für 1D-Elemente, die in Fließrichtung definiert sind, gibt es keine Änderungen.
- In der Ausgabedatei Kst_Gesamt werden die Rauheiten für disabled-Elemente auf 9999 gesetzt (analog zu strickler.dat).

10.37 Version 4.2.5 (04.04.2017)

- Linux-Version unterstützt nun bis zu 70 Millionen Knoten. Windows-Version ist unverändert.

10.38 Version 4.2.4 (08.12.2016)

- Binäre Ausgabedateien können nun auch mit der Linux-Version geschrieben werden.
- Die Linux-Version kann mit 7 Mio. (statt 3,5 Mio.) Knoten umgehen.
- Die Randbedingung „H-Rand“ wurde stabilisiert.
- Fehler in der WQ-Beziehung, der nur bei negativen Geländehöhen auftritt, wurde behoben.
- Das Format der Datei „Kst_Gesamt“ wurde verändert. Bei konstanten Kst-Werten, werden die Ergebnisse nur einmal und nicht mehr für jeden SMS-Ausgabezeitschritt geschrieben.
- Selten auftretender Fehler in der Ausgabe der Simulationszeit wurde behoben.
- Texte der Fehlermeldungen wurden verbessert.

10.39 Version 4.2.3 (02.11.2016)

- Das Skript upgrade2dm.py wurde erweitert, um Warnungen in SMS zu vermeiden.
- Texte der Fehlermeldungen wurden verbessert.
- Das Ausgabeformat in der Datei „max_time“ wurde verbessert.

10.40 Version 4.2.2 (11.10.2016)

- Fehler in der Ausgabe von binären Dateien bei sehr großen Netzen wurde behoben.
- Neue Fehlermeldung zu fehlerhaft definierten Nodestings wurde hinzugefügt.
- Neue Fehlermeldung wird ausgegeben, wenn der Rechenlauf fortgesetzt wird, aber die Daten des alten Rechenlaufs (CPR-Datei, POD-Datei, usw.) fehlen.
- Fehler in der Vorlagedatei in der Definition der Wehrgeschwindigkeit wurde behoben.
- Format der Ausgabedatei „Wehr.dat“ wurde verbessert.
- Das Skript upgrade2dm.py wurde erweitert, um Warnungen in SMS zu vermeiden.

10.41 Version 4.2.1 (01.09.2016)

- Das Skript upgrade2dm.py wurde erweitert, um Probleme bei der Bearbeitung und Abspeicherung von Zuflüssen in SMS zu vermeiden.
- Behebung eines selten auftretenden Fehlers in Convert2DM.exe.

10.42 Version 4.2.0 (22.08.2016)

- Die SMS-Ausgabedateien können unter Windows sowohl im ASCII-Format, als auch im XMDF-Format (binär) erstellt werden.
- Eine räumliche Analyse der internen Zeitschrittweite ist durch die neue Ausgabedatei „timesteps“ gegeben.
- Es gibt die neuen Auslauf-Randbedingungen H-Wehr (Zeit) und H-Wehr (Steuerung).
- 1D-Durchlässe können mit vorgegebener Richtung berechnet werden (Siel) D
- Die Zeitpunkte des maximalen Wasserstands werden für jeden Knoten in der neuen Datei „max_time“ ausgegeben. Da die aktuellen Werte in die CPR-Datei geschrieben werden, können Rechenläufe aus älteren Versionen nicht fortgesetzt werden. Die CPR-Datei muss mit der aktuellen Version neu erstellt werden.
- Zusätzliche Ausgaben, die durch die Steuerungsdatei „hydro_as-sim.INP“ angefordert werden, werden in zusätzlich angelegte Dateien geschrieben. Die Dateinamen enthalten dabei den Zeitpunkt.
- Die Zufluss- und Auslauf-Randbedingungen wurden stabilisiert.
- Die Nodestings für einen Zulauf gebunden an Auslauf können unterschiedliche Knotenanzahlen aufweisen.
- Einige Hilfsprogramme werden nicht mehr ausgeliefert, da sie nur für SMS10 gültig sind.

10.43 Version 4.1.6 (23.04.2016)

- Fehler bei der Zuordnung einer Zufluss-/Abflussreihe in NodeSource.dat-Datei zu mehreren Schächten/ Regenüberläufen behoben.

10.44 Version 4.1.5 (13.04.2016)

- Weiteres Problem bei Abstürzen mit vielen Threads behoben.
- Tool „Modellausschnitt“ unter Hilfsprogramme ergänzt.
- Für Amin kann in SMS der Wert Null angegeben werden. Das bedeutet, Kontrollvolumina werden nicht angepasst.

10.45 Version 4.1.4 (11.02.2016)

- Problem bei Abstürzen mit vielen Threads behoben.
- Ausgabe in SMS-Dateien teilweise mit mehr Nachkommastellen.
- Erweiterung: Eine Zufluss-/Abflussreihe in NodeSource.dat-Datei kann nun mehreren Schächten/ Regenüberläufen zugeordnet werden.

10.46 Version 4.1.3 (26.01.2016)

- Konverter für 2dm-Dateien erzeugte keine Vorlage mit H-Randbedingungen.

10.47 Version 4.1.2 (07.01.2016)

- Sporadische Abstürze beim Aufruf des Präprozessors und bei Simulationen mit vielen Threads behoben.

10.48 Version 4.1.1 (15.10.2015)

- Fehlermeldung im Umfeld von falsch definierten Nodestrings ergänzt. (Bauwerks-Nodestrings mit mehr als zwei Knoten, Kontrollquerschnitte in deaktivierten Bereichen).
- Fehlerhaften Batch-Job korrigiert (Tippfehler bei "OMP_NUM_THREADS").
- Bei Knoten in deaktivierten Bereichen wurden NaN-Geschwindigkeiten in die SMS-Ergebnis-Dateien geschrieben. Dies verhinderte die direkte Nutzung dieser Dateien in SMS.

10.49 Version 4.1.0 (15.10.2015)

- HYDRO_AS-2D kann mehrere Prozessorkerne (Cores) deutlich besser nutzen. Zur vollen Nutzung vieler Prozessorkerne ist eine entsprechende Lizenzierung erforderlich.
- Eine neue Randbedingung für Nodestrings wurde eingeführt (Dirichlet-Randbedingung): Wasserstände können jetzt zeitlich variabel vorgegeben werden. Damit sind beispielsweise Tide-Randbedingungen einfach abzubilden (vgl. [Kapitel 4.5.9](#)).
- Speziell zur einfacheren Abbildung von Starkregenereignissen werden Niederschläge sowie eine Anbindung an Kanalsysteme (Schächte, Regenüberläufe) unterstützt (vgl. [Kapitel 4.6.3](#) und [Kapitel 4.6.5](#)).
- Das Programm upgrade_2dm.bat wurde ergänzt: Es verändert eine bestehende 2dm-Datei so, dass die neuen Merkmale (Niederschlag, H-Randbedingungen, ...) in SMS bearbeitet werden können.
- Das Verfahren zur Fortsetzung der Wasserspiegellagen an den trocken/nass-Grenzen wurde überarbeitet.
- Anwender können mit einer speziellen Steuerdatei während einer Simulation HYDRO_AS-2D kontrolliert abbrechen bzw. für jeden internen Rechenzeitschritt zusätzliche Informationen anfordern (vgl. [Kapitel 5.9](#)).
- Für die SMS-Ausgabe werden die Zeitschritte auf Sekunden gerundet. Damit können Ausgaben aus verschiedenen Rechenläufen nun in SMS miteinander kombiniert werden.
- Die Ausgabedateien können nun vom Anwender in SMS frei benannt werden.
- Die Datei VOLUMEN.dat wird mit 4 statt 3 Nachkommastellen (cm²) rausgeschrieben.
- Infolge von Verbesserungen der Methoden beim Fortsetzen eines Rechenlaufs hat sich das Format der Datei HYDRO_AS-2D.CPR geändert. Dadurch können CPR-Dateien, die aus einer älteren Version stammen, nicht mehr mit der Version 4.1.0 weiterbenutzt werden.
- HYDRO_AS-2D ist in einer Linux-Version verfügbar.
- Das Programm Check2DM wurde ergänzt. Dieses Programm führt erweiterte Konsistenzprüfungen auf den 2dm-Dateien durch.
- Bei Berechnungen mit Geländehöhe auf oder unter Meereshöhe wurden falsche Wasserspiegellagen in die Dateien WSPL und WSPL-MAX geschrieben.

- Falls die Konstruktionsunterkante gleich der Geländehöhe ist, wird eine Fehlermeldung ausgegeben.
- Fehlermeldungen an Auslaufrandbedingungen sind verbessert.

10.50 Version 4.0.4 (30.07.2015)

- Berechnungsabbruch bei Simulationszeiträumen von mehr als 3 Tagen (genauer: 262.144 Sekunden) behoben.

10.51 Version 4.0.3 (23.07.2015)

- Fehler bei der Behandlung höhenabhängiger Rauheiten behoben.

10.52 Version 4.0.2 (18.05.2015)

- Beim Rechnen mit mehreren parallelen Prozessoren kam es in manchen Fällen zu falschen Berechnungen. Diese Fehler äußerten sich in unplausiblen Oszillationen.

10.53 Version 4.0.1 (23.04.2015)

- Wenn in einem Modell keine Nodestings (d.h. keine Zulauf- oder Auslaufrandbedingungen, Durchlässe etc.) definiert wurden, stürzte der HYDRO_AS-Präprozessor ab. Dieses Problem wurde behoben.
- Wenn die Gesamtanzahl aller Knoten in allen Nodestings (Aktuelle Version: 10.000) überschritten wird, erfolgt eine Fehlermeldung. Bisher wurden die überzähligen Knoten ohne Meldung ignoriert.
- In einigen Fällen kam es bei großen Modellen im HYDRO_AS-Präprozessor zu einem Überlauf des Programmspeichers.

10.54 Version 4.0.0 (10.02.2015)

Die Version 4.0.0 stellt für HYDRO_AS-2D einen Meilenstein dar. Sie ist die erste gemeinsam von Dr. Nujić und Hydrotec entwickelte Version.

Die Entwicklung sämtlicher HYDRO_AS-Produkte erfolgt fortan durch Hydrotec.

- HYDRO_AS-2D wird nun in zwei Varianten ausgeliefert: Neben der CPU-Variante, die mit früheren Releases vergleichbar ist, wird nun eine GPU-Variante angeboten, die bei großen Netzen deutlich schneller rechnet. Die beiden Varianten werden jeweils ab der Version 4.0.0 ausgeliefert.
- Die Anwenderdokumentation wurde überarbeitet, vereinheitlicht und um das Kapitel Release-Notes erweitert.
- Der Softwareschutz wurde umgestellt, sodass auch Software-Keys möglich sind.
- Das Setup-Programm legt Dateien in einer veränderten Struktur ab.
- Die Fortschrittsanzeige wurde leicht verändert: Sie nennt auch die aktuellen Uhrzeiten, damit Abschätzungen über den Rechenfortschritt einfacher möglich sind.
- Die Fortschrittsanzeige der Konsole wird zusätzlich auf die „HYDRO_2DM.MEL“ geschrieben.

- Für Auslaufrandbedingungen, die nicht am Rand liegen wird keine Energieerhaltung berechnet. In früheren Versionen war in solchen Fällen die Richtung für die Energieerhaltung nicht korrekt. Die Datei „HYDRO_2DM.MEL“ enthält einen entsprechenden Hinweis.
- Falls ein Nodestring mehr als die für HYDRO_AS-2D zulässige Anzahl von 500 Knoten enthält, wird eine Fehlermeldung ausgegeben.
- Falls ein Nodestring weniger als 2 Punkte enthält, wird eine Fehlermeldung ausgegeben. Insbesondere wurden Bauwerke (wie Durchlässe) in solchen Situationen ggfs. falsch berechnet.
- Der Datei „HYDRO_2DM.MEL“ ist zu entnehmen, mit welcher HYDRO_AS-Version gerechnet wurde.
- Die Datei „KSt_Sohle.dat“ wird nicht mehr erzeugt. Sie enthielt lediglich die Daten, die auch in SMS eingegeben wurden.
- Der HYDRO_AS-Rechenkern belegt nur noch so viel Speicherplatz wie tatsächlich benötigt wird und nicht mehr Platz entsprechend der maximal zulässigen Netzgröße.
- In manchen Fällen wurden in der Version 3.x bei gebundenen Abfluss/Zufluss-Bindungen fehlerhafte Knoten zugeordnet. Dieses Problem wurde gelöst.
- In der Datei Q_strg.dat werden Abflusswerte mit 13 statt mit 10 Stellen rausgeschrieben.
- Es gibt keine verschiedenen Programme (exe-Dateien) für unterschiedliche Knotenzahlen. Diese Information für die maximal zulässige Knotenzahl wird dem Dongle entnommen.

Literatur

- [1] Abbott, M.B. (1979): Computational Hydraulics, Elements of the Theory of Free-Surface Flows, Pitman Publ., London.
- [2] ASCE TASK COMMITTEE (1988): Turbulence Models in Hydraulic Computation, Jour. of Hydr. Eng., Spec. Issue, Vol. 114, No. 9.
- [3] Bechteler, W.; Kulisch, H.; Nujić, M. (1992): 2-D Flooding Waves - Comparison between Experimental and Calculated Results, 3rd Int. Conf. on Flood and Flood Management, Florence, 24-26 November 1992.
- [4] Bechteler, W.; Nujić, M.; Otto, J.A. (1994): An Analysis of Flood Propagation using Program Package FLOODSIM, Speciality Conference on Modelling of Flood Propagation over Initially Dry Areas, 29-30 June 1994, Milan, Italy.
- [5] Bechteler, W.; Nujić, M. (1997): 2-D morphologische Simulation einer Flußaufweitung, Darmstädter wasserbauliches Kolloquium, Mitteilungsheft Nr. 98 des Instituts für Wasserbau und Wasserwirtschaft, TH Darmstadt.
- [6] Bechteler, W.; Nujić, M. (1998): Predicting Reservoir Sedimentation with 2d-Model FLOODSIM, Int. Journal of Sediment Research, Vol. 13, No. 1.
- [7] Preißler, G.; Bollrich, G. (1985): Technische Hydromechanik Band1, Verlag für Bauwesen, Berlin.
- [8] BWK (1999): BWK-Merkblatt 1 - Hydraulische Berechnung von Fließgewässern, Teil 1 – Stationäre Berechnung der Wasserspiegellinie unter besonderer Berücksichtigung von Bewuchs und Bauwerkseinflüssen, Düsseldorf.
- [9] Di Giammarco, P.; Todini, E. (1994): A Control Volume Finite Element Method for the Solution of 2-D Overland Flow Problems, Int. Conf. on Modelling of Flood Propagation over Initially Dry Areas, 29-30 June 1994, Milan, Italy.
- [10] DVWK (2000): DVWK Schriften - Heft Nr. 127: Numerische Modelle von Flüssen, Seen und Küstengewässern, Bonn.
- [11] Fischer, H.B.; List, E.J.; Imberger, R.C.Y., et al. (1979): Mixing in inland and coastal waters, Academic Press, New York.
- [12] Hydretec;(2020): Benutzerhandbuch HYDRO_FT-2D Erweiterung zu HYDRO_AS-2D zur Simulation des Stofftransports, Version 5.3.0, Aachen, Deutschland.
- [13] Hydretec; (2020): Benutzerhandbuch HYDRO_WT-2D Erweiterung zu HYDRO_AS-2D zur Simulation des Wärmetransports, Version 5.3.0, Aachen., Deutschland
- [14] Hydretec; Nujić, M. (2018): LASER_AS-2D Erweiterung zu HYDRO_AS-2D zur Ausdünnung und Aufbereitung von Laserscandaten für die 2D-Modellierung, Benutzerhandbuch Version 2.0, Aachen.
- [15] Hydretec (2015): JabPlot, Benutzerdokumentation Version 3.1, Aachen.
- [16] Kölling, C. (1994): Finite - Elemente Simulation der Geschwindigkeitsverteilung in Kanälen und teilgefüllten Rohrleitungen, Mitteilung des Lehrstuhls für Hydraulik und Gewässerkunde der Technischen Universität München, Nr. 60.
- [17] Lafon, F.; Osher, S. (1991): High Order Filtering Methods for Approximating Hyperbolic Systems of Conservation Laws, Jour. of Comp. Phys. 96, pp. 110-142.

- [18] Lauber, G.; Hager, W. H. (1998): Experiments to dambreak wave: Sloping channel, J. Hydr. Research, Vol. 36, No. 5.
- [19] Li, C.W.; Falconer, R.A. (1995): Depth Integrated Modelling of Tide Induced Circulation in a Square Harbour, Jour. of Hydr. Res., Vol. 33, No. 3, pp. 321-332.
- [20] Molinaro, P. (1992): A Review of 2-D Mathematical Models for the Simulation of Flood Propagation on Dry Bed, Hydraulic Engineering Software IV, Fluid Flow Modelling, Valencia, Spain, Computational Mechanics Publications, Elsevier Applied Science, pp. 431-443.
- [21] Naudascher, E. (1992): Hydraulik der Gerinne und Gerinnebauwerke, Springer-Verlag, Wien New York.
- [22] Nujić, M. (1995): Efficient Implementation of Non-oscillatory Schemes for the Computation of Free-surface Flows, Journal of Hydraulic Research, 33(1).
- [23] Nujić, M. (1996): Discussion of the paper 'Finite Volume Two-dimensional Unsteady-flow Model for River Basins' presented by D.H. Zhao, H.W. Shen, G.Q. Tabios III, J.S. Lai and W.Y. Tan, J. Hydr. Engrg. ASCE, Nr.1.
- [24] Nujić, M. (1997): Dam-break Flood-wave Propagation using the Program Package FLOODSIM, Edt. Hiver/ Zech, CADAM Workshop, Brüssel, 21-22 June.
- [25] Nujić, M. (1999): Praktischer Einsatz eines hochgenauen Verfahrens für die Berechnung von tiefengemittelten Strömungen, Mitteilungen des Instituts für Wasserwesen der Universität der Bundeswehr München, Nr. 64.
- [26] Nujić, M. (2003): ÖWAV-Seminar, 26.-27. Februar 2003; Fließgewässermodellierung – von der Einzur Mehrdimensionalität, BOKU Wien.
- [27] Nujić, M.; Elsner, T.; Eidelsburger, W. (2006): Schwebstoffverlandung in den Donauvorländern unterhalb der Staustufe Vohburg, Wasserwirtschaft 3.
- [28] Patt, H. (Hrsg.) (2001): Hochwasser – Handbuch, Auswirkungen und Schutz, mit Beiträgen von: W. Bechteler, H. Brombach, R. Dillmann, K.-D. Fröhlich, P. Jürging, W. Kron, O. Niekamp, M. Nujić, H. Patt, W. Richwien, K.-H. Rother, G. Vogel, R. Vogt, Springer.
- [29] Pironneau, O. (1989): Finite Element Methods for Fluids, Masson, Paris.
- [30] Schmutz, M. (2003): Eigendynamische Aufweitung in einer geraden Gewässerstrecke. Mitteilung des Lehrstuhls für Hydraulik und Gewässerkunde der Technischen Universität München, Nr. 96.
- [31] Schröder, R.C.M. (1990): Hydraulische Methoden zur Erfassung von Rauheiten, DVWK Schriften, Heft 92, Hamburg-Berlin, Parey.
- [32] Stephan, U.; Nujić, M. (2005): Physikalische und numerische Strömungsmodellierung – Vorteile und Grenzen, Wasser & Abfall, Okt. / Nov.
- [33] Tan, W. (1992): Shallow Water Hydrodynamics, Elsevier, Amsterdam.
- [34] Yörük, A.; Sacher, H. (2014): Methoden und Qualität von Modellrechnungen für HWGefahrenflächen. In: Stamm, J.; Graw, K.-U. (Hrsg.): Simulationsverfahren und Modelle für Wasserbau und Wasserwirtschaft. 37. Dresdner Wasserbaukolloquium 2014. Dresden, 13./14. März 2014 (Dresdner Wasserbauliche Mitteilungen 50), S. 55 – 64.
- [35] Yörük, A. (2009): Unsicherheiten bei der hydrodynamischen Modellierung von Überschwemmungsgebieten (Dissertation), Mitteilungen des Instituts für Wasserwesen der Universität der Bundeswehr München, Heft 99, München.