



# TUFLOW Classic and HPC

## 2020-01 and 2020-10 Release Notes

### Document Updates and Important Notices

(in chronological order)

**Feb 10, 2020:** The 2020-01-AA Build is a major release that includes several new industry leading features built into the TUFLOW HPC 2D solver. There are also a range of general new features, enhancements, and minor bug fixes, including much faster start-ups for large 1D network models. TUFLOW Classic results should be unchanged from 2018-03-AE. However, the HPC solver has several new default settings due to the new functionality and without setting the backward compatibility switches the results will be different, albeit for most models the differences should not be large.

**May 8, 2020:** 2020-01-AB Build (not to be confused with the 2020-10-AB Build further below), which addresses several bug fixes and enhancements. A description of bug fixes and other changes are highlighted in light green.

**October 16, 2020:** New 2020-10 Release (2020-10-AA Build), which incorporates: new default settings and computational enhancements to the 2020-01 release; a change in interfacing with external 1D schemes; new features and enhancements; bug fixes. Descriptions of the changes are in light blue.

**External 1D Scheme Users.** Use of the 2020-10-AA Build with one of the TUFLOW linked external 1D schemes, i.e. 12D DDA, Flood Modeller or XP-SWMM, will require an update from these software due to a change in the calling routines between external 1D schemes and TUFLOW – see Section 9.10.

**September 3, 2021:** 2020-10-AB Build, which includes enhancements, bug fixes and a some of the new features for the 2022 release that we are seeking feedback on in advance – these are marked as Beta and please send feedback to [support@tuflow.com](mailto:support@tuflow.com). The changes and new features are highlighted in light orange in these release notes.

**Note:** 2020-10-AB includes an improvement to a new feature built into 2020-10-AA for how the HPC solver handles dry cell faces with high inertia when becoming wet. This change may cause a very slight change in results (a backward compatibility switch is provided) – see Section 5.8.3 for more details. It is recommended that 2020-10-AB is used instead of 2020-10-AA, especially if performing impact mapping.

**December 9, 2021:** 2020-10-AC Build, which contains several minor bug fixes. A description of these changes is highlighted in light grey.

**March 9, 2022:** 2020-10-AD Build, which contains some minor enhancements including significantly faster runtimes for 1D/2D models using a Quadtree mesh and several bug fixes. A description of these changes is highlighted in light purple.

**June 15, 2022: Updated to address issue with beta functionality SGS subsampling of materials when used in combination with depth varying Manning's n values. See Section 3.7.3 for further information.**

**September 20, 2022: 2020-10-AE, which addresses several bug fixes and enhancements. A description of bug fixes and other changes are highlighted in light red. Notably due to the issue identified in the June release notes update, for the beta SGS Approach == Method C approach, SGS Material sampling is off by default in the 2020-10-AE update.**

**All users of the 2018-03 and prior releases, especially if using the HPC solver, are recommended to utilise the superior functionality built into the 2020-10-AB Build or later.**

**Note: The [TUFLOW Manual](#) has been updated to align with the 2018-03-AD build, therefore, these release notes cover changes since Build 2018-03-AD.**

This document may be updated from time-to-time with new content from software updates.

## Summary

The TUFLOW Classic/HPC 2020-01 and 2020-10 releases represent a major step forward in 2D hydraulic modelling based on several years research, development, and benchmarking of three industry leading features built into the HPC 2D solver. And in addition to these features, the HPC 2D solver now also supports **Non-Newtonian Flow** and a new **Advection-Dispersion (AD)** modelling. The three major features are:

### Cell Size Insensitive Sub-Grid Turbulence (Eddy Viscosity)

The new sub-grid turbulence model built into the HPC 2D solver is based on the physics of turbulence and research in the literature and has been successfully benchmarked across a wide range of scales from small flumes to large rivers. Unlike the Smagorinsky and Constant models, which require calibration of their respective parameters as cell sizes become smaller than their depths or if used for modelling flume scale hydraulics, the new HPC turbulence model is cell size insensitive and can be applied from flume scale to large river without need, unless fine-tuning a calibration, to adjust the turbulence parameters.

### Quadtree Mesh

TUFLOW HPC now supports variable cell sizes using a quadtree mesh. A quadtree mesh is constructed by dividing a cell into four cells, with these cells able to be divided into four, and so on, allowing modellers to use larger cells in areas of flat terrain (eg. large flat floodplains, parks) and smaller cells where the terrain is variable or along primary flow paths (eg. river channels, road gutters, open channels). The benefits include: (a) much improved hydraulic computational delineation where most needed, (b) smaller memory footprint on the GPU card as a mesh structure is used rather than bounding rectangles that include a large percentage of inactive cells that consume memory, and (c) often a much-reduced total cell count typically leading to faster simulations by a factor of 2 to 5.

### Sub-Grid Sampling (SGS)

Sub-grid sampling (SGS) stores and uses curves representing the sub-2D-cell terrain data of the DEMs, TINs and Z shapes used to construct the model instead of each 2D cell and each 2D face having one elevation. Benchmarking has shown the benefits to be substantial and to be a game changer for certain types of applications, for example:

- Catchment scale models flow much more effectively with water not being “trapped” by a coarse cell resolution, and, importantly, amazing cell size convergence (ie. demonstration that by reducing the cell size(s) the model results do not demonstrably change) at much coarser cell sizes.
- Disturbed flow fields that can be apparent along a “saw-tooth” regular mesh wet-dry boundary completely disappear, with no spurious additional head losses generated and the results consistent with a well-designed flexible mesh. This has major benefits in that open channels can now be accurately modelled using TUFLOW HPC using coarse cell sizes at any orientation to the channel, removing the need to utilise 1D open channels carved through the 2D domain.

In summary, TUFLOW HPC is the first 2D regular grid solver to offer a cell size insensitive turbulence scheme, easy cell size refinement using a quadtree mesh and substantially more accurate hydraulic conveyance along flow paths by using SGS. Add in TUFLOW HPC solver’s superior stability, 2nd-order accuracy and fast run-times on GPU devices, the 2020-01 and 2020-10 releases are, without doubt, the most exciting new offering in the hydraulic modelling industry for many years.

## Licensing and Executable Versions

To run simulations using Build 2020-01-AA and 2020-01-AB requires payment of the 2019/2020 annual software maintenance fee (invoiced mid-2019) and for the TUFLOW licence to have been updated (ie. via RaC/RaU files).

For Build 2020-10-AA, payment of the 2020/2021 annual software maintenance fee (invoiced mid-2020) is required.

For Build 2020-10-AB, payment of the 2021/2022 annual fee (invoiced mid-2021) is required.

For tutorial and demo models, or if running in free demo mode, no licence is required. For any licensing enquiries please contact [sales@tuflow.com](mailto:sales@tuflow.com), or for general support [support@tuflow.com](mailto:support@tuflow.com). Use of the TUFLOW software in any mode is bound by the [TUFLOW Products Licence Agreement](#).

The 2020-01 and 2020-10 releases include the new TUFLOW HPC Quadtree solver. To access the quadtree functionality a TUFLOW M2D/Quadtree Module Licence is required. The Classic M2D and HPC Quadtree module licences are one and the same, therefore, the Classic M2D Module licence can be used to run a HPC Quadtree model. Both the standard HPC solver and the new Quadtree solver can be run on CPU hardware without GPU hardware licenses. To run either solver on NVidia GPU devices a TUFLOW GPU Hardware Module Licence is required for each GPU device. Please refer to the [TUFLOW Price List](#) for more details or contact [sales@tuflow.com](mailto:sales@tuflow.com).

**Note: If running TUFLOW HPC on GPU hardware the NVidia drivers may need to be updated for the 2020-01 and 2020-10 releases. This is due to a change in the CUDA compiler version. If using TUFLOW HPC on a NVidia GPU device it is recommended to update the NVidia drivers prior to using the TUFLOW 2020-01 and 2020-10 releases.**

For the 2020-01 and 2020-10 releases, two executables are provided; 64-bit single precision (TUFLOW\_iSP\_w64.exe) and 64-bit double precision (TUFLOW\_iDP\_w64.exe). Note, if using the HPC solver (including Quadtree), it is rare that the double precision version is required due to the nature of the solution scheme. If in doubt, run the model using single and double precision, and if there is no significant change in results use single precision as the simulation will be faster and use less memory.

## Table of Contents

<b>Document Updates and Important Notices.....</b>	<b>1</b>
<b>Summary .....</b>	<b>2</b>
<b>Licensing and Executable Versions .....</b>	<b>3</b>
<b>1 Overview.....</b>	<b>13</b>
<b>1.1 2020-01 Release .....</b>	<b>13</b>
<b>1.2 2020-10 Release .....</b>	<b>13</b>
<b>2 1D Domain Construction .....</b>	<b>14</b>
<b>2.1 Faster Start-Up Times for Large 1D Networks.....</b>	<b>14</b>
<b>2.2 Dambreak (beta function) .....</b>	<b>14</b>
2.2.1 1D Piping Failure ( <i>PF</i> channel).....	14
2.2.2 1D Dam Failure ( <i>DF</i> channel) .....	16
2.2.3 Bug Fix for 1D DF and PF Channel Initialisation ERROR 1334 .....	20
2.2.4 Flow Width Contraction for Weir and DF Channels .....	20
2.2.5 Weir Calibration Factor for Operational Weirs .....	20
2.2.6 Bug Fix for WW Type DF Channel Flow Area Calculation .....	21
2.2.7 Reporting Weir Coefficient and Submergence Factor for DF Channels .....	21
<b>2.3 Operational Control Enhancements .....</b>	<b>21</b>
2.3.1 New User Defined Variable “Time Stamp” (2022 Beta Feature) .....	21
2.3.2 Access Status/Variables from Other Operational Control Channels (2022 Beta Feature) .....	22
<b>2.4 Bug fixes and Minor Enhancements.....</b>	<b>23</b>
2.4.1 Bug fix for minimum Nodal Area when using US Customary Units .....	23
2.4.2 Coefficients “a” and “b” for Operational Spillway (SPO) Channel .....	23
<b>3 2D Domain Construction .....</b>	<b>24</b>
<b>3.1 Quadtree Mesh (HPC Only).....</b>	<b>24</b>
3.1.1 Introduction.....	24
3.1.2 Quadtree .tcf Commands .....	25
3.1.3 Quadtree Control File (.qcf) – Mandatory Commands.....	25
3.1.4 Quadtree Control File (.qcf) – Optional Commands.....	26
3.1.5 Defining Mesh Refinement Polygons .....	26
3.1.6 Quadtree Module Licence .....	30
3.1.7 Minor Bug Fixes and Changes for 2020-01-AB .....	30
3.1.8 Bug Fix for Quadtree for Thick 2d_zsh Lines and Boundary Lines .....	31
3.1.9 Minor Bug Fixes and Changes for 2020-10-AC .....	33

<b>3.2</b>	<b>Sub-Grid Sampling, SGS (HPC Only, including Quadtree)</b>	<b>34</b>
3.2.1	Introduction	34
3.2.2	Without SGS (Traditional Approach)	36
3.2.3	SGS Methodology	37
3.2.4	SGS .tcf Commands	38
3.2.5	SGS .tgc Commands	40
3.2.6	SGS Check File Output: zpt	42
3.2.7	SGS Check File Output: DEM_Z	43
3.2.8	Map Output and Re-Mapping	43
3.2.9	Treatment of Infiltration and Negative Rainfall with SGS Enabled	44
3.2.10	Bug Fix for Read TIN Zpts	44
3.2.11	Bug Fix for Z Shape Polygon with “Add” Option.	44
3.2.12	Bug Fix for Corner Output with SGS and Variable Z Shape Output	44
3.2.13	SGS and Double Precision TUFLOW	45
3.2.14	Bug Fix for SGS with TINS Smaller than Cell Size	45
3.2.15	Bug Fix for SGS with Layered Flow Constrictions.	45
3.2.16	Bug Fix for SGS and Variable Z Shape	45
3.2.17	Bug Fix for SGS Models with very Large TINs	45
3.2.18	Bug Fix for SGS Models with Gully Lines and Read GIS Zpts Polygons	46
3.2.19	Change to “Zout” Attribute in _zsh_zpt_check File	46
3.2.20	SGS Approach == Method C (2022 Beta Feature)	46
<b>3.3</b>	<b>Non-Linear Failure of Variable Z Shapes</b>	<b>46</b>
<b>3.4</b>	<b>2D Layered Flow Constrictions - 2D Bridge Decks</b>	<b>47</b>
3.4.1	CFD Benchmarking Study	47
3.4.2	New 2d_LFC Approaches for Modelling Pressure Flow	50
<b>3.5</b>	<b>Bug Fixes and Minor Enhancements</b>	<b>55</b>
3.5.1	Bug Fix for Read GIS Z Line MIN	55
3.5.2	Support for LandXML Files with Multiple Surfaces	55
3.5.3	Bug Fix for Layered Flow Constrictions	55
3.5.4	Bug Fix for XF Files for TINS Created by TUFLOW	55
3.5.5	Bug Fix for TIN XF Files that Merge Elevations	56
3.5.6	Bug Fix for Geometry Control Update for HPC/Quadtree Models with Restart File	56
3.5.7	Additional check for “Grid Size” commands in .tgc	57

<b>3.6</b>	<b>Support for 12DA SuperTINs .....</b>	<b>57</b>
<b>3.7</b>	<b>New SGS Approach and High-Resolution Output (2022 Beta Feature) .....</b>	<b>58</b>
3.7.1	Description (2022 Beta Feature) .....	58
3.7.2	.tcf Commands (2022 Beta Feature) .....	58
3.7.3	Sampling of Material Manning's n Values (2022 Beta Feature) .....	59
3.7.4	High Resolution (HR) Grid/Raster Map Outputs (2022 Beta Feature) .....	61
3.7.5	High Resolution Output Water Level Interpolation (2022 Beta Feature) .....	66
3.7.6	Treatment of Thin Breakline in High Resolution Output (2022 Beta Feature) .....	69
3.7.7	Breakline Detection (2022 Beta Feature) .....	72
<b>4</b>	<b>1D Solver (ESTRY) .....</b>	<b>75</b>
<b>4.1</b>	<b>Operational Spillway (SPO) Channel Bug Fixes .....</b>	<b>75</b>
<b>4.2</b>	<b>Change to Ogee Crest Weir (WO) Discharge Coefficient Calculation .....</b>	<b>75</b>
<b>4.3</b>	<b>Enhancement to QO Channel Depth/Head Discharge Calculation .....</b>	<b>77</b>
<b>4.4</b>	<b>Minor Changes and Bug fixes .....</b>	<b>77</b>
4.4.1	Bug Fix for Incomplete Cross-Section Data .....	77
<b>5</b>	<b>2D Solvers (Classic/HPC) .....</b>	<b>78</b>
<b>5.1</b>	<b>Overview .....</b>	<b>78</b>
<b>5.2</b>	<b>Mesh Size Insensitive Turbulence Model (HPC Only) .....</b>	<b>78</b>
5.2.1	Discussion on Why a New Turbulence Approach is Needed .....	78
5.2.2	Turbulence (Eddy Viscosity) Formulation .....	79
5.2.3	Viscosity Approach .....	80
5.2.4	Q&A on Turbulence .....	80
5.2.5	Limiting Manning's n for Turbulence Calculation (2022 Beta Feature) .....	82
<b>5.3</b>	<b>Flow Depth Used for Bed Friction Calculation (HPC Only, incl. Quadtree excl. SGS) .....</b>	<b>83</b>
<b>5.4</b>	<b>Non-Newtonian Flow (HPC Only) .....</b>	<b>83</b>
5.4.1	Non-Newtonian Enhancements for 2020-10-AB .....	85
<b>5.5</b>	<b>HPC GPU Advection-Dispersion (AD) Solver .....</b>	<b>85</b>
5.5.1	Bug Fix for Advection Dispersion (AD) in HPC .....	86
5.5.2	Support for Settling and Decay with HPC AD .....	86
<b>5.6</b>	<b>Specification of Maximum Timestep for HPC Simulations .....</b>	<b>86</b>
<b>5.7</b>	<b>2D Weir Flow Factor in HPC Solver (2022 Beta Feature) .....</b>	<b>86</b>
<b>5.8</b>	<b>HPC Stability and Computational Improvements for 2020-10-AB .....</b>	<b>87</b>
5.8.1	Log-Law Roughness Improvements .....	87
5.8.2	Depth-Varying Manning's n Improvements .....	87
5.8.3	Dry Face treatment (HPC Only) .....	87

<b>6</b>	<b>Boundaries and Links</b>	<b>89</b>
<b>6.1</b>	<b>Improved HPC Water Level (H and QT) Boundaries (HPC Only)</b>	<b>89</b>
6.1.1	Additional Approach for HPC Boundary Approach	89
<b>6.2</b>	<b>Structure Loss Adjustment for SX Connected 1D Structures (Beta Function)</b>	<b>89</b>
<b>6.3</b>	<b>Bug Fixes and Minor Enhancements</b>	<b>91</b>
6.3.1	1d_bc QT Boundary Duplication	91
6.3.2	Bug Fix for HT Boundaries in HPC	91
6.3.3	Bug Fix for SX links in Quadtree	91
6.3.4	Bug Fix for Storage Reduction Factor on SX links in Quadtree	91
6.3.5	Bug Fix for Global Rainfall Initial Losses in Quadtree	91
6.3.6	Bug Fix for Quadtree Boundary Control File not stopping on ERROR	92
6.3.7	Bug Fix for Quadtree with Wu Turbulence	92
6.3.8	Bug Fix for Small Timestepping in HPC (including Quadtree)	92
6.3.9	Bug Fix for NetCDF Rainfall Times	92
6.3.10	Bug Fix for NetCDF Rainfall Grids and Output Drive	92
6.3.11	Bug Fix for Quadtree HX Links with CN Weighting Factors	92
6.3.12	Quadtree Support for SA Points	93
6.3.13	Bug Fix for Quadtree QT Boundary with no Elevation Change	93
6.3.14	H2D Trigger Supported in Quadtree for Operational Structures	93
6.3.15	Bug Fix for Reading RAFTS loc and tot with over 10,000 Timesteps	93
6.3.16	Bug Fix for HQ Boundary Types with Log-Law Material Approach	93
6.3.17	Boundary Lines Creating an Invalid Quadtree Mesh	93
6.3.18	HQ Boundaries Missing Cells	93
6.3.19	Small SA Regions in Quadtree Meshes	93
6.3.20	Support for Streams in Quadtree Meshes	94
6.3.21	Bug Fix for HPC with SA ALL and Streams	94
6.3.22	Update to Message 2208	94
6.3.23	Improved Messaging for ERROR 2048	94
6.3.24	Boundary Updates to Ensure Reproducibility of Results on HPC and Quadtree	94
6.3.25	Improved Error Message if 1D Boundary is Read into 2D Boundary Control File	95
6.3.26	SX Cell Elevation Check Not Conducted for 2d_bc SX lines with Multiple Segments	95
6.3.27	Bug fix for SX Cell Elevation Check when using 1d_nwk nodes	95
6.3.28	Bug fixes to SX Sag connections when more than one 2D cell used	95



6.3.29	Speed Improvements for Quadtree Boundaries .....	95
6.3.30	Cell Face Elevation Lowered by SX boundary Z Flag .....	96
6.3.31	SA Triggers Fixed in HPC .....	96
6.3.32	Bug fix for QT boundary for Quadtree models .....	96
6.3.33	Fixed elevation check for Quadtree Models with HX links .....	96
6.3.34	Bug fix for Flood-Modeller with HX links and Quadtree Solver .....	96
6.3.35	Significantly Improved Runtimes of 1D/2D Models Using Quadtree Solver .....	96
<b>6.4</b>	<b>Changes for Global Rainfall Losses and Material Rainfall Losses .....</b>	<b>97</b>
6.4.1	Bug Fix for Global Losses and HPC .....	98
<b>6.5</b>	<b>HPC New Drying approach with Soil Infiltration.....</b>	<b>98</b>
<b>6.6</b>	<b>HPC (including Quadtree) support Virtual Pipes.....</b>	<b>99</b>
6.6.1	ESTRY Support Virtual Pipes Outlets .....	99
<b>6.7</b>	<b>HPC (including Quadtree) Support Classic Style HQ Curve .....</b>	<b>99</b>
6.7.1	Bug Fix for Quadtree if HQ Boundary Selects Only a Single Cell .....	99
<b>6.8</b>	<b>HPC Support for Reducing Groundwater Level .....</b>	<b>100</b>
<b>7</b>	<b>Outputs and Check Files .....</b>	<b>101</b>
<b>7.1</b>	<b>Time-Series Output in NetCDF format .....</b>	<b>101</b>
7.1.1	Bug Fix for NetCDF Time-Series and Quadtree .....	101
<b>7.2</b>	<b>Extra Plot Outputs .....</b>	<b>101</b>
7.2.1	Flow Regimes.....	101
7.2.2	Mass Balance Error.....	101
7.2.3	Channel Losses .....	102
<b>7.3</b>	<b>NetCDF Map Output.....</b>	<b>102</b>
7.3.1	NetCDF Raster Grid Output (NC) Enhancement and Bug Fix.....	102
7.3.2	NetCDF Raster Grid FEWS Configuration .xml File .....	102
<b>7.4</b>	<b>Quadtree Models – Map Output Considerations .....</b>	<b>102</b>
7.4.1	Overview .....	102
<b>7.5</b>	<b>Sub-Grid Sampling (SGS) – Map Output Considerations.....</b>	<b>103</b>
7.5.1	Overview .....	103
7.5.2	Map Cutoff SGS == Commands .....	104
7.5.3	SGS Interpolation Commands for Map Output .....	105
7.5.4	SGS Mapped Extent Options .....	106
7.5.5	SGS Corner Interpolation Enhancement at Wet/Dry Interface .....	107
7.5.6	SGS Cell Averaged Depth Output (2022 Beta Feature) .....	107



7.5.7	SGS Velocity Based Output Depth .....	109
<b>7.6</b>	<b>SGS – Remapping Model Outputs to a Finer Resolution DEM .....</b>	<b>109</b>
<b>7.7</b>	<b>Maximum Velocity Cutoff Depth now supported in HPC.....</b>	<b>114</b>
<b>7.8</b>	<b>New Map Output Data Types .....</b>	<b>114</b>
7.8.1	New Hazard Output – Port Macquarie-Hastings Council.....	114
7.8.2	New Hazard Output – Hamilton City Council NZ .....	114
7.8.3	New User Defined Hazard Output.....	116
<b>7.9</b>	<b>1D External Scheme WLL Temporal Map Output Bug for HPC Linked Models .....</b>	<b>117</b>
7.9.1	Build 2020-01-AB .....	117
7.9.2	Build 2020-10-AA Onwards.....	117
<b>7.10</b>	<b>Output Changes for 2020-10-AA .....</b>	<b>117</b>
7.10.1	Reporting Location Supported for HPC including Quadtree .....	117
7.10.2	Structure Groups Supported for HPC including Quadtree .....	117
7.10.3	Evacuation Routes Now Supported for HPC including Quadtree .....	117
7.10.4	GIS Workspaces for QGIS and MapInfo Support Relative Filepaths .....	117
7.10.5	12D TMO Map Output Format Supported for Quadtree Simulations.....	118
7.10.6	12D TGO Velocity Angle Map Output in Radians .....	118
7.10.7	New Plot Output (PO) Data Types .....	118
7.10.8	Additional Fields in the _ccA GIS Output File .....	119
7.10.9	Improved High Control Number Reporting.....	120
<b>7.11</b>	<b>Bug fixes and Minor Enhancements for 2020-10-AA .....</b>	<b>120</b>
7.11.1	Bug Fix for 2d_po Lines Outside of Quadtree Mesh.....	120
7.11.2	Bug Fix for 2d_po Regions Havg Type in Quadtree .....	120
7.11.3	Bug Fix for 1d_chan.csv.....	120
7.11.4	Bug Fix in Check File if using Read CnM Mannings.....	120
7.11.5	Increase Maximum Length of ID field in _TS GIS output layers .....	120
7.11.6	Extra Precision in _grd Check Files for SRF Value .....	121
7.11.7	Extra Precision in _grd Check Files for Z_Map_Cutoff in Quadtree (SGS only) .....	121
7.11.8	Extra Precision in _lfcsh_uvpt for Form Loss Values.....	121
7.11.9	Fix an Issue with Multiple Evacuation Routes with the Same Name .....	121
7.11.10	Bug Fix for SAC Check File .....	121
7.11.11	Fix an Issue with Holes in Raster Outputs .....	121
7.11.12	Vector Output at Dry Zpts .....	121
7.11.13	Map Cutoff Depth and Vector Output.....	123

<b>7.12</b>	<b>Output Changes for 2020-10-AB .....</b>	<b>123</b>
7.12.1	Output Zones are Now Supported for Quadtree Meshes .....	123
7.12.2	Time Output Cutoff Hazard Support in HPC .....	123
7.12.3	TMO Update for Quadtree .....	124
<b>7.13</b>	<b>Bug fixes and Minor Changes for Build 2020-10-AB .....</b>	<b>124</b>
7.13.1	HPC PO Online Causing Zero Values in Results .....	124
7.13.2	_TS.mif not Loading in Mapinfo .....	124
7.13.3	Q_ / QS PO Point Output Causing "Should not be here" Error.....	124
7.13.4	D_ PO Point Output in HPC Causing "Should not be here" Error.....	124
7.13.5	PO Character Limit Causing "Should not be here" Error .....	124
7.13.6	Bug Fix for QS PO Outputs in Models with no 1D Elements .....	125
7.13.7	Increase Decimal Places for Time Output Cutoff Depths .....	125
7.13.8	"Should not be here" While Writing 1D Check Files.....	125
7.13.9	Bug Fix for dGW Output in HPC Solver .....	125
7.13.10	Bug Fix for HX Flags in the 1d_to_2d Check File .....	125
7.13.11	_zpt_check File Update for SGS Models .....	125
7.13.12	Create TIN Zpts not Writing to sh_obj_R Check File .....	125
7.13.13	sh_obj_R Check File Not Writing when using XF Files .....	126
7.13.14	Quadtree Orientation Angle output in the .tlf File Now in Degrees .....	126
7.13.15	zsh_zpt Check Points at Deactivated Quadtree Cell Corners .....	126
7.13.16	DEM_Z Check File and SXL/SXZ Flag .....	126
7.13.17	Water "Puddles" in Dry Quadtree Cell .....	126
7.13.18	Number of Irregular Culvert Barrels in hydprop_check File .....	126
7.13.19	Extra precision in .hpc.dt.csv .....	126
7.13.20	Bug Fix for PO Region Outputs.....	126
7.13.21	Bug Fix for Model with Record Gauge Data .....	126
7.13.22	Advanced Weir "number of" attribute in Check File .....	127
<b>7.14</b>	<b>Bug Fixes and Minor Changes for Build 2020-10-AC .....</b>	<b>127</b>
7.14.1	Bug Fix for PO Output in Quadtree Model .....	127
7.14.2	Bug Fix for HR Output in double precision.....	127
7.14.3	Bug Fix for HR Output in Output Zone .....	127
7.14.4	PO Header Change for Quadtree .....	127
<b>7.15</b>	<b>Bug Fixes and Minor Changes for Build 2020-10-AD .....</b>	<b>127</b>

7.15.1	Bug Fix for PO Output in HPC .....	127
<b>8</b>	<b>Simulation Control .....</b>	<b>129</b>
<b>8.1</b>	<b>Default Simulation Log Folder Changed .....</b>	<b>129</b>
8.1.1	Include Username in Simulation Log Folder .....	129
8.1.2	Extra Licence Control File Location .....	129
8.1.3	Bug Fix for Simulation Log Folder.....	129
<b>8.2</b>	<b>Package Model (-pm) Enhancements .....</b>	<b>129</b>
8.2.1	Package Model Fixes 2020-10-AA.....	129
8.2.2	Package Model Fixes 2020-10-AB.....	130
8.2.3	Package Model Change in Build 2020-10-AD .....	130
<b>8.3</b>	<b>Version (-version) Option.....</b>	<b>131</b>
<b>8.4</b>	<b>Optimising Multi-GPU Performance (HPC Only) .....</b>	<b>131</b>
<b>8.5</b>	<b>NVLink – Multi-GPU Performance (HPC Only).....</b>	<b>131</b>
8.5.1	New Option to Control Peer to Peer GPU Access .....	132
<b>8.6</b>	<b>Shapefile Projection Changes .....</b>	<b>132</b>
8.6.1	Projection Check .....	132
<b>8.7</b>	<b>Ability to Pause a Simulation at Set Time / Interval .....</b>	<b>133</b>
<b>8.8</b>	<b>Copy Model Functionality .....</b>	<b>133</b>
8.8.1	Copy Model Fixes for 2020-10-AB .....	133
8.8.2	Copy Model Change for 2020-10-AD.....	133
<b>8.9</b>	<b>Backwards Compatibility “Defaults” command .....</b>	<b>133</b>
<b>9</b>	<b>Minor Enhancements and Bug Fixes for 2018-03 Release .....</b>	<b>134</b>
<b>9.1</b>	<b>Bug Fix for FC Shape Where Negative Depth Caused NaN .....</b>	<b>134</b>
<b>9.2</b>	<b>Bug Fix for SA streams .....</b>	<b>134</b>
<b>9.3</b>	<b>Bug Fix for PLOT_R.....</b>	<b>134</b>
<b>9.4</b>	<b>Enhancement to the QGS File .....</b>	<b>134</b>
<b>9.5</b>	<b>Bug Fix for Folder Creation with UNC Paths .....</b>	<b>134</b>
<b>9.6</b>	<b>Bug Fix for Default == Pre 2013 .....</b>	<b>134</b>
<b>9.7</b>	<b>Message Box Changes.....</b>	<b>134</b>
<b>9.8</b>	<b>Minor Changes for 2020-01-AB .....</b>	<b>134</b>
<b>9.9</b>	<b>Minor Changes for Build 2020-10-AA .....</b>	<b>135</b>
<b>9.10</b>	<b>Bug Fix for the Temporal WLL Map Output of an External 1D Scheme’s Results.....</b>	<b>136</b>
<b>9.11</b>	<b>Minor Changes for Build 2020-10-AB .....</b>	<b>136</b>
<b>10</b>	<b>Licensing and Installing.....</b>	<b>137</b>
<b>10.1</b>	<b>Security Certificate .....</b>	<b>137</b>
<b>10.2</b>	<b>Bug fixes and Minor Enhancements.....</b>	<b>137</b>

10.2.1	Excess Licences Released for Quadtree Simulations .....	137
10.2.2	Support Period Now Displayed .....	137
10.2.3	Command Line Argument -nwk (Network only) Now Works for CodeMeter Licences .....	137
10.2.4	Check on the GPU .ptx File Version .....	137
10.2.5	Updated licencing when running from 12D Model .....	138
<b>11</b>	<b>Backward Compatibility to the 2018-03 Release .....</b>	<b>139</b>
<b>12</b>	<b>Backward Compatibility to the 2020-01 Release .....</b>	<b>140</b>
<b>D</b>		

# 1 Overview

## 1.1 2020-01 Release

The 2020-01 release is a major release that includes substantial new features that provide major enhancements and benefits to 2D hydraulic modelling. A range of general new features, enhancements and bug fixes are also included.

The major new features are largely within the HPC 2D solver, making it arguably the most powerful 2D solver in the industry. The new functionality to HPC includes:

- Quadtree Mesh refinement – see Section 3.1.
- Sub Grid Sampling (SGS) of elevations for cell volume / cell face definition – see Section 3.2.
- Mesh size insensitive turbulence (eddy viscosity) solution – see Section 5.1.
- Non-Newtonian Flow – see Section 5.3.
- AD (Advection-Dispersion) scheme – see Section 5.5.

There are also some nice enhancements to both Classic and HPC such as much faster 1D model start-up time for models with large 1D networks.

The benefits of the new HPC sub-grid scale turbulence (eddy viscosity) scheme and sub-grid sampling (SGS) are anticipated to be significant and far-reaching for the industry, whilst Quadtree offers the modeller amazing flexibility to optimise the model resolutions across their study area according to the hydraulics, topography and objectives of the modelling. For a general description of Quadtree, SGS and new turbulence model refer to the Summary at the beginning of these release notes.

As always, it is recommended that when switching to a new build with an established model that test runs are carried out and comparisons made between the old and new builds (subtracting the two maximum h data sets and reviewing any differences is an easy way to do this). If you have any queries on the comparison outcomes, or require clarification or more detail on any of the points below, please email [support@tuflow.com](mailto:support@tuflow.com).

## 1.2 2020-10 Release

The 2020-10 release is a substantial update to the 2020-01 release that includes minor enhancements to the HPC and Quadtree solvers and new default settings (e.g. HQ boundaries are now defaulting to the same approach as used by the Classic solver). These improvements may cause a slight change in results in some models plus there is a change to the interfacing with external 1D schemes, hence the need to issue as a new release. There are also a range of bug fixes primarily relating to the new functionality in the 2020-01 release described above and several nice new features including: dambreak piping failure and new 1D dambreak channel; automatic adjustment of 1D structure losses according to the approach/departure 2D velocities; and two new 2D Layered Flow Constriction approaches for improved modelling of pressure flow at bridges.

## 2 1D Domain Construction

### 2.1 Faster Start-Up Times for Large 1D Networks

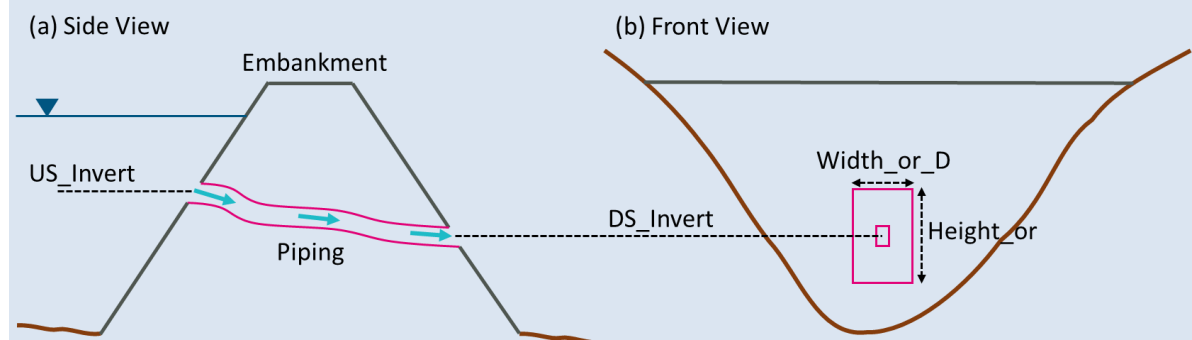
Reading and processing of 1D inputs has been significantly improved, particularly for large urban drainage models (>1,000 1D pipe network elements). For a tested model with 25,000 1D channels, the start-up was approximately 40 times faster with the 2020 version compared to the 2018 release changing the start-up time from nearly two hours to less than 3 minutes. No changes in model files required.

### 2.2 Dambreak (beta function)

#### 2.2.1 1D Piping Failure (PF channel)

Dam/levee breaks can be triggered by flow overtopping, as well as internal erosion (often called as “piping” or “piping failure”) when water seepages through the embankment forming a tiny flow path. Build 2020-10-AA has introduced a beta feature to model the “pipe failure” process using “PF” type 1D operational channel. In the 1d\_nwk layer, the following attributes can be used to set up the channel parameters:

- ID = Unique Channel ID.
- Type = “PF”.
- US\_Invert = Inlet elevation of the piping.
- DS\_Invert = Outlet elevation of the piping.
- Inlet\_Type = Refers to the Piping Failure operational control definition in .toc file.
- Width\_or\_Dia = Width of the piping orifice when fully breached.
- Height\_or\_WF = Height of the piping orifice when fully breached.



**1D Piping Failure**

Operational control commands specific to the Piping Failure channel are provided below:

- “**Period Failure**”: Defines the time in hours taken to reach the full size of piping failure. The default value is 0.0166667 (i.e. 60 seconds).
- “**Orifice Width Fully Open**”: Same as the “Width\_or\_Dia” attribute.
- “**Orifice Height Fully Open**”: Same as the “Height\_or\_WF” attribute.

- “Period Collapse”: Defines the time in hour taken to ‘shut down’ the piping failure channel after it reaches the full extend due to the channel collapsing. The default value is 0 (i.e. no collapsing at the end of pipe failure).
- “Orifice Opening”: Sets the piping failure status.
- “Cd”: Sets the discharge coefficient.

The flow rate through a piping failure channel is calculated using a simple orifice flow equation:

$$Q = C_d A (H_{up} - H_{down})^{0.5}$$

$$Q = C_d A \sqrt{2g(H_{up} - H_{down})}$$

Where  $C_d$  is the discharge coefficient set by “cd” command,  $A$  is the cross-sectional area of the orifice,  $H_{up}$  and  $H_{down}$  are water levels at the upstream/downstream 1D nodes.

**Note:** that the default  $C_d$  value was 4.8 in Build 2020-10-AA with the following flow equation:

$$Q = C_d A (H_{up} - H_{down})^{0.5}$$

which assumed  $g$  was in US Customary unit ( $32.17 \text{ ft s}^{-2}$ ). Build 2020-10-AB has updated the flow equation and changes the default  $C_d$  value to 0.6 to accommodate both SI and US Customary units.

For 2020-10-AA, it is assumed that the orifice shape is rectangular, and the orifice size increases linearly with time.

Build 2020-10-AB now supports circular shaped orifice flow, with rectangular shaped remaining as the default.

An example of a “Define Pipe Failure Control” block is provided below:

```
Define Pipe Failure Control == PF01      ! Start defining 'PF01'
    Period Failure == 0.5                ! (hr) Orifice reaches full size
                                         in 0.5 hr
    Orifice Shape == RECTANGULAR         ! Options CIRCULAR and RECTANGULAR
    Orifice Width Fully Open == 2.0
    Orifice Height Fully Open == 1.0
    Period Collapse (s) == 60            ! (s) Orifice closes in 60 s after
                                         reaching the full size
    Orifice Opening == CLOSE              ! Sets the initial status
    Cd == 0.6                           ! Q = Cd*A*(2g*(Hup-Hdown))^0.5
! User Variables -----
    Trigger == HU                        ! Sets the variable 'trigger' to
                                         the upstream node water level
! Logic Commands -----
    If Trigger > 51.5                    ! If the value of variable
                                         'trigger' is greater than 51.5
        Orifice Opening == OPEN          ! Changes the status of 'PF01' to
                                         be OPEN (start to breach)
    Else
        Orifice Opening == NO CHANGE     ! Sets the status the 'PF01' to
```



be remain unchanged

```
End IF
End Define ! End block of commands for 'PF01'
```

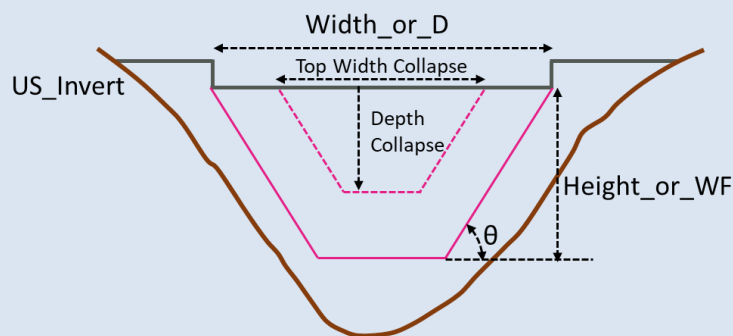
Note that the default units for “Period Failure” and “Period Collapse” are in hours, but minutes or seconds can be used if “(min)” or “(s)” is specified. For example, specifying either of the commands below will set the time taken to reach the full size of piping failure to half an hour:

```
Period Failure == 0.5
Period Failure (min) == 30
Period Failure (s) == 1800
```

## 2.2.2 1D Dam Failure (DF channel)

Build 2020-10-AA has also implemented “DF” type 1D operational channel to model dam/levee breaks in 1D. This channel can commence independently or can be triggered by 1D pipe failure using “PF” type trigger. The flow rate is calculated based on weir flow equation (Section 5.7.3 in TUFLOW manual). Either rectangular or trapezoidal cross-section can be applied, and the opening size is assumed to increase linearly with time. In the 1d\_nwk layer, the following attributes can be used to set up the channel parameters:

- ID = Unique Channel ID.
- Type = “DF”.
- US\_Invert = Elevation of the dam crest level before failure
- Inlet\_Type = Refers to the Dam Failure operational control definition in .toc file.
- Width\_or\_Dia = Top width of the dam failure channel opening when fully breached.
- ~~Height\_or\_WF = Height of dam failure opening when fully breached.~~
- Height\_or\_WF = Weir coefficient  $C_d$  of the intact part.



### 1D Dam Failure

Operational control commands specific to the Dam Failure channel are provided below:

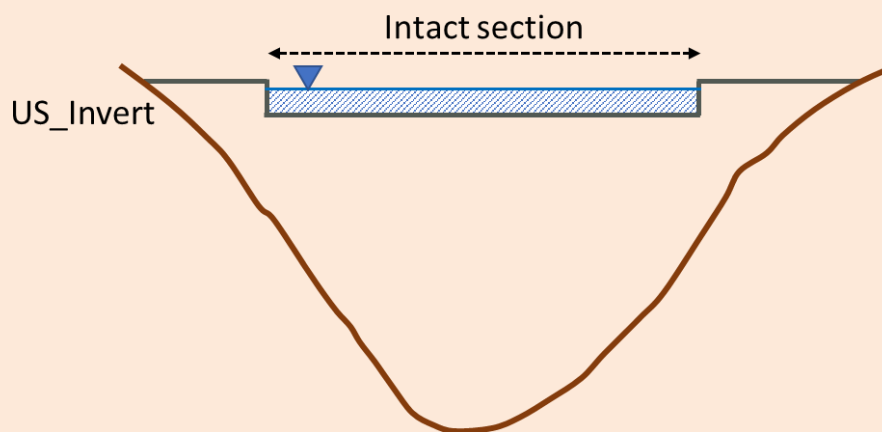
- “Period Failure”: Defines the time in hours taken to reach the full size of dam failure. The default value is 0.0166667 (i.e. 60 seconds).
- “Top Width Fully Breached”: Same as the “Width\_or\_Dia” attribute. Top width of the dam failure channel opening when fully breached. This value can be set smaller than the “Width\_or\_Dia” value if the maximum breach width does not reach the total width of the dam crest.

- “Depth Fully Breached”: Same as the “Height\_or\_WF” attribute. Height of dam failure opening when fully breached.
- “Side Slope (degree)”: Sets the angle of side slope. This is used for bottom width and flow area calculation.
- “Period Collapse”: Defines the time in hour taken to rapidly change the dam breach opening at the beginning of the dam break. This is usually the case, especially if a dam failure is triggered by a piping failure (i.e. linked with a “PF” channel). The default value is 0 (i.e. no collapsing at the beginning dam failure).
- “Depth Collapse”: defines the depth that the dam crest collapse during the “Period Collapse” at the beginning of the dam break.
- “Top Width Collapse”: defines the top width that opens up during the “Period Collapse”. This command is optional. If not defined, the default collapse width is set proportionally as “Top Width Fully Breached” \* “Depth Collapse” / “Depth Fully Breached”.
- “Breach Opening”: Sets the dam failure status.

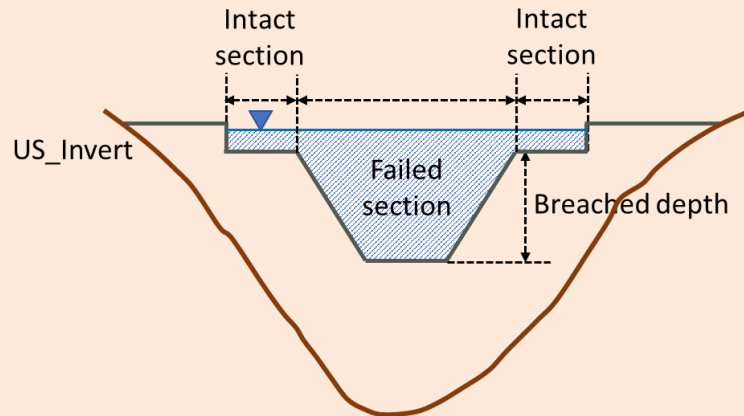
During any period of the dam failure, the Side Slope is set as constant. If the bottom width calculated from the width, depth and the side slope becomes negative, a V shaped cross section is assumed based on the width and the side slope.

As of Build 2020-10-AB, a DF channel can be subject to one of the following three flow phases, depending on the upstream water level and state of the breach.

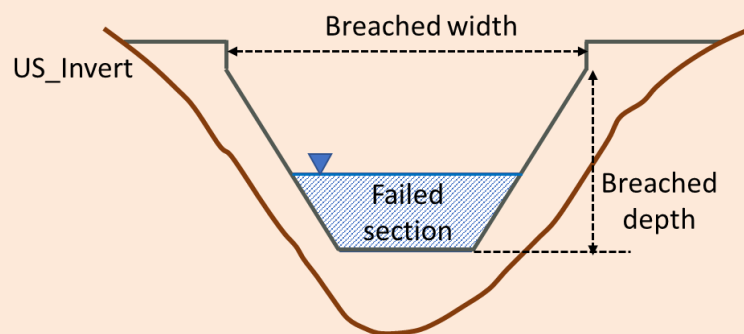
- Flow overtopping above the crest level can occur before the dam break. At this stage, all the water is flowing through the “Intact” section of the dam crest.



- After the dam break commences, the water can flow through both the “Intact” section of the dam crest (provided the upstream water level is greater than the invert of the intact section), and the “failed” section. The total flow is reported in the .eof and the \_1d\_Q.csv output file, while the flow rate for each section is reported separately in the \_1d\_O.csv output file.



- Once the maximum breach width is reached or there is no flow over any remaining intact sections, the flow is confined entirely within the "Failed" section.



The flow calculation through a 1D dam break channel is computed for both the intact and failed sections, with the user able to set the weir flow equation parameters separately for both sections as per the table below.

"Intact" Section	"Failed" Section	Sets Flat Cell/Face
Weir Type Intact	Weir Type Failed	Sets the weir channel type using the flags in Table 5-7 of the TUFLOW manual. If this command is not specified, the defaults are "WB" for the Intact section and "WW" for the failed section. Note that the "Side Slope" is not considered in the cross-section calculation for non-WW type weirs.
Cd Intact	Cd Failed	Sets the weir coefficient, $C_d$ , in its dimensionless form.
Cf Intact	Cf Failed	Sets the weir coefficient adjustment factor $C_f$ .
Ex Intact	Ex Failed	Sets the weir flow equation exponent $E_x$ .
Sa Intact	Sa Failed	Sets the submergence factor "a" exponent in the Villemonte Equation for calculating the weir submergence factor $C_{sf}$ .
Sb Intact	Sb Failed	Sets the submergence factor "b" exponent in the Villemonte Equation for calculating the weir submergence factor $C_{sf}$ .

The 1d\_nwk attributes can be used to set the  $C_d$ ,  $E_x$ ,  $a$  and  $b$  values for the "Intact" section. If a parameter is set by both a .toc command and a 1d\_nwk attribute, the .toc command will prevail over the 1d\_nwk attribute, with the final value applied output to the .eof file for cross-checking.

An example of a "Define Dam Failure Control" block is provided below:

```

Define Dam Failure Control == DF01      ! Start defining 'DF01'
    Period Failure == 0.5                ! (hr) Breach opening reaches full
                                         size in 0.5 hr

    Top Width Fully Breached == 15.0
    Depth Fully Breached == 5.0
    Side Slope (degree) == 45
    Period Collapse (s) == 60           ! (s) Breach opens rapidly to the
                                         depth defined by 'Depth Collapse'
                                         in the first 60 s after the dam
                                         failure commences.

    Depth Collapse == 4.0
    Breach Opening == CLOSE             ! Sets the initial status

Weir Type Intact == WB
Cf Intact == 1.2
Weir Type Failed == WW
Cf Failed == 0.8

! User Variables -----
    Trigger == PF PF01                 ! Sets the variable 'trigger' to
                                         the status of the Piping Failure
                                         at 'PF01'

! Logic Commands -----
    If Trigger > 0.999                 ! If PF01 reaches 99.9% of its
                                         full breach size.

        Breach Opening == OPEN         ! Sets the status the Dam Failure
                                         to be OPEN (start to breach)

    Else

        Breach Opening == NO CHANGE    ! Sets the status the Dam Failure
                                         to be remain unchanged

    End IF

End Define                             ! End block of commands for 'DF01'

```

Note that in the example above, dam crest collapses by 4m (“Depth Collapse”) in the first 60 seconds (“Period Collapse (s)”), and it takes another 0.5 hr (“Period Failure”) to reach the “Depth Fully Breached” of 5m.

### 2.2.3 Bug Fix for 1D DF and PF Channel Initialisation ERROR 1334

Build 2020-10-AB fixes an initialisation issue when counting the number of 1D DF and PF channels for allocating memory. Previously ERROR 1334 could be output and the simulation stopped when DF or PF channels were included in the model.

### 2.2.4 Flow Width Contraction for Weir and DF Channels

Section 9.11 of USBR (1987) includes an allowance for side or horizontal flow contraction caused by piers and abutments that contract the flow and reduces the flow width over the weir ( $W$ ) to an effective flow width ( $W'$ ) using:

$$W' = W - 2(NK_p + K_a)H_e$$

where  $N$  is the number of piers,  $K_p$  is the pier contraction coefficient, and  $K_a$  is the abutment contraction coefficient. The following new .toc commands have been added to the operational weirs and the 1D dam failure (DF) channels to take into account this effect of flow width contraction.

Operational weir	1D Dam Failure (DF channel)	Sets Flat Cell/Face
Number of Piers	Number of Piers	Sets the number of piers over a weir
Kp	Kp Intact	Sets the pier contraction coefficient
Ka	Ka Intact	Sets the abutment contraction coefficient

Note that the flow width contraction is applied to the “Intact” part of the DF channel only (see Section 2.2.2). The default values for these parameters are set to zero (0) for backward compatibility.

Section 9.11 of USBR (1987) recommends the following values for the pier contraction coefficient  $K_p$ :

- Square-nosed piers with corners rounded on a radius equal to about 0.1 of the pier thickness:  $K_p = 0.02$
- Round-nosed piers:  $K_p = 0.01$
- Pointed-nose piers:  $K_p = 0.0$

The recommended values for the abutment contraction coefficient  $K_a$  are:

- Square abutments with headwall at 90° to direction of flow:  $K_a = 0.20$
- Rounded abutments with headwall at 90° to direction of flow, when  $0.15H_0 \leq r \leq 0.5H_0$ :  $K_a = 0.10$
- Rounded abutments where  $r > 0.5H_0$ , and headwall is placed not more than 45° to direction of flow:  $K_a = 0.0$

Note that in the relationship above, the effective flow width ( $W'$ ) reduces linearly as the upstream head ( $H_e$ ) increases. The ratio of contraction calculated by this equation is usually small relative to the actual width of weir ( $W$ ). But to avoid unreasonable contraction ratios, a lower limit of  $0.6W$  is applied:

$$W' = \max(0.6W, W')$$

### 2.2.5 Weir Calibration Factor for Operational Weirs

Prior to Build 2020-10-AB, it was not possible to specify the weir calibration factor ( $C_f$ ) for operational weirs because the “Height\_or\_WF” attribute is used to define the height of an operational weir above the crest level. Build 2020-10-AB includes a new .toc command to define  $C_f$ , e.g.:

```
Cf == 1.2
```

The default value is 1.0 for backward compatibility, i.e. no adjustment.

## 2.2.6 Bug Fix for WW Type DF Channel Flow Area Calculation

2020-10-AD fixes a flow area calculation issue for DF (Dam Failure) channels with side slope. Prior to this version, the flow area was slightly overestimated by the DF channel routine, which could cause overestimation of the flow rate for WW type DF channels. For DF channels with other weir types, this fix does not affect the modelled flow rates, but could change the output velocity of the channels. This issue is specific to DF channels only and does not affect the 1D weir channel routines.

## 2.2.7 Reporting Weir Coefficient and Submergence Factor for DF Channels

2020-10-AD build outputs the weir coefficient and the submergence factor for DF channels in the \_TSL\_P output file.

## 2.3 Operational Control Enhancements

### 2.3.1 New User Defined Variable “Time Stamp” (2022 Beta Feature)

Build 2020-10-AB includes a new variable setting labelled “Time Stamp” to set a variable to a specific time (in hours) during a simulation. Unlike the variable setting “Time of Model”, which updates the variable every timestep to the current simulation time, “Time Stamp” will set the variable to the current simulation time then keep the variable constant until the variable is changed by a repeat execution of “Time Stamp”. This allows users to trigger an operation with a time lag from a fixed point in time during the simulation.

In the example below, the pump will remain operational whilst the upstream water level exceeds 10 m, but once the water level falls below 10 m the pump remains on for 15 min (0.25 hours) before shutting down unless the upstream water level once again exceeds 10 m, and the 15 min shutdown period is reset.

```
! User defined variables
Hup == HU                                ! upstream water level
Model_Time == Time of Model              ! current simulation timestep
Last_Time_above_10m == 0

! Operational Logic
If Hup > 10
    Pump Operation == On
    Last_Time_above_10m == Time Stamp ! update Last_Time_above_10m as long
as Hup is above 10m
Else If Hup <= 10 and Model_Time > Last_Time_above_10m + 0.25
    Pump Operation == Off
Else
    Pump Operation == No Change
End If
```

### 2.3.2 Access Status/Variables from Other Operational Control Channels (2022 Beta Feature)

Build 2020-10-AB adds a powerful feature to allow operational control channels to refer to the status/variables from other operational control channels. This makes coordinated operations possible between multiple operational structures. The supported status/variables include:

- “Period No Change” returns the time in hours since there was a change in operation in another operational channel.
- “Status” returns the following keywords in text format for different types of operational channels.

Operational Channels	Status
Pumps (PO)	"Off", "Dry", "Below Soffit", "Starting", "Stopping", "Constant", "Pump Curve"
Gated Drowned Rectangular Culverts (RO) Sluice Gates (SGO) Spillways with Gates (SPO) Weirs (WBO, WCO, WDO, WOO, WRO, WTO) Dam failure (DF) Pipe failure (PF)	"Opening", "Closing", "Steady", "Closed", "Fully Open"

- “Operational Variables” returns the numeric values of operational variables from other operational channels, such as “Gate Height”, “Weir Width”, etc. The full list of keywords is summarised below.

Operational Channels	Operational Variables
Pumps (PO)	Pump Flow
Q channel (QO)	Q Opening Q Fully Open
Gated Drowned Rectangular Culverts (RO)	Gate Height Gate Width
Sluice Gates (SGO) Spillways with Gates (SPO)	Gate Opening
Weirs (WBO, WCO, WDO, WOO, WRO, WTO)	Weir Height Weir Width
Dam failure (DF)	Breach Depth Top Width
Pipe failure (PF)	Orifice Height Orifice Width

The operational control name must be added at the end of user defined variable commands listed above to refer to the Period of No Change, Status or Operational Variable from other operational channels. In the example below, the model has an operational sluice gate (“SGate1”) and an operational pump (“Pump1”). Pump1 calls the “status” of SGate1 and operates only if SGate1’s status is “Closed”.

```
Define Sluice Gate Control == SGate1
...
End Define
```



```

Define Pump Control == Pump1
    Pump Operation == Off
    Pump Capacity == 0.5
    ! User defined variables
    Hup == HU                                ! upstream water level
    SGatel_Status == STATUS SGatel          ! status of SGatel
    ! Operational Logic
    If Hup > 10 and SGatel_Status = Closed
        Pump Operation == On
    Else
        Pump Operation == Off
    End If
End Define

```

## 2.4 Bug fixes and Minor Enhancements

### 2.4.1 Bug fix for minimum Nodal Area when using US Customary Units

Build 2020-10-AB correctly converts the default minimum Nodal Area (NA) value of 1 m<sup>2</sup> to ft<sup>2</sup> if using US Customary units. Prior to this if using US Customary units, a value of 1 ft<sup>2</sup> was used instead of ~10.76 ft<sup>2</sup>. The default Minimum NA Pit == value was correctly converted but not the default Minimum NA == value. For backward compatibility with US Customary models "Minimum NA == 1" can be set in the .ecf file or as a 1D command in the .tcf. This change was made after the Beta3 version of 2020-10-AB.

### 2.4.2 Coefficients "a" and "b" for Operational Spillway (SPO) Channel

Prior to Build 2020-10-AB, it was not possible to specify the coefficients "a" and "b" used for the weir submergence factor  $C_{sf}$  calculation (Section 5.7.3.4 of the 2018 TUFLOW manual) through the 1d\_nwk "EntryC\_or\_WSa" and "ExitC\_or\_WSb" attributes. This is now fixed in Build 2020-10-AB. Previously values of a = 6.992 and b = 0.648 were adopted.

### 2.4.3 User Specified Maximum Number of Operational Channels

Build 2020-10-AE allows the user to specify the maximum number of operational control channels. Builds prior to 2020-10-AE had a hard limit of 1,000 which includes operational channels in the 1d\_nwk layer and the operational control definition in the .toc. If more than 1,000 operational channels and operational definitions are found, then ERROR 1528 or 1504 will occur. For the 2020-10-AE update an .ecf command can be specified that allows this to be user defined. The default value remains at 1,000. The command syntax is:

```
Maximum Operational Controls == <number of operational channels/control definitions>.
```

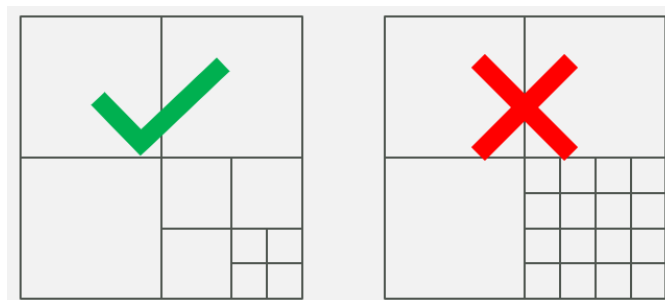
## 3 2D Domain Construction

### 3.1 Quadtree Mesh (HPC Only)

#### 3.1.1 Introduction

The Quadtree mesh refinement functionality allows the user to vary the resolution of a model using the HPC 2D Solver. Quadtree refinement allows for recursive division of square TUFLOW cells into four smaller squares. As such, using Quadtree the refined cells all share a common orientation.

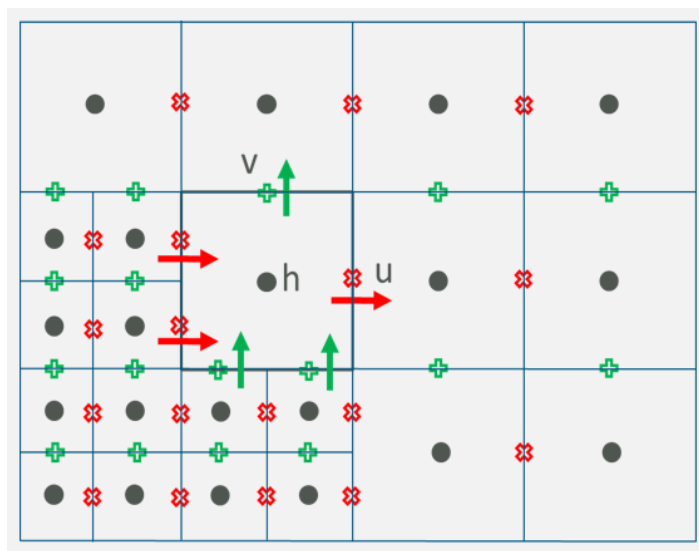
Each 2D cell face can have a maximum of 2 adjoining cell faces, meaning that multiple levels of refinement will need to transition through the intermediate resolutions. For example, the image on the left is a valid Quadtree mesh whereas the image on the right shows an invalid mesh. It is possible for a cell to have up to 8 neighbouring (smaller) cells.



**Quadtree Mesh Examples**

The mesh is automatically generated by TUFLOW based on a series of user specified GIS polygons. This process for generating a Quadtree mesh is outlined in Section 3.1.5 below.

The Quadtree solver uses a modified version of the HPC solver, with water levels calculated at cell centres and flows at cell faces, as per TUFLOW HPC and TUFLOW Classic. In the image below, the black dots represent cell centred mean depth data, the red crosses the face centred u velocity data, and the green plus symbols the face centred v velocity data.



**Quadtree Scheme Computational Locations**

The full 2D SWE are applied across changes in cell resolution, with complete computation of all the 2D SWE terms, unlike the TUFLOW Classic Multiple 2D Domain feature which utilises hidden 1D nodes. The result is a seamless solution across changes in resolution without artefacts or wave reflections.

Like TUFLOW HPC, the Quadtree solver uses an explicit finite volume solution that is 2<sup>nd</sup> order in space and 4<sup>th</sup> order in time. However, there are some subtle differences between the HPC single grid and Quadtree solvers that mean they produce near identical, though not identical results if both run are over the same single grid (same cell size) mesh.

Like the HPC single grid solver, the HPC Quadtree solver can run on either GPU or CPU hardware. For the initial TUFLOW 2020-01 release build only a single GPU device or CPU core can be used for a simulation, however, parallelisation of the HPC Quadtree solver across multiple GPU devices and CPU cores is now completed and will form part of the 2022 release.

**Note:** Quadtree models can have a much smaller memory footprint than non-Quadtree (single grid) models, because Quadtree models only store a mesh of active cells compared with single grid models that store the bounding rectangle, which may include large areas of inactive (redundant) cells consuming memory. Therefore, typically very large HPC single grid models that required two or more GPU devices to provide enough memory, may run on a single GPU device if using Quadtree.

### 3.1.2 Quadtree .tcf Commands

To run a Quadtree simulation the solution scheme should be set to HPC and then a Quadtree Control file specified using the command “`Quadtree Control File ==` “. For example:

```
If Scenario == HPC
    Solution Scheme == HPC
    Hardware == GPU
Else If Scenario == Quadtree
    Solution Scheme == HPC
    Quadtree Control File == ..\model\quadtree_001.qcf
    Hardware == GPU
End if
```

The keyword “Single Level” can be used instead of a control file (e.g. “`Quadtree Control File == Single Level`”), to run the Quadtree solver on a single grid model (ie. a fixed cell size). The Quadtree Control File is described in the next section.

### 3.1.3 Quadtree Control File (.qcf) – Mandatory Commands

The Quadtree control file is used to define the mesh refinement areas and optionally the model location and extent for a Quadtree model.

The following commands are mandatory in the new Quadtree Control File.

```
Base Cell Size == <cell size in m/ft> | {TGC}
```

Used to set the Level 1 (parent) cell size. If set to a numerical value can be used to override the cell size command in the .tgc file. If set to TGC, then the cell size defined in the .tgc is used.

```
Model Origin and Extent == Auto | TGC
```

If set to “Auto” the extents of the Level 1 GIS polygon are used to define the model origin and extents. If set to “TGC”, the model is located as per the commands in the .tgc file. Note the angle of the model is defined with the Orientation Angle command below. Also note, if set to “Auto” the GIS nesting polygons must have a Level 1 polygon defined, otherwise an ERROR is generated. The default setting is “TGC” if `Quadtree Control File == Single Level` and “Auto” if a .qcf (Quadtree control file) is specified.

```
Orientation Angle == <angle in degrees> | Optimise | {TGC}
```

If set to a numerical value defines the model orientation angle and overwrites any angle / location .tgc commands. If Set to “Optimise” the parent Level 1 polygon is used to optimise the angle of the mesh. As such the GIS nesting polygon must have a Level 1 polygon defined.

```
Read GIS Nesting == <gis file in 2d_qnl format>
```

This can be used to define polygons of mesh refinement (different levels). This is described in Section 3.1.5.

#### 3.1.4 Quadtree Control File (.qcf) – Optional Commands

When pre-processing the Quadtree mesh, a hidden 2D domain is used for areas of refinement to allow fast processing of geometry on a regular grid. The default approach is that each nesting level is treated as a domain, therefore with 3 levels of nesting the geometry control file is processed 3 times. To reduce initialisation memory demands it is possible to treat each GIS polygon in the 2d\_qnl as a separate domain for the processing of geometry inputs. This is set using the optional .qcf control file command:

```
Quadtree Mesh Processing Method == {FAST} | Memory Efficient
```

which allows changing to a more memory efficient approach to process each polygon in the 2d\_qnl layer. Whilst being more memory efficient during mesh creation, this may be slower to initialise. It has no effect on the speed of the hydraulic computations or the memory demand during the hydraulic calculations.

#### 3.1.5 Defining Mesh Refinement Polygons

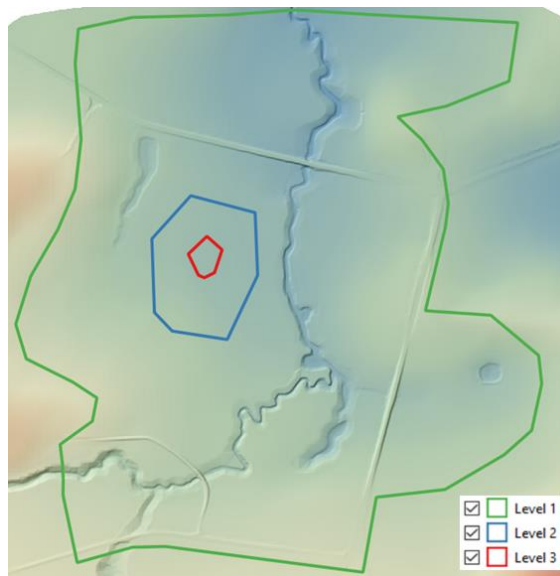
A 2d\_qnl (Quadtree Nesting Level) GIS layer is used to define the location and levels of mesh refinement. An empty (template) 2d\_qnl GIS layer can be created in the usual manner by using “Write GIS Empty Files ==”. Alternatively, the layer only requires a single attribute of Integer type, nominally called “Nest\_Level” should you wish to create the layer manually. 2d\_qnl layers should only contain polygon / region objects, with all other GIS object types (lines, polylines, points etc.) ignored.

The nesting level attribute must be in the range 1 to 9. A value of 1 indicates that the cell size to be used for that polygon is the Level 1 or base cell size (see “Base Cell Size ==” above). A value of 2 indicates the cell size within the polygon would be at Level 2 (i.e. half the base cell size). 3 would

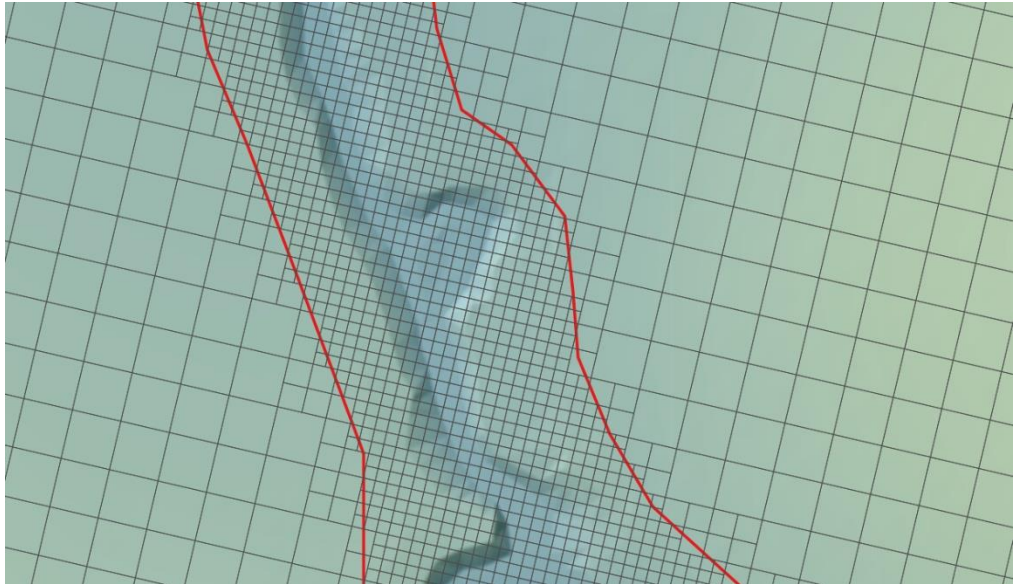
be cells at  $\frac{1}{4}$  of the base cell size, 4 for  $\frac{1}{8^{\text{th}}}$  and so on up to a maximum of 9 ( $\frac{1}{256^{\text{th}}}$ ). For numerical precision reasons, the maximum nesting level of 9 or  $\frac{1}{256}$  of the base cell size has been adopted but can in the future be increased for double precision mode should there be requests by users.

**Note: there should only be a maximum of one Level 1 polygon defined in the 2d\_qnl layer, but for all other levels there is no limit on the number of polygons.**

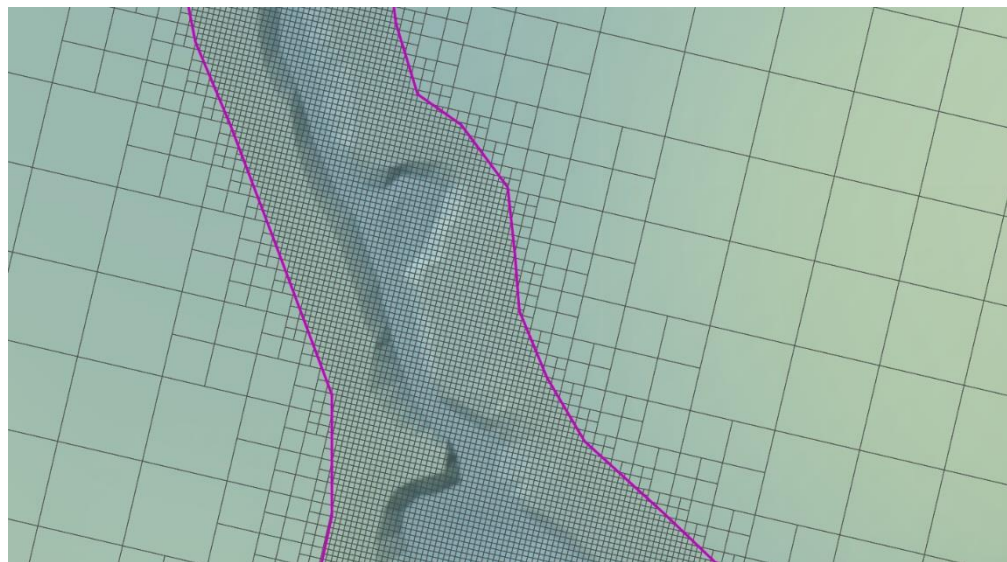
When refining mesh areas, if a refinement polygon sits within a polygon of the next higher level, e.g. a Level 3 polygon is defined within a Level 2, as per the image below, no automatic meshing is required.



If a nesting level polygon that does not sit within a polygon of the next higher level, e.g. a Level 4 polygon is defined within a Level 1 or Level 2 polygon, intermediate areas of refinement are automatically generated by TUFLOW. For example, the images below show the mesh generated when transitioning from a Level 1 to Level 3 and a Level 1 to Level 5.



**Automatic Quadtree Meshing from Level 1 to Level 3**



**Automatic Quadtree Meshing from Level 1 to Level 5**



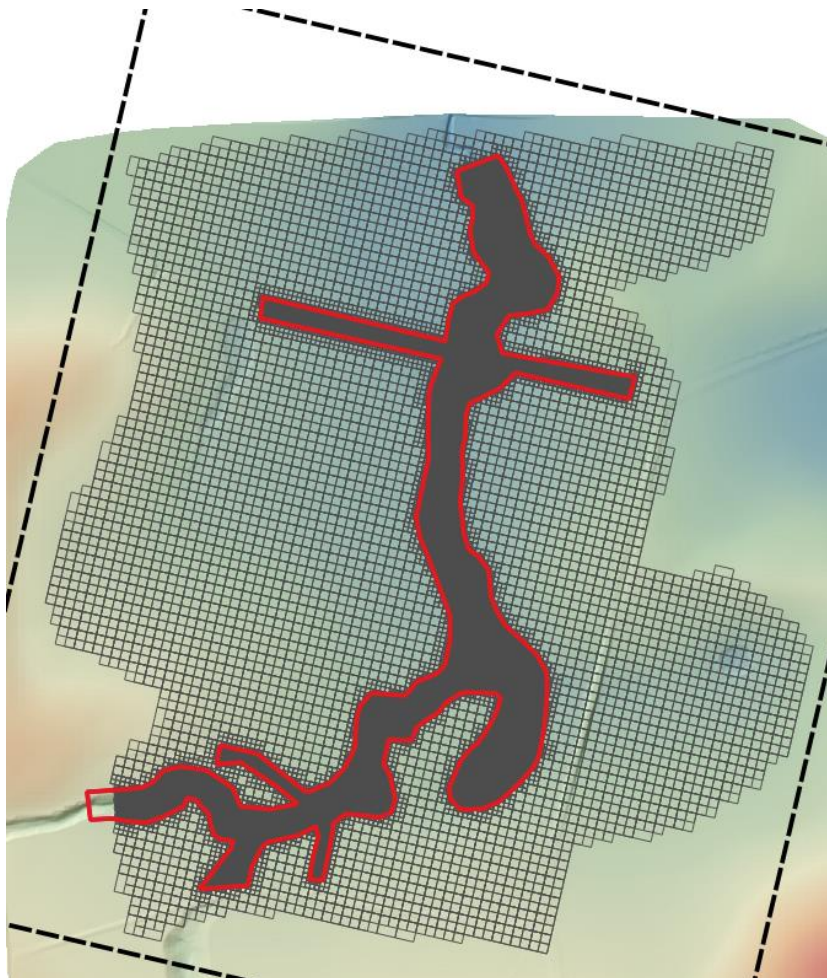
**Note:** No Level 1 polygon is required if the model origin and extent are defined in the .tgc file. In this situation the rectangle representing the .tgc computational domain is used as the Level 1 polygon. For example, if the .qcf file includes the following commands and the only 2d\_qnl polygon is Level 3 (red polygon in the image below). The mesh created is based on the rectangular computational domain in the .tgc file (as shown by the thick dashed black line) with inactive cells removed from the mesh to reduce memory.

```
Base Cell Size == TGC
```

```
Model Origin and Extent == TGC
```

```
Orientation Angle == TGC
```

```
Read GIS Nesting == gis\2d_qnl_999_R.shp
```





### 3.1.6 Quadtree Module Licence

If a Quadtree mesh requests a level greater than Level 1, then a Multiple 2D Domain / Quadtree Module licence is required. Licensees already holding a Multiple 2D Domain (M2D) Module licence for the TUFLOW Classic 2D solver can use the same module licence to simulate a TUFLOW HPC Quadtree mesh with more than one level.

If the Quadtree mesh only has Level 1, and no other levels, then a licence for the M2D/Quadtree Module is not needed, but to use the HPC Quadtree solver (rather the HPC single grid solver), `Quadtree Control File == Single Level` will need to be specified as discussed in Section 3.1.2.

Small Quadtree models with refinement can be run with the demo or free version of TUFLOW by using “Demo Model == ON” in the .tcf file. The same limitations as for a single grid apply (i.e. the total number of cells in the Quadtree mesh must be less than 30,000 regardless of refinement level and the simulation time must be less than 10 (clock) minutes.

### 3.1.7 Minor Bug Fixes and Changes for 2020-01-AB

Build 2020-01-AB includes minor bug fixes and changes for Quadtree models:

- 1 Build 2020-01-AB fixes a bug that causes simulations to stop with ERROR 2830 when creating a mesh if the .tcf command “Verbose == ON” was present.
- 2 Multiple 2d\_qnl files are not currently supported in the .qcf, build 2020-01-AB outputs ERROR 2846 if multiple “Read GIS Nesting == “ commands have been specified.
- 3 Build 2020-01-AB now correctly reports ERROR 2051 if there is an unused CN connection line in a 2d\_bc layer. For the 2020-01-AA release, if there was a CN line not snapped to a SX or HX object, the simulation would start without issuing an ERROR. The CN line was not used.
- 4 Fixed issue with Quadtree and layered flow constriction shapes, which could cause the TUFLOW simulation to crash with no error message.
- 5 Fixed an issue with If Scenario logic in the Quadtree Control File.
- 6 Added ERROR 2844 if Solution Scheme == QPC specified, this command was valid in beta builds of TUFLOW prior to the release but should no longer be used.
- 7 Ensures all input GIS layers are closed after being read. Build 2020-01-AA could produce WARNING 0255 unclosed GIS layers.
- 8 Change ERROR 2807 to WARNING 2807, for the 2020-01-AA release an error was issued, but the simulation would continue. This can occur when multiple HT or HQ boundaries apply to the same cell, in this the case the first boundary applies.
- 9 Build 2020-01-AB fixes an issue with the grid and dem\_M check files which caused the integer value of the Material ID being the order of the Material ID in the materials .csv or .tmf file. For example, if the materials file contained non sequential numbers as per the table below. For the 2020-01-AA version of TUFLOW, the check files for Quadtree would have contained the material IDs as 1,2 and 3 rather than 1, 5 and 10. This did not affect computations.

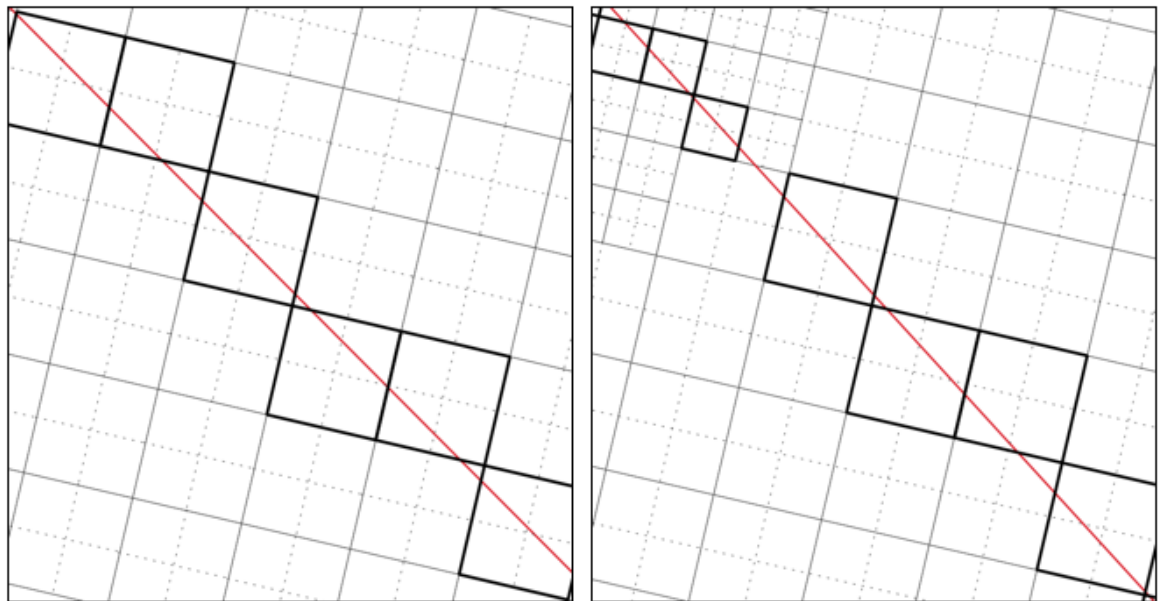
A	B
Material ID	Manning's n
1	0.06
5	0.022
10	3

### 3.1.8 Bug Fix for Quadtree for Thick 2d\_zsh Lines and Boundary Lines

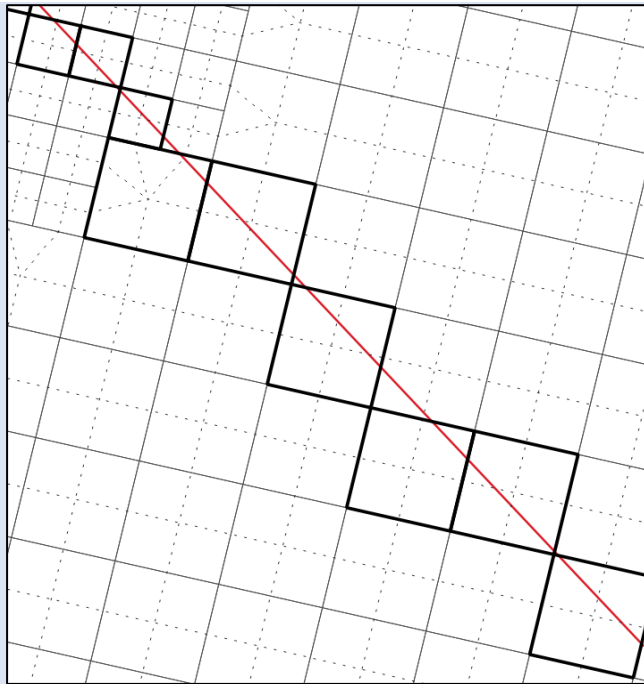
Build 2020-10-AA fixes a bug in cell selection for thick Z-Shape lines and boundary lines that could potentially produce gaps in a breakline or boundary line at areas of change in refinement. This bug did not affect thin Z-Shape lines, wide Z-Shape lines, or gully Z-Shape lines.

The issue occurred due to the crosshair method used by TUFLOW for selecting cells. This approach determines cell selection if the line intersects a cell's crosshair as shown below – the red line is the selection line (e.g. thick Z-Shape line) and the bold cells are cells selected by the line (dashed line are the cell cross-hairs).

First image demonstrates this approach on a regular grid and the second image adds a refined area. As can be seen, a gap occurs along the line in the second image caused by the line not intersecting any cross-hairs at the transition area.



Build 2020-10-AA now adopts a modified cell cross-hair approach or 'spider web' approach for Quadtree models. Instead of using cross-hairs, the 'spider web' approach draws an imaginary line from cell centre to each connected cell face. In areas of consistent cell size this will look identical to the cross-hair approach, however at transition zones there will be a line for each nested grid cell, creating a continuous intersection line and removing potential gaps in the cell selection.



Because this is a new approach, different cells may be selected at instances where a line intersects adjacent cells almost exactly when comparing between Quadtree and HPC/Classic (and between Quadtree 2020-10-AA and previous builds). This is due to numerical precision; however, effort has been made to keep the approaches as consistent as possible.

At external boundary lines where the spider web test may fail since there are no neighbouring cells on one or more sides, TUFLOW will now automatically buffer refined areas to fill gaps.



### 3.1.9 Minor Bug Fixes and Changes for 2020-10-AC

Build 2020-10-AC includes the following minor bug fixes and changes for Quadtree models:

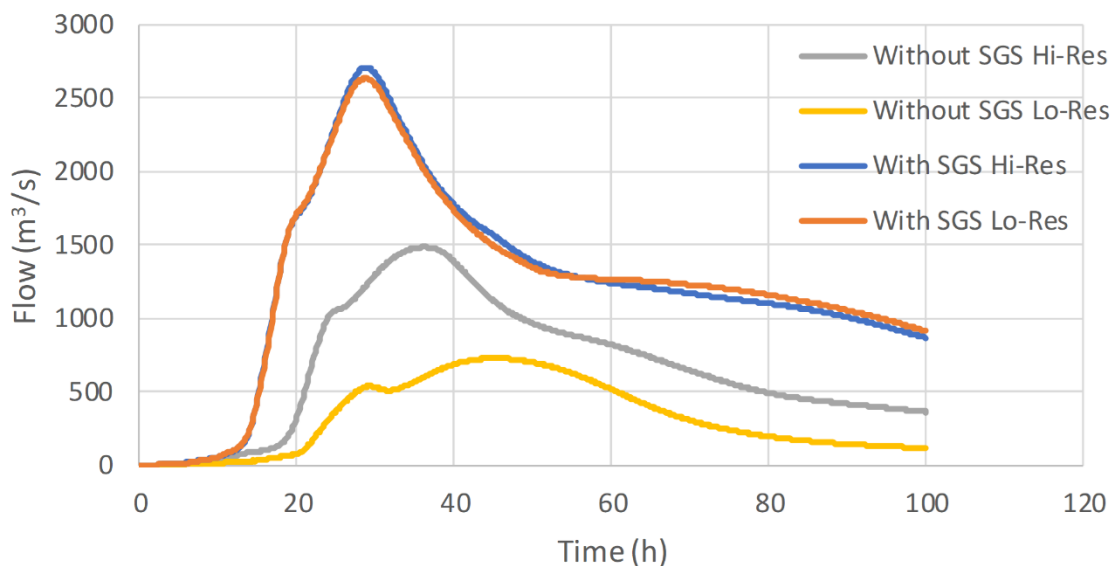
- Fixed issue that could cause "WARNING 2089 - Object ignored. Looking for first Line, Polyline or Region in 2d\_loc layer" to be spuriously generated. The location file was correctly being read and processed, the message was incorrect.

## 3.2 Sub-Grid Sampling, SGS (HPC Only, including Quadtree)

### 3.2.1 Introduction

Sub-grid sampling (SGS) stores elevation and material points sampled on a finer resolution within each 2D cell to represent the variations more accurately in terrain and land-use inside the cell, instead of each 2D cell and each 2D face having one elevation and one land-use (material) ID. Benchmarking has shown the benefits to be substantial and to be a game changer for certain types of applications such as:

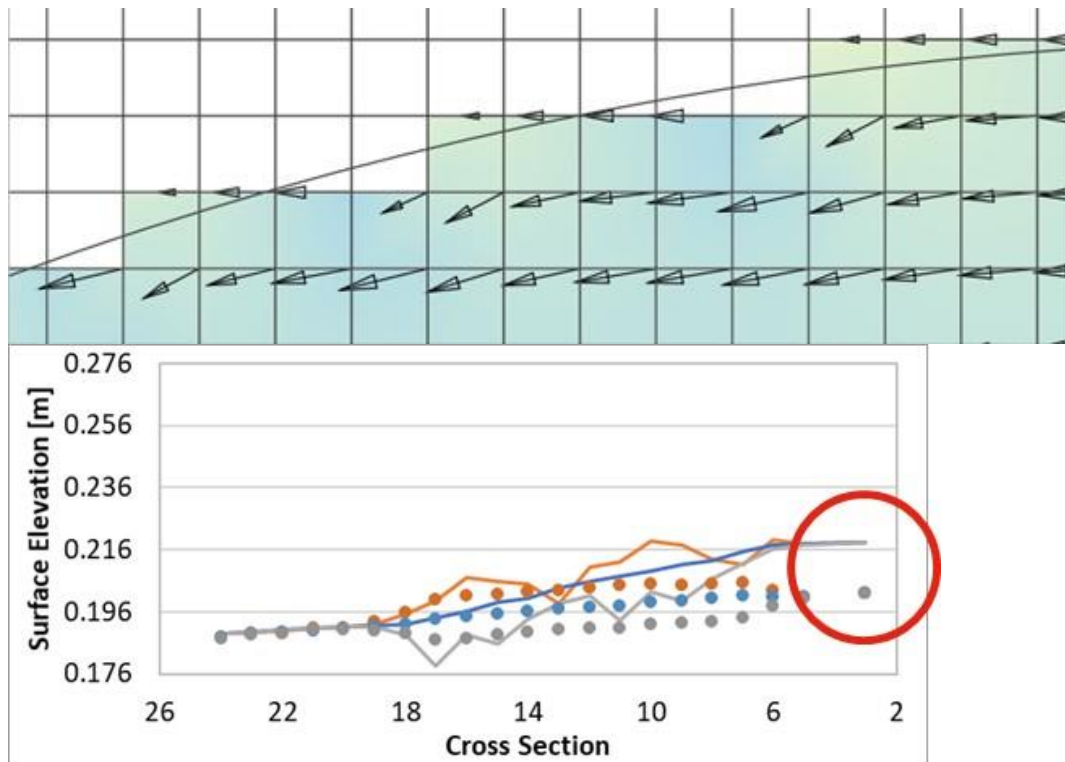
- Catchment scale models flow much more effectively with water not being “trapped” by a coarse cell resolution, and, importantly, excellent cell size convergence (ie. demonstration that by reducing the cell size(s) the model results do not demonstrably change) at much coarser cell sizes. The chart below shows the flow hydrographs for a Quadtree direct rainfall whole of catchment model using two base cell size resolutions. The Hi-Res Quadtree mesh has a base cell size half that of the Lo-Res mesh. The grey and yellow hydrographs are for without SGS and their marked difference in peak flow, shape and timing demonstrate significantly different results between the two resolutions, and therefore a cell size convergence test failure and the need for further refinement of the cell sizes (and much longer run times). In contrast, the blue and orange hydrographs are for with SGS on and show very similar results between the two resolutions, thereby demonstrating excellent cell size convergence and the ability to use the faster running Lo-Res model for day-to-day modelling.



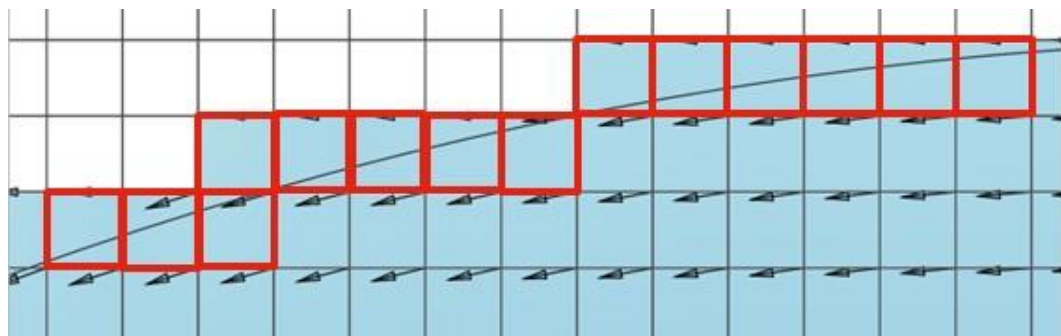
- Disturbed flow fields that can be apparent along a “saw-tooth” regular mesh wet-dry boundary completely disappear, with no spurious additional head losses generated and the results consistent with a well-designed flexible mesh. This has major benefits in that open channels can now be accurately modelled using TUFLOW HPC using coarse cell sizes at any orientation to the channel, removing the need to utilise 1D open channels carved through the 2D domain. The images and charts below show benchmarking to a U-Bend flume test for without SGS and with SGS. SGS causes a much smoother flow field to occur and importantly the head drop around the bend is correctly modelled with SGS on. Note, the red highlighted cells are partially wet cells with SGS

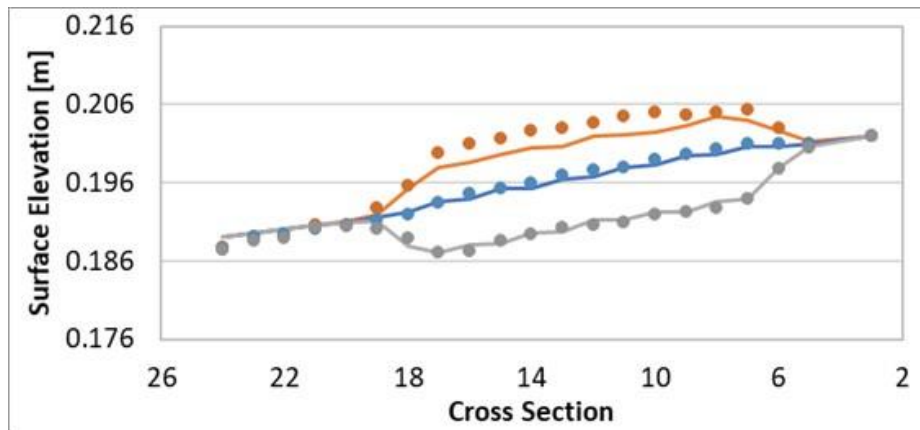
on. The charts show the longitudinal profile on the outside (orange), centre (blue) and inside (grey) of the bend with lines being modelled and points measured – as shown, with SGS off the upstream water level is overpredicted as shown by the red circle.

SGS OFF:



SGS ON:





**Note:** SGS is only available in TUFLOW HPC (with or without Quadtree). It is not available for the TUFLOW Classic 2D engine.

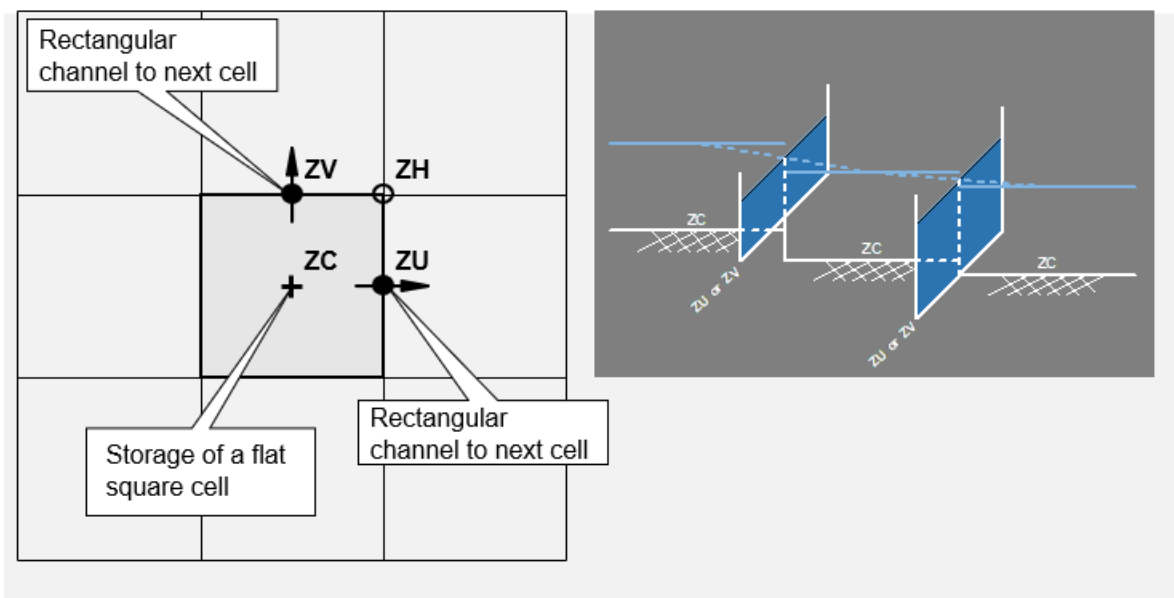
**Note:** SGS is not set as default for the 2020-01 release. However, users are encouraged to use SGS given the potentially substantial benefits thus far demonstrated through benchmarking and applications.

### 3.2.2 Without SGS (Traditional Approach)

Without SGS enabled, the cells and cell faces are represented in the conventional or traditional manner as per the diagram below. The topography of a cell is handled as follows:

- The cell volume is represented as a square bucket and calculated as the cell centre depth times the cell area.
- The flow area across a cell face is represented as a rectangular section (i.e. cell side centre depth times the cell width).
- The cell face radius value (as used in Manning's equation) is set to the depth (i.e. this is the Resistance Radius approach, which uses the flow width rather than the wetted perimeter).



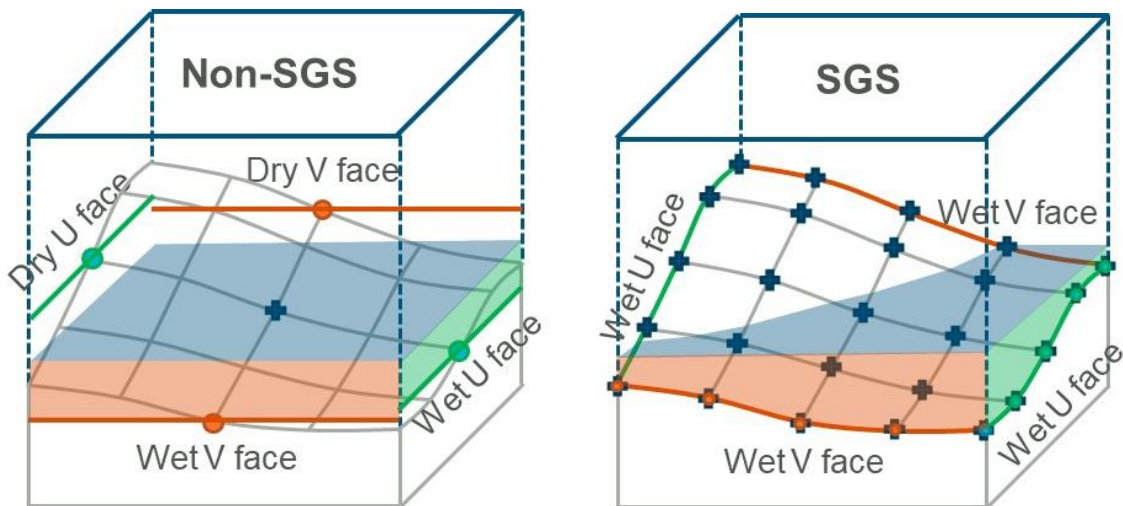


**Diagram of Standard TUFLOW Cell Architecture**

### 3.2.3 SGS Methodology

With SGS enabled for a cell the topography of the cell is handled as follows:

- The cell volume is a non-linear function of elevation (i.e. a curve of cell volume versus elevation).
- The flow area across a cell face is also a non-linear function of elevation (i.e. a curve of flow area versus elevation).
- The radius, as used in Manning's equation, is a non-linear function of elevation. There is a choice of utilising either the Resistance Radius or Hydraulic Radius approach via the command "`SGS Radius Approach == {Resistance} | Hydraulic`".
  - For Resistance Radius the radius value is equal to the flow area divided by the flow width. This is traditionally the approach used by 2D solvers and is the default setting for the TUFLOW 2020-01 release.
  - For Hydraulic Radius the radius value is calculated as the flow area divided by the wetted perimeter. As the Hydraulic Radius approach considers side wall friction it should be slightly more resistive than the Resistance Radius approach. Please note that the Hydraulic Radius approach has not undergone extensive testing at the time of the 2020-01-AA release and should be treated as under-development. In particular we will be checking for any effects on time-stepping and cell size sensitivity.



The traditional approach of a single elevation per cell centre and cell face shown on the left versus the SGS approach on the right. In the example above, with SGS all four cell faces would be active for the same water level compared with only two faces for without SGS.

The resolution at which the elevation datasets are sampled can be defined by the user. For example, with a 10m TUFLOW cell size and a 2m SGS Sample Distance the DEM is inspected using a regular 2m grid, so 25 elevation points are used to define the volume vs elevation relationship within the 2D cell, and 5 points are used for defining the area-elevation relationship for the faces

### 3.2.4 SGS .tcf Commands

To implement SGS in the control file the only command required in the .tcf is “`SGS == ON`”. However, optional .tcf commands can be used to control SGS behaviour as detailed below. For .tgc commands refer to the next section.

`SGS == ON | {OFF} ! Mandatory: Set to ON to enable the SGS functionality`

`SGS SX Z Flag Approach == Method A | {Method B}`

If set to Method A, cells that are lowered by the “Z” flag on SX connections are assumed flat (ie. as per the approach for no SGS). The default Method B retains the SGS information, but shifts it all to match the lowered elevation, as per the image below.

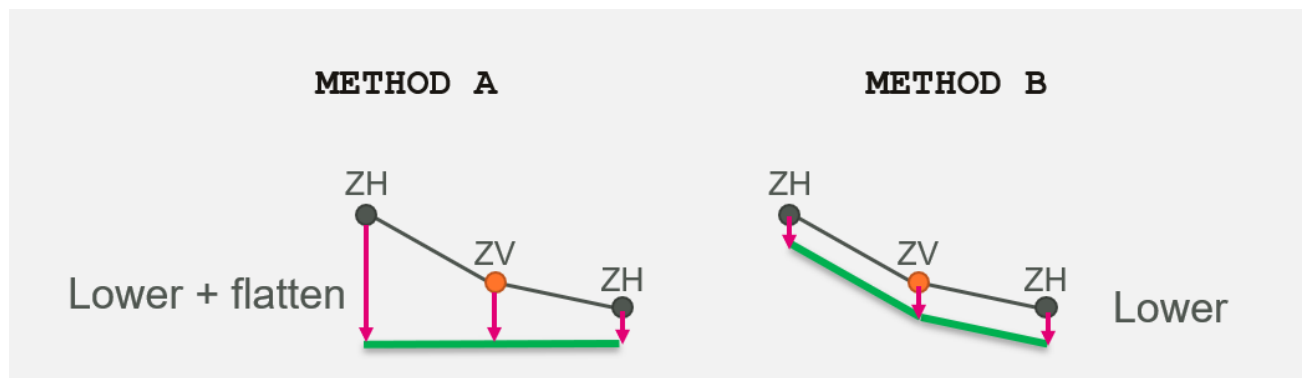
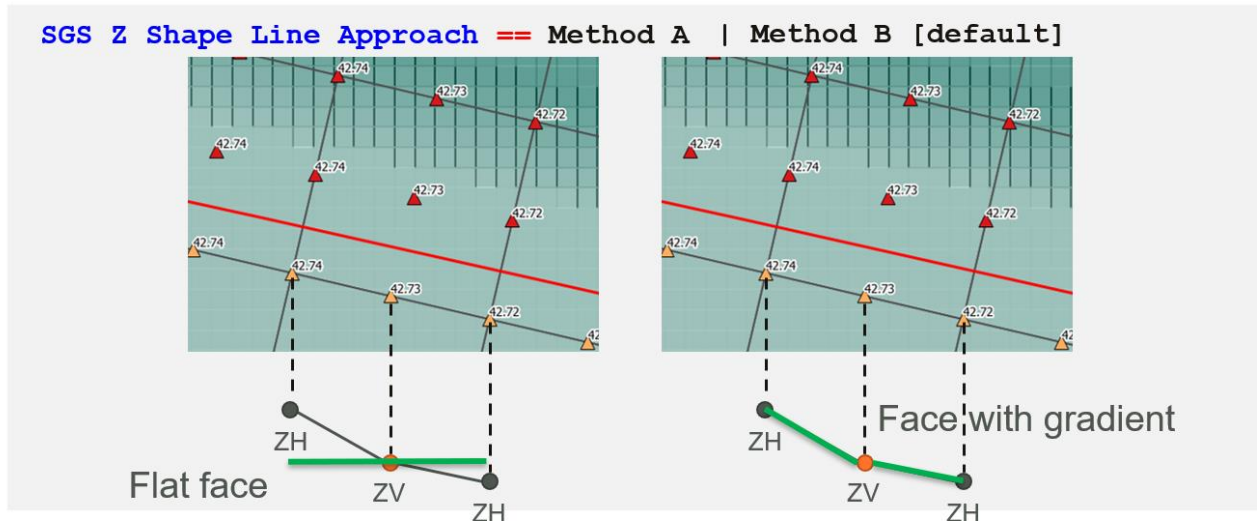


Diagram of SGS SX Z Flag Approach Options

`SGS Z Shape Line Approach == Method A | {Method B}`

If set to Method A, cell faces are assumed flat (i.e. SGS is not applied and a rectangular section / flat cell is used). The default Method B applies a gradient along the face based on the cell corners and cell side Zpt values and for thick lines uses the ZC, ZU, ZV and ZH values to apply a sloping cell area for the cell volume.



**Diagram of SGS Z Shape Line Approach Options**

`Map Cutoff SGS == <datum_or_method> | <value>`

See discussion in Section 7.5.2 for this command.

`SGS Zpt MAX/MIN Approach == IGNORE | {MINIMUM} | MEDIAN | CENTRE`

When MAX/MIN options are used in SGS .tgc commands, the minimum elevations are used to determine whether the new elevation is higher/lower than the previous one (default option, MINIMUM). However, as illustrated by the image below, the new elevation (green line) has a median elevation lower than the previous elevation (blue line), and in some situations, the green line should be considered as the “lower” elevation. This command allows users to specify which elevation is used for the geometry updates using the MIN or MAX settings.

`SGS Zpt MAX/MIN Approach == IGNORE`

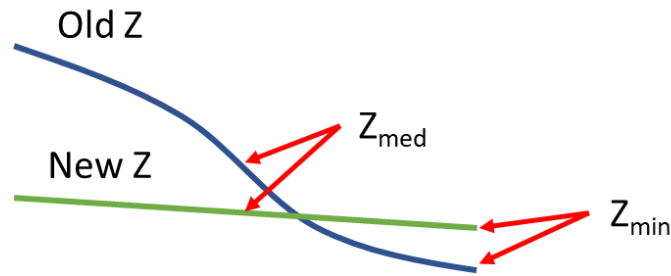
Ignores the MAX/MIN options and always applies the new elevations.

`SGS Zpt MAX/MIN Approach == MEDIAN`

Uses the median elevation for the comparison.

`SGS Zpt MAX/MIN Approach == CENTRE`

Uses the cell centre / face mid-point elevations for the comparison.



**Example of SGS Zpts comparison at a cell face**

`SGS Infiltration Approach == {AUTO} | TOTAL AREA | WETTED AREA`

This command was introduced in 2020-01-AB, it controls the cell area used when infiltration is applied with SGS. The “Total Area” option uses the whole cell area, regardless of the portion of the cell that is wet. The “Wetted Area” option uses only the wetted portion of the cell, for direct rainfall boundaries this may underpredict the infiltration. The “Auto” option uses total area if there is a rainfall boundary and wetted area if no rainfall boundaries are present. The “Auto” option is the default for the 2020-01-AB release, for the 2020-01-AA release the “Wetted Area” option was used.

`SGS Negative Rainfall Approach == TOTAL AREA | {WETTED AREA}`

This command was introduced in 2020-01-AB, it controls the cell area used when negative rainfall is applied in conjunction with SGS. The “Total Area” option uses the whole cell area, regardless of the portion of the cell that is wet. The “Wetted Area” option uses only the wetted portion of the cell. The “Wetted Area” option is the default, for the 2020-01-AA release the “Wetted Area” option was used.

**Note:** SGS .tcf commands relating to output are detailed in Section 7.5.

### 3.2.5 SGS .tgc Commands

The SGS sampling distance can be set in the geometry control (.tgc) file, with different settings for DEMs (grid or raster) and TINs as follows:

- Grid (raster) data sets:

`SGS Grid Sample Distance == <distance in metres / feet>`

The sample distance to be used for grid (raster) datasets (Read Grid Zpts ==).

Build 2020-01-AB introduces new options for dealing with raster grids when the grid only partially covers a cell or cell face. The following new commands were incorporated:

`SGS Grid Max Null Frac == Maximum Null Fraction | <0.5>`

This command controls the behaviour if the input grid only has partial coverage and the existing elevation in the cell has not been initialised either with a Set Zpts == command or with an elevation in a previous dataset. If the fraction of the cell that has no value (null) in the input grid is above this value then the zpt is not updated.

`SGS Partial Grid Update Null Frac == Lower Limit, Upper limit | <0.1, 0.9>`

This command controls the behaviour if the input grid only has partial coverage and the cell has been initialised either with a Set Zpts == command or with an elevation in a previous dataset.

This sets lower and upper limits for the fraction of the SGS values that can be null in the grid. This applies to both cells and cell faces. The default values for the lower limit is 0.1 and for the upper limit 0.9. The limits are used as per the below:

- If the null fraction is below the lower limit, TUFLOW applies the values from the new DEM.
- If the null fraction is between the lower and upper limits, update the null value from current ZC ZU ZV and ZH values.
- If the null fraction is higher than the upper limit, do not update the Zpt.

The above changes the way the elevations are processed. This will force a regeneration of any XF files for grid inputs created by previous versions of TUFLOW.

- TIN data sets:

`SGS TIN Sample Distance == <distance in metres / feet>`

The sample distance to be used for TIN datasets (Read TIN Zpts ==).

Alternatively, the sample distance for both Grid and TIN datasets can be set using

`SGS Sample Distance == <distance in metres / feet>`

to set both the grid and TIN sample distances.

For grid inputs, if no `SGS Grid Sample Distance` or `SGS Sample Distance` has been set, the resolution of the DEM is used by default. For TIN datasets, either `SGS TIN Sample Distance` or `SGS Sample Distance` is mandatory.

These commands can be used repeatedly throughout the .tgc file to vary the sampling distance for different elevation data sources (the last occurrence of these commands prior to the data source is used).

**Note:** the sample “frequency” is capped at 31 by default to avoid long pre-processing time. In general, a sample frequency smaller than 31 is sufficient for most of natural water ways or artificial structures, and users may not benefit from applying a super fine sample distance against the model cell size. This upper limit can be increased by using a second argument in the “`SGS Sample Distance`” command, e.g. “`SGS Sample Distance == 1 | 51`” sets the sampling distance to 1 m and the frequency limit to 51 sample points per face (2601 per cell). However, a hard limit of 127 per face (16,129 per cell) still applies.

**Note:** Not all topography commands are SGS compliant yet. The following table summarises the status of available features.

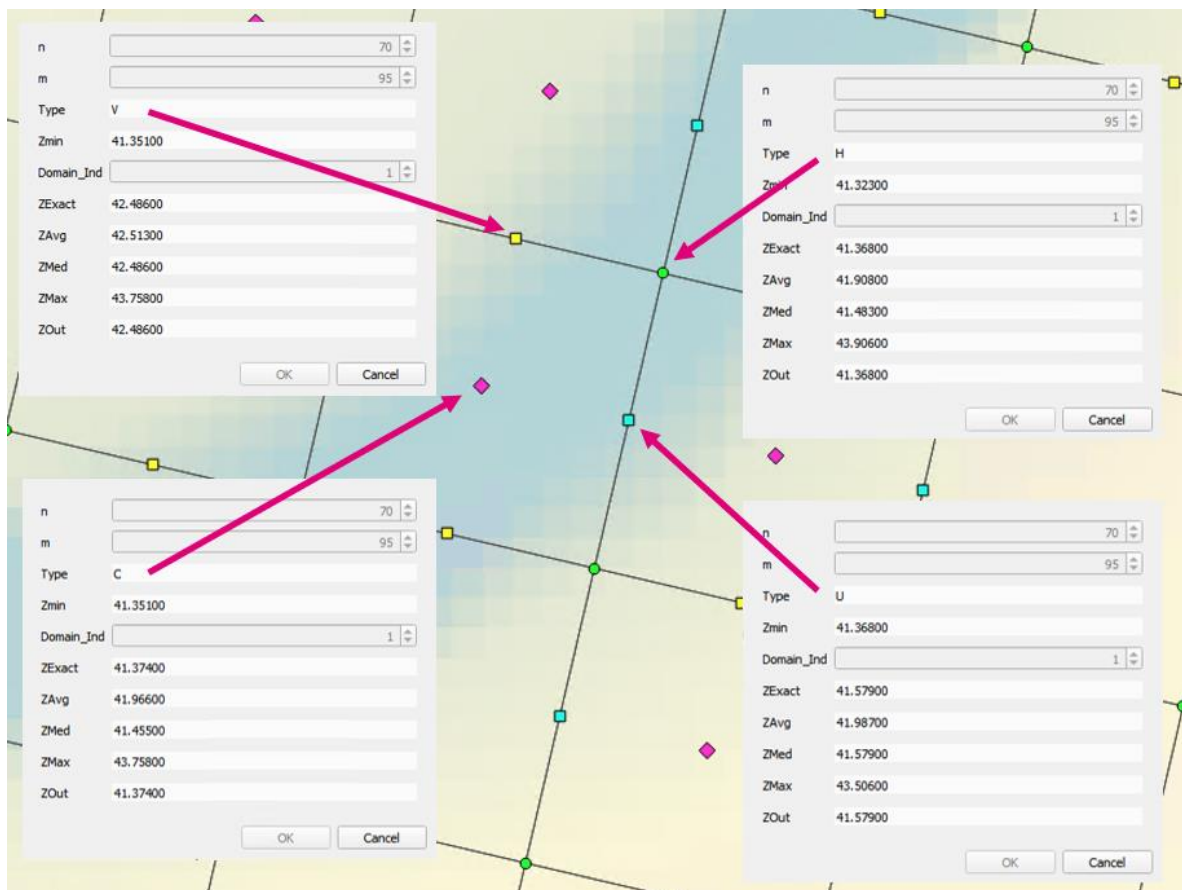
Full SGS Sampling using Sample Distance	Uses ZC/ZU/ZV/ZH values	Sets Flat Cell/Face	Unsupported Commands as of Build 2020-01-AA
Create TIN Zpts Read Grid Zpts Read TIN Zpts Read GIS Z Shape (Regions) Read GIS Layered FC Shape (Regions)	Read GIS Z Shape (Breaklines) Read GIS Layered FC Shape (Breaklines) Read GIS Z Line Read GIS Zpts Read GIS Layered FC Shape (Breaklines)	Set Zpt Read GIS Variable Z Shape Read GIS Z Shape Route 1D Nodes with SXZ flag 1D Pits with SXL flag	Read GIS Z HX Line <del>Read GIS Z Shape Route</del> Read GIS FC Shape Read GIS Zpts Modify Conveyance Read RowCol Zpts Interpolate ZC/ZHC/ZUV/ZUVC/ ZUVH Set Code with Clean Zpt ZC == MIN(ZU,ZV)

### 3.2.6 SGS Check File Output: zpt

When running a model with SGS enabled, if the \_zpt check layer is output, then additional attribute information is provided. If a cell has SGS applied the attributes are:

- The “Zmin”, previously called “Elevation”, attribute for ZC points now represents the minimum elevation within the cell and for ZU/ZV points along the cell face. Note, these points are still located at the centre of the cell or cell face, but the minimum value is not necessarily at this location.
- “ZExact” is the elevation at the exact location of cell centre, face mid-point, or cell corner.
- “ZAvg” is the average (mean) elevation of the sampled values.
- “ZMed” is the median elevation of the sampled values.
- The “ZMax” attribute is the maximum elevation, i.e. the elevation at which the cell area or cell face flow width is fully wet.
- “ZOut” is the elevation used for the SGS Depth Output (Section 7.5.3)

When running a model with SGS enabled, if the \_zpt check layer is output, then additional attribute information is provided. If a cell has SGS applied the attributes are:



Example SGS Check File Output

### 3.2.7 SGS Check File Output: DEM\_Z

For Build 2020-01-AB, when running a model with SGS enabled, if the `_dem_z` check layer is output then two elevation grids are produced:

- `_DEM_Z` – contains a raster based on the SGS elevations used for depth output interpolation ([SGS Depth Interpolation Approach ==](#) ). Refer to section 7.5.3 for more details on the depth interpolation approach.
- `_DEM_Zmin` - contains a raster based on the minimum SGS elevations.

### 3.2.8 Map Output and Re-Mapping

For map output considerations and options for handling partially wet cells see Section 7.5.

**Note:** For 2020-10-AB, a new 2022 release beta feature is available to output high resolution raster outputs from TUFLOW as outlined in Section 3.7.4.

The remapping of map output if using SGS to a fine resolution DEM is being built into TUFLOW as a direct output option for a future release update (see note above). In the meantime, remapping can be carried out using the `asc_to_asc` utility using ASC and FLT formats (with NetCDF in a future update to `asc_to_asc`). The workflow to carry this out is documented in Section 7.6.



### 3.2.9 Treatment of Infiltration and Negative Rainfall with SGS Enabled

With SGS enabled, cells that are partially wet the treatment of infiltration and direct rainfall is as follows. See Section 3.2.4 above for details on the SGS Infiltration Approach and SGS Negative Rainfall Approach options.

- For positive rainfall, i.e. rainfall on to the 2D cell, the volume source for each cell is the total cell area times the rainfall irrespective of whether the cell is partially wet or not.
- For negative rainfall (evaporation), the volume of evaporation is factored by the wet area fraction of the cell. That is, if the cell is only 10% wet, only one tenth of the cell's total area contributes to the negative source term.
- For models with soil infiltration the infiltration rate is proportional to the wet area fraction of the cell. However, initial infiltration losses are based on the total area of the cell (i.e. infiltration will proceed at the maximum possible rate until the cumulative infiltration – also based on total cell area - equals the initial loss value) even if the infiltration occurs with the cell partially wet. This approach is adopted to conform with that required for direct rainfall, which assumes the rainfall is applied over the entire cell irrespective of whether the cell is partially wet or not. Likewise, soil capacity is based on the total cell area (i.e. infiltration will cease once the cumulative infiltration equals soil capacity), and the cumulative wet time for the Horton model will also increment for cells that are partially wet.

### 3.2.10 Bug Fix for Read TIN Zpts

Build 2020-01-AB fixes a bug with SGS and Read TIN Zpts. For the 2020-01-AA release models with XF files on (default) would crash when processing Read TIN commands. With XF files off, the Read TIN Zpts would process but not update the elevations.

Models with Read TIN Zpts and utilising SGS should upgrade to the 2020-01-AB (or newer) release. Any XF files associated with Read TIN Zpts should be deleted.

Build 2020-10-AC fixes a memory allocation issue reading a 12d TIN when the TIN is significantly larger than the model domain.

### 3.2.11 Bug Fix for Z Shape Polygon with “Add” Option.

Build 2020-01-AB fixes a bug with the elevations at cell corners if using SGS in combination with Z shape objects with the “Add” shape\_option attribute. This issue does not affect the hydraulic calculations as it only changed the cell corner elevations which are not used computationally. However, this could cause incorrect depths to appear in the map output at the cell corners.

Build 2020-10-AB fixes an issue where polygons with the “ADD” option read by the “[Read GIS Z Shape](#)” or “[Read GIS Zpts](#)” commands modifies sub-grid elevations multiple times if a Zpt point is “snapped” to the polygon vertices.

### 3.2.12 Bug Fix for Corner Output with SGS and Variable Z Shape Output

Build 2020-01-AB fixes an issue with the cell corner elevation which is only used for output if using SGS with variable Z Shape objects. The cell corner elevations are not used computationally, however, this could cause incorrect depths to be output.

Build 2020-10-AB fixes a similar cell corner elevation output issue for Quadtree meshes with variable Z Shape objects.

### 3.2.13 SGS and Double Precision TUFLOW

Due to an Intel compiler issue; grid inspection (Read Grid Zpts ==) with SGS on, has compiler optimisation disabled (just for the grid inspection step) when running the double precision version of TUFLOW. These means that the grid inspection may be much slower than the single precision version. If XF files are enabled (the default) subsequent simulation initialisation will use these and no signification difference between start up time for single and double precision version should be observed.

**Note:** As per the TUFLOW manual, HPC rarely requires the use of double precision. Therefore, build 2020-01-AB now outputs WARNING 3525 if performing a grid inspection with SGS enabled using a double precision version of TUFLOW.

As of Build 2020-10-AB, the above issue with slow initial processing time for SGS for double precision simulations has been resolved and therefore WARNING 3525 has been removed.

### 3.2.14 Bug Fix for SGS with TINS Smaller than Cell Size

2020-10-AA fixes an issue where small TIN surfaces completely inside a cell with SGS did not modify SGS elevations.

### 3.2.15 Bug Fix for SGS with Layered Flow Constrictions.

2020-10-AA fixes an issue where SGS enabled models with any layered flow constriction polygons could fail to initialise. If the model had neither a “SGS Sample Distance == “ or “SGS TIN Sample Distance == “ command defined, when reading the layered flow constriction polygons TUFLOW would report an access violation error and crash.

### 3.2.16 Bug Fix for SGS and Variable Z Shape

Build 2020-10-AA fixes an issue with SGS models with variable Z shapes. For cells with variable Z shape applied SGS is not used, and the cell/cell face is defined by a single elevation. The elevations applied in variable Z shape GIS layer control the final elevations after the variable z shape has been triggered. The pre-failure elevations are the elevations defined in the geometry control file before the variable Z shape is read in.

Prior to the 2020-10-AA release the pre failure elevation used the SGS minimum elevation, from 2020-10-AA onwards the exact (elevation at the location of the cell centre / face) elevation is used.

### 3.2.17 Bug Fix for SGS Models with very Large TINs

Build 2020-10-AB fixes an issue when reading a very large TIN elevation dataset with SGS enabled that would cause a “Stack Overflow” error and the simulation to terminate. The only reported occurrence of this happened with a TIN dataset consisting of more than 120 million points.

### 3.2.18 Bug Fix for SGS Models with Gully Lines and Read GIS Zpts Polygons

Build 2020-10-AB fixes an issue where the Zpts point selected by gully lines and Read GIS Zpts polygons use minimum elevations instead of exact elevations at cell corners to update SGS curves. This inadvertently made the cell/face elevations lower than they should be and created higher conveyance along the gully lines and near the Read GIS Zpts polygons. This fix may increase the modelled water level near these features, and for this reason it is recommended to conduct model sensitivity test for SGS models with gully lines or Read GIS Zpts polygons. No backward compatibility is provided for this change.

**Note:** If using SGS, it is generally recommended not to use gully lines as the SGS sampling will pick up the low flow paths through cells and compute a much more accurate low flow conveyance. The exception would be where the quality of the DEM or TIN along gullies is poor, and the user needs to enforce a continuous low flow path.

### 3.2.19 Change to “Zout” Attribute in \_zsh\_zpt\_check File

For consistency between SGS and non-SGS models, Build 2020-10-AB now reports the exact elevations to the “Zout” attribute in the \_zsh\_zpt\_check layer. The “exact” elevation refers to the elevation sampled exactly at the cell centre for ZC points, the cell mid-sides for ZU and ZV points, and the corner elevations for ZH points. Prior to Build 2020-10-AB, the minimum elevation inside the cell for ZC and along the cell sides for ZU and ZV were output if using SGS. This change was made for the Beta4 version of 2020-10-AB release.

### 3.2.20 SGS Approach == Method C (2022 Beta Feature)

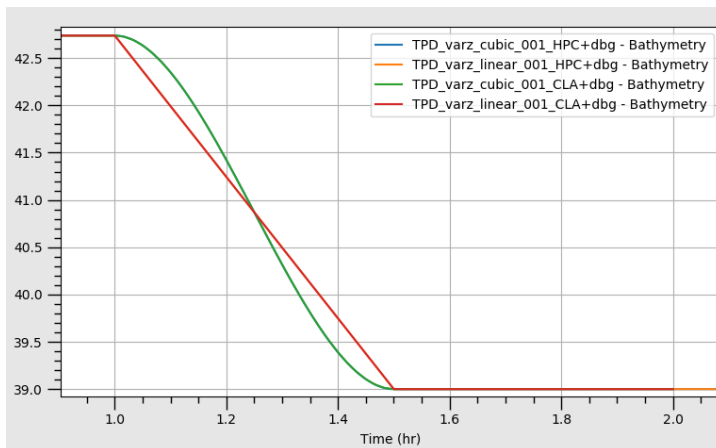
An enhanced approach for SGS referred to as Method C is available for testing. The new approach has superior treatment of 2D cells partially covered by a data layer, includes sampling of Manning’s n values along cell faces, a potentially missing breakline detection feature and facilitates the new high resolution mapping feature. Refer to Section 3.7 and Section 3.7.4 respectively for details.

## 3.3 Non-Linear Failure of Variable Z Shapes

Variable Z shapes can now be failed using a temporal cubic transition, as can be needed for dam or embankment failures. The cubic transition:

$$z(t) = (1 - a)z_0 + az_1 \quad a = (3 - 2x)x^2 \quad x = \frac{t-t_0}{t_1-t_0}$$

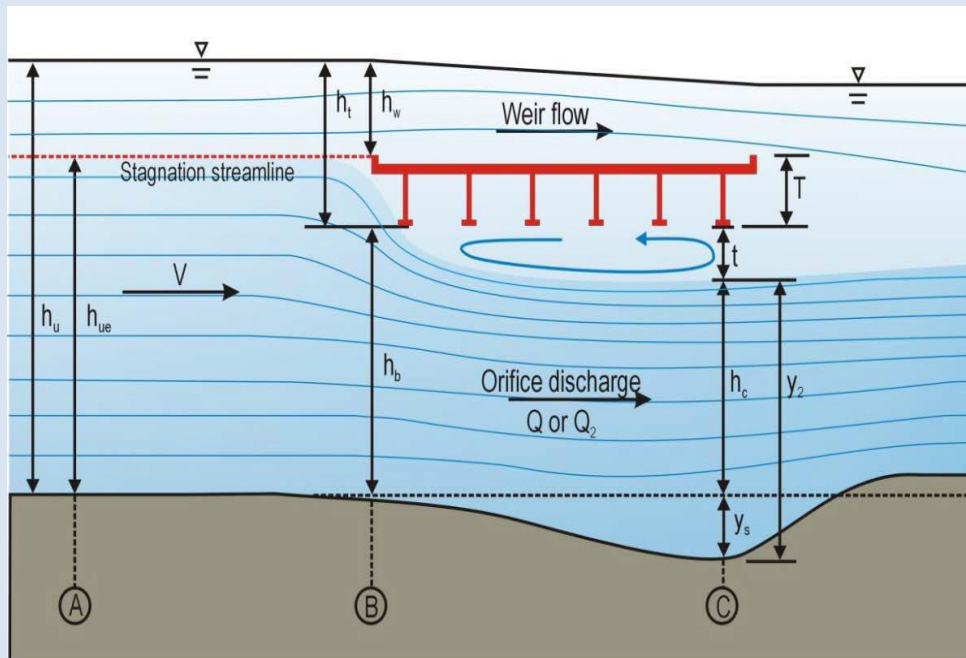
If “Cubic” is specified in “Shape\_options” attribute for the 2d\_vzsh object a cubic transition is applied. An example of a linear (red line) and cubic transition (green line) are illustrated in the chart below.



### 3.4 2D Layered Flow Constrictions - 2D Bridge Decks

#### 3.4.1 CFD Benchmarking Study

A joint research study between the Queensland Department of Transport and Main Roads (DTMR) and TUFLOW to provide better functionality for modelling bridge decks that are surcharged, under pressure flow or drowned out (see image below), is being undertaken. Preliminary results are promising with the next stage to benchmark against field measurements for which DTMR have installed gauges at a low level bridge.



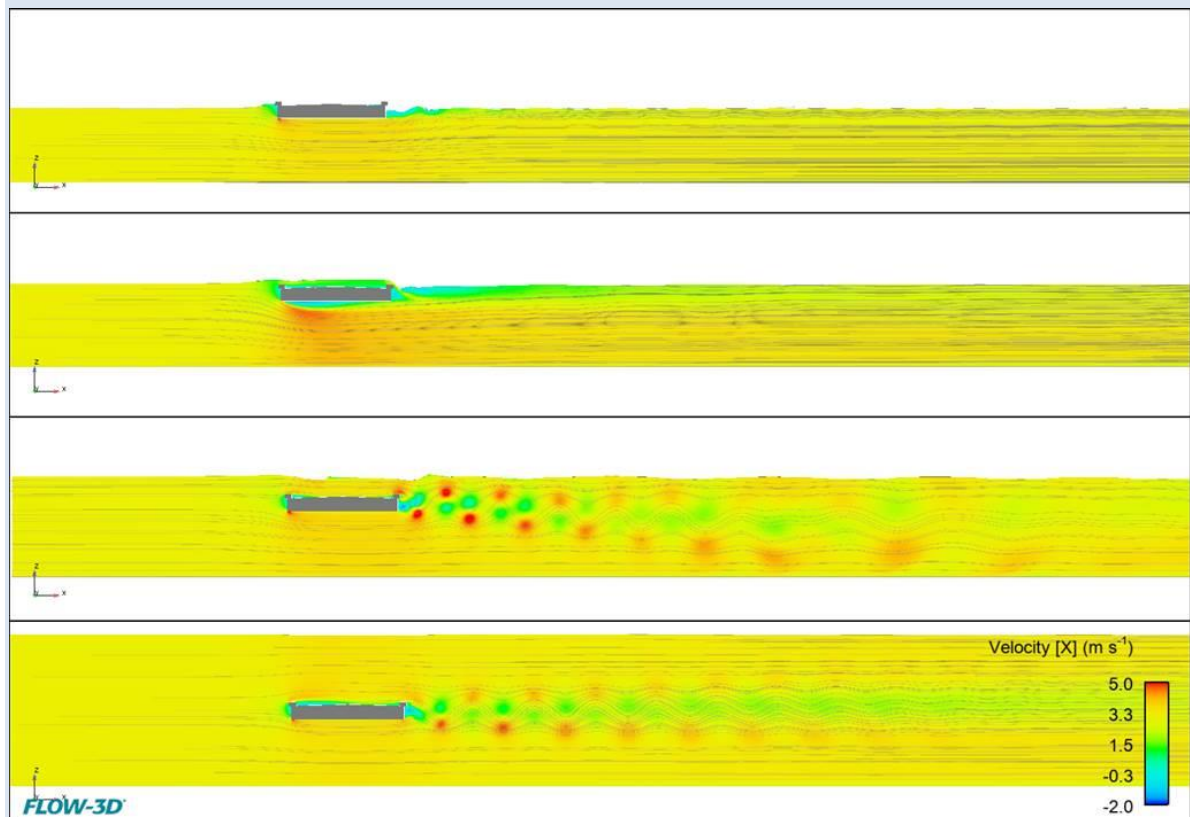
Thus far, DTMR have carried out Flow3D CFD simulations across a range of deck dimensions and deck to depth ratios for a solid deck configuration to determine head losses for flow surcharging against or over a bridge deck, including pressure flow conditions.

Promisingly, a reasonably consistent relationship and shape of the energy loss versus the deck depth, downstream water level and other parameters has been observed. For the 2020-10-AA release two

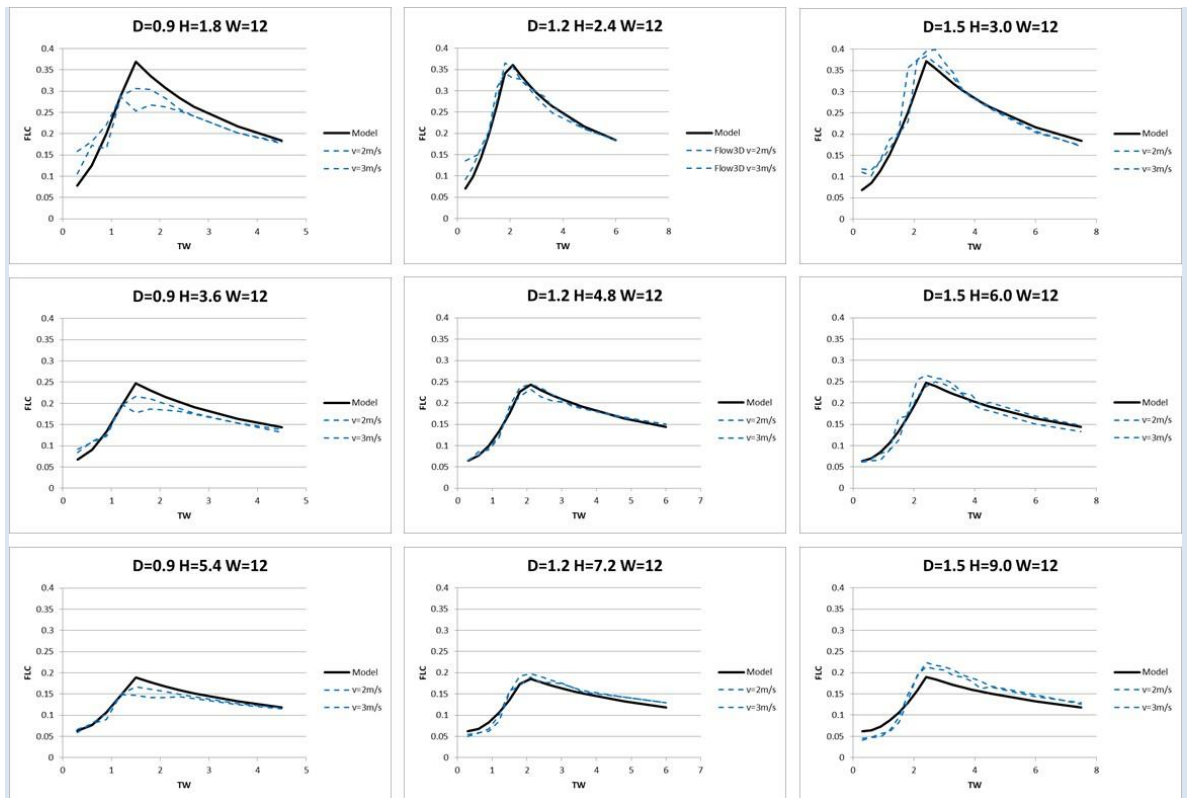
new methods, described in Section 3.4.2, have been incorporated into Layered Flow Constrictions to allow the modeller to reproduce this behaviour, with METHOD D allowing the modeller to control the point of maximum energy losses.

The charts below show the results from a range of deck configurations for various flow stages (CFD results are the blue lines and the black line is an initial curve).  $D$  is the deck depth (thickness),  $H$  is the depth below the deck and  $W$  is the width of the deck (in the direction of flow). As can be seen, the energy loss coefficient (FLC) peaks from around 0.3 to 0.4 for the lower set bridges (high  $D/H$  ratio) compared to 0.15 to 0.27 for higher set bridges (low  $D/H$  ratio). Of interest is the peak energy loss occurs at tailwater depths above the deck, typically at 50 to 100% of the deck depth  $D$ .

We are continuing to test and benchmark the new 2d\_LFC approaches and provide improved guidelines on modelling bridges in 2D. If you or your organisation has any suitable data (flume or real-world) that can be used for benchmarking, please contact [support@tufLOW.com](mailto:support@tufLOW.com). The data would need to provide reliable estimates of the flow rate and water levels upstream and downstream of the bridge.



**CFD Modelling of Surcharged Bridge Deck**



**Charts of Energy Loss (FLC) versus Tailwater Depth from CFD Modelling (Blue Lines) and a Parameter Fit Model (Black Lines)**



### 3.4.2 New 2d\_LFC Approaches for Modelling Pressure Flow

Build 2020-10-AA introduces two new layered flow constriction approaches to modelling form (energy) losses within the vertical that provides better options for representing surcharging, submergence and upstream controlled pressure flow of bridge decks. Pressure flow occurs from water surcharging against a bridge deck causing orifice type flow, which eventually becomes drowned out with higher water levels. During the period of pressure flow, higher energy losses are required to simulate the orifice flow.

The new approaches, labelled METHOD C and METHOD D, are based on initial observations from a joint R&D exercise currently underway between the Queensland Department of Transport and Main Roads (DTMR) and TUFLOW as discussed in Section 3.4.1.

To provide greater clarity, the CUMULATE approach can now also be referred to as METHOD A and PORTION as METHOD B.

**Note 1: The default setting can be controlled using**

**Layered FLC Default Approach == [ CUMULATE | {PORTION} | METHOD C | METHOD D ].**

**Care must be taken as there can be significant changes in the attribute values switching between different approaches (see example below). It is strongly recommended that the approach adopted be hard-coded into the model for each structure by entering the approach into the 2d\_Lfc Shape\_Options attribute.**

**Note 2: Whilst the default approach of using PORTION is unchanged for backward compatibility purposes, it is recommended the new approaches of METHOD C or METHOD D be considered as they are shown to emulate behaviour from CFD modelling more closely for pressurised flow conditions. METHOD C and METHOD D overcome the tendency of CUMULATE to overestimate losses once the bridge becomes drowned out and for PORTION to have a greater dependency on bridge deck and rail FLC values that are depth dependent due to the portioning of losses with depth.**

METHOD C effectively combines the CUMULATE and PORTION approaches by utilising CUMULATE through to the top of Layer 3 and PORTION above Layer 3. METHOD C adjusts the FLC value in the vertical as follows.

- Water level below Layer 2: The same result as for all other approaches, i.e. a form loss based on that specified for Layer 1 is applied. This is typically used for the energy losses associated with bridge piers.
- Water level below top of Layer 3: Same approach as for CUMULATE.
- Water level above Layer 3: Gradually reduces the energy loss with increasing depth by proportioning with depth. The energy loss is calculated as that accumulated from Layers 1 to 3 pro-rated by the depth of Layers 1 to 3, plus no losses pro-rated by the depth above the Layer 3.

$$\zeta_{total} = \left( \zeta_1 + \zeta_2 \frac{y_2}{D_2} + \zeta_3 \frac{y_3}{D_3} \right) \frac{(y_1 + y_2 + y_3)}{y_{total}}$$

$$y_{total} = y_1 + y_2 + y_3 + y_4$$



Where  $\zeta_{total}$  is the overall form loss coefficient applied,  $\zeta_n$  is the form loss coefficient of layer  $n$ ,  $y_n$  is the water depth at layer  $n$ ,  $D_n$  is the layer  $n$  thickness.

METHOD D is similar to METHOD C but allows the modeller to control the depth at which the losses start to reduce when the flow transitions between pressure flow and drowned flow. Of particular note for METHOD D is that the L2\_FLC attribute value now refers to the energy loss caused by pressure flow against the Layer 2 and 3 combined (i.e. the bridge deck and rails), and the L3\_FLC attribute value is used to set the inflection point (IP) at which the transition from pressure flow to drowned flow commences. (These attributes have now been labelled as L2\_or\_L23\_FLC and L3\_FLC\_or\_fIP if empty files are created.) L3\_FLC is specified as a factor of the combined depths of Layers 2 and 3, with observations thus far indicating this inflection point is around 1.5 to 2 times the depth of the bridge deck (assuming no rails). The effect of partial blockage (e.g. 20% blockage of Layer 3 representing the bridge deck rails) is to reduce the inflection point proportionally by the blockage amount.

METHOD D adjusts the FLC value in the vertical as follows.

- Water level below Layer 2: The same result as for all other approaches, i.e. a form loss based on that specified for Layer 1 is applied. This is typically used for the energy losses associated with bridge piers.
- Water level between invert of Layer 2 and below the Inflection Depth measured from the invert of Layer 2. The Inflection Depth and total energy loss are calculated as follows:

$$D_{IP} = f_{IP}(D_2 Blockage_2 + D_3 Blockage_3)$$

$$\zeta_{total} = \left( \zeta_1 + \zeta_2 \frac{y_I}{D_{IP}} \right) \frac{(y_1 + y_{Inflection})}{y_{total}}$$

Where  $D_{IP}$  is the Inflection Depth discussed above,  $f_{IP}$  is the factor entered into the L3\_FLC attribute to set the inflection point,  $y_{Inflection}$  is the water depth at the inflection point.

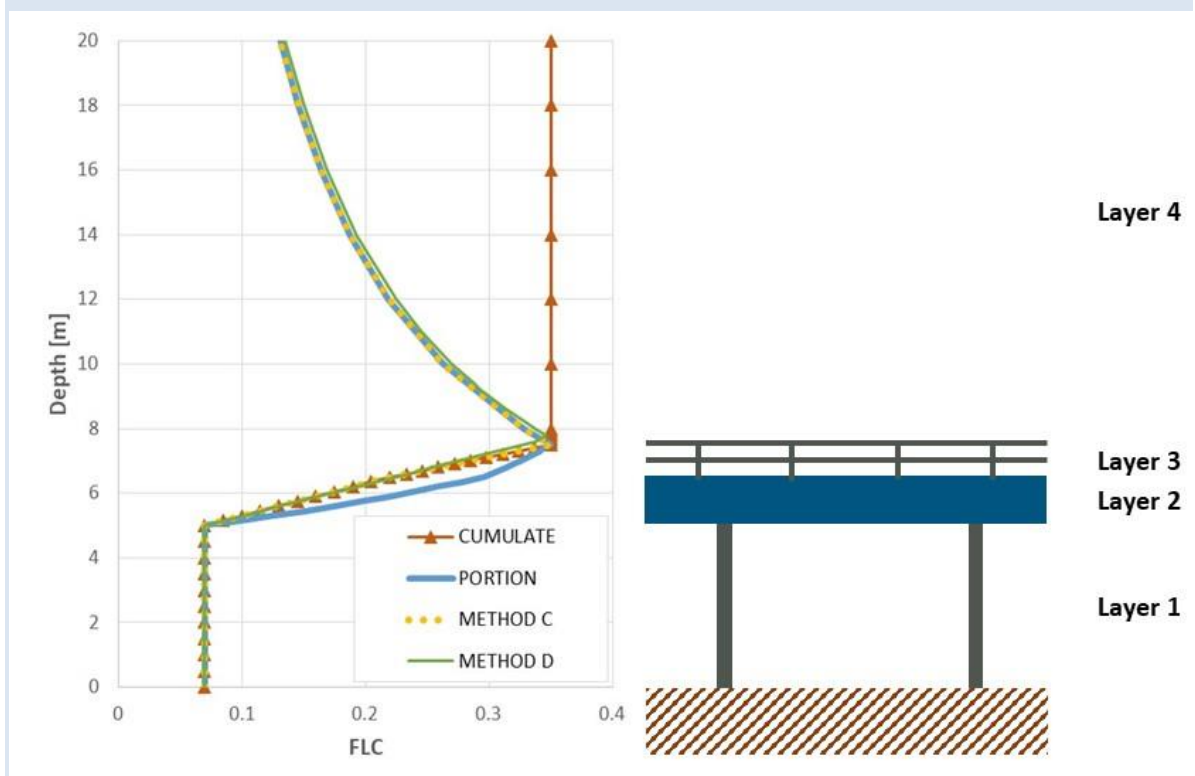
- Above the Inflection Point the energy gradually reduces with increasing depth in a similar manner to PORTION and METHOD C to simulate the transition to drowned flow and tendency to zero energy losses with increasing depth over the bridge deck.

**Example 1:** This example, taken from a calibration of a bridge structure on the Iowa river that is being surcharged with water slightly overtopping the deck, requires a form loss coefficient of 0.35 to match the observed head loss. Of interest is that a form loss of 0.35 is consistent with the CFD modelling described in Section 3.4.1. The attribute values for each method shown in the table below have been adjusted to target the 0.35 so as to achieve calibration. The figure further below compares how the form loss value varies with height for the four methods.

As can be seen, METHOD C applies exactly the same overall form loss as the CUMULATE approach below Layer 4, but reduces proportionally with the depth above Layer 3. For METHOD D the Inflection Point is located  $1.6 \times (1.5 \times 1.0 + 1.0 \times 0.2) = 2.72$  metres above Layer 2 invert, and therefore follows a slightly different path. Also, of note is that the sum of the FLC values for CUMULATE, METHOD C and METHOD D are the same. i.e. 0.35, whereas for PORTION much higher values are needed for Layers 2 and 3 due to the proportioning with depth.

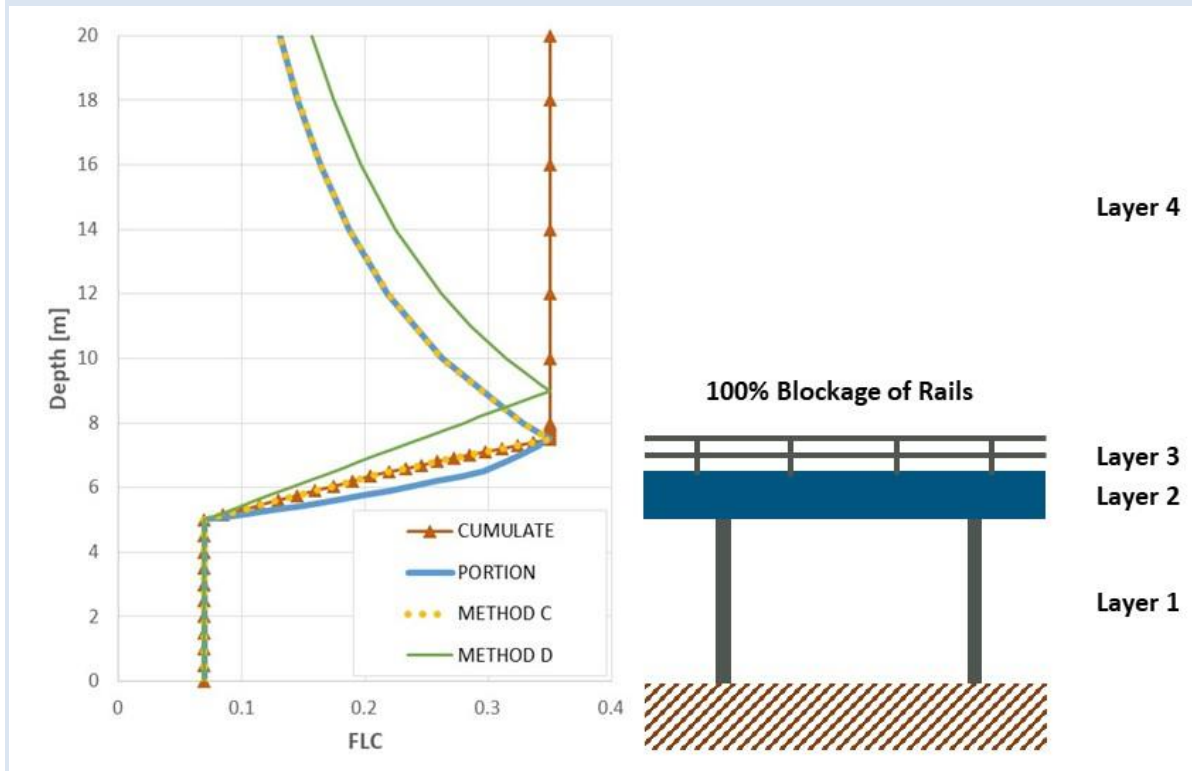
			FLC Attribute (See note for METHOD D above)			
	Depth (m)	Blockage (%)	CUMULATE (METHOD A)	PORTION (METHOD B)	METHOD C	METHOD D
Layer 1	5.0	5	0.07	0.07	0.07	0.07
Layer 2	1.5	100	0.15	1.05	0.15	0.28
Layer 3	1.0	20	0.13	0.70	0.13	1.6*

\* Note this is not a form loss, refer to description above for this parameter.



**Example 1: Layered Flow Constriction Approaches**

**Example 2:** One of the advantages of METHOD D is if a 100% rail blockage of the rails needs to be considered, METHOD D automatically raises the Inflection Point to a higher level as shown in the figure below.



A summary of the attribute description for METHOD C and METHOD D are outlined in the following table:

Default GIS Attribute Name	Description
<b>Read GIS Layered FC Shape Command</b>	
Invert	<b>Method C:</b> Unchanged <b>Method D:</b> Unchanged
dZ	<b>Method C:</b> Unchanged <b>Method D:</b> Unchanged
Shape_Width_or_dMax	<b>Method C:</b> Unchanged <b>Method D:</b> Unchanged
Shape_Options	<b>2020-10-AA now supports two additional options:</b> <ul style="list-style-type: none"> <li>Method C – invokes Method C approach for layer</li> <li>Method D – invokes Method D approach for layer</li> </ul>
L1_Oververt	<b>Method C:</b> Unchanged <b>Method D:</b> Unchanged

L1_pBlockage	<b>Method C:</b> Unchanged  <b>Method D:</b> Unchanged
L1_FLC	<b>Method C:</b> Unchanged  <b>Method D:</b> Unchanged
L2_Depth	<b>Method C:</b> Unchanged  <b>Method D:</b> Unchanged
L2_pBlockage	<b>Method C:</b> Unchanged  <b>Method D:</b> Unchanged
L2_or_L23_FLC	<b>Method C:</b> Unchanged  <b>Method D:</b> Form loss coefficient for pressure flow against the Layer 2 and 3 combined (i.e. the bridge deck and rails)
L3_Depth	<b>Method C:</b> Unchanged  <b>Method D:</b> Unchanged
L3_pBlockage	<b>Method C:</b> Unchanged  <b>Method D:</b> Unchanged
L3_FLC_or_fIP	<b>Method C:</b> Unchanged  <b>Method D:</b> Sets the inflection point (IP) at which the transition from pressure flow to drowned flow commences. The input value is dimensionless and factors the combined depths of Layers 2 and 3 (e.g. 1.2 is an inflection point 20% above Layer 2 and 3). Observations thus far indicate the inflection point is around 1.5 to 2 times the depth of the bridge deck (assuming no rails). The effect of partial blockage (e.g. 20% blockage of Layer 3 representing the bridge deck rails) is to reduce the inflection point proportionally by the blockage amount.

### 3.5 Bug Fixes and Minor Enhancements

#### 3.5.1 Bug Fix for Read GIS Z Line MIN

Build 2020-10-AA fixes a bug with Read GIS Z Line when using the MIN or GULLY options (which both do the same). E.g. Read GIS Z Line MIN == or Read GIS Z Line GULLY ==. For the 2018-03-AC to 2020-01-AB Builds this option failed to modify any z points. This issue does not affect the other Read GIS Z Line options including MAX / RIDGE, THICK or ADD. This also does not affect other topography modifiers such as Read GIS Z Shape.

**Note: Due to this bug fix it is recommended that users of the 2018-03-AC, 2020-01-AA and 2020-01-AB Builds which have models that use the Z Line Min command update to 2020-10-AA or later.**

#### 3.5.2 Support for LandXML Files with Multiple Surfaces

Build 2020-10-AA now supports LandXML TIN files with multiple surfaces. For 2020-10-AA points and triangles for all surfaces are read in together. Previously this could cause ERROR 3500 to occur if SGS was enabled.

#### 3.5.3 Bug Fix for Layered Flow Constrictions

Build 2020-10-AA fixes an issue with the processing of thin or thick lines in the layered flow constrictions layer that could result in duplicate points being created in the \_lfcsh\_uvpt\_check file. Testing has shown that this does not affect the hydraulic computations.

#### 3.5.4 Bug Fix for XF Files for TINS Created by TUFLOW

Build 2020-10-AA fixes an issue which could result in very slight differences in final elevations due to floating point precision issue when reading triangles from an XF file, compared to reading the original dataset. This could occur for region objects in the following input layers:

- Create TIN Zpts
- Read GIS Z Shape, Read GIS Z Shape and Read GIS Variable Z Shape
- Read GIS FC Shape and Read GIS Layered FC Shape

When reading XF files, prior to 2020-10-AA the order of each triangle was stored in a consistent clockwise direction. However, when initially processed (i.e. not using an XF), this was processed using either clockwise or anticlockwise triangles depending on the direction the GIS polygon was digitised. This could result in very minor differences in the elevation assigned at zpts based on floating point precision when interpolating elevations from a triangle. These differences are typically less than  $1.0 \times 10^{-4}$ , are too small to be noticeable in the check files, but occasionally could be seen in a very slight change to results. From 2020-10-AA the XF file also stores the original triangle direction and should now provide consistent results with and without XF files for both single and double precision builds.

Build 2020-10-AA will automatically recreate the XF files for the layers listed above so there is no need to remove affected XF files, but a slower start-up time will be evident for the first simulation.

Note: That due to a bug in TUFLOW releases prior to 2020-10-AA, when reading XF files created in incompatible builds, TUFLOW will flag the XF as being an incorrect version and issue WARNING 2236, however, TUFLOW will then issue an ERROR 2146 and stop.

For example, if running 2020-01-AB with XF files created with the 2020-10 release the log file will show:

Reading XF file...

NoXY: WARNING 2236 - TIN XF file is an old version or wrong format. XF file not being used.

Wiki Link: [https://wiki.tuflow.com/index.php?title=TUFLOW\\_Message\\_2236](https://wiki.tuflow.com/index.php?title=TUFLOW_Message_2236)

NoXY: ERROR 2146 - Could not allocate memory (RAM) for TIN Interpolation Work Arrays.

Wiki Link: [https://wiki.tuflow.com/index.php?title=TUFLOW\\_Message\\_2146](https://wiki.tuflow.com/index.php?title=TUFLOW_Message_2146)

Closing any unclosed GIS layers...

TUFLOW will then stop. **In order to switch back to a build prior to 2020-10-AA, the XF files will need to be turned off or deleted.**

### 3.5.5 Bug Fix for TIN XF Files that Merge Elevations

Build 2020-10-AB fixes an issue with Read GIS Z Shape, Variable Z Shape, FC Shape, Layered FC Shape and Create TIN Zpt, where the region is merging with the elevations at the perimeter of the polygon. Previously when running a model with a range of cell sizes, the merging locations / elevations would be based on the first cell size run. For 2020-10-AB, the XF file name now includes the domain number and cell size. This means that each cell size will get a unique XF file created. This issue could previously cause (typically minor) changes in the elevations if running with and without XF files enabled.

Build 2020-10-AB also fixes an issue with those commands that would read an XF file even after the original input files have been updated. Build 2020-10-AB now checks whether the following inputs have been changed before reading the XF file: (1) number of input file(s) in command; (2) all file names read by the command (instead of just the first file name prior to Build 2020-10-AB); and (3) modified dates of all input files.

### 3.5.6 Bug Fix for Geometry Control Update for HPC/Quadtree Models with Restart File

Build 2020-10-AB fixes a discrepancy of restart model geometry between the Classic and the HPC models. The Classic model reads the water level/velocity from the restart file, while the model geometry can be changed from the original simulation. On the other hand, the HPC model reads the bed elevations from the restart as well, which means the change made in .tgc files won't be applied in the restarted model. The intention of reading bed elevations from the restart file was to inherit the topographic update done by variable Z shape, as the variable Z shape is reset after restart in the

Classic solver. To fix the inconsistency, Build 2020-10-AB introduced a new command “`HPC Restart Geometry == TGC | Restart File`”. The default option is “TGC”, which uses the bed elevation created by .tgc files and this is consistent with the Classic solver. The bed elevation from the restart file can be used by selecting the option “Restart File”. This option is available for HPC including Quadtree mesh refinement as well.

A further update is underway to store the status of variable Z shapes in the restart file, in order to inherit the topography updates and allow modellers to change other .tgc commands at the same time.

### 3.5.7 Additional check for “Grid Size” commands in .tgc

Build 2020-10-AB adds an additional check for “Grid Size” command in the .tgc. Previously if “Grid Size ==” was specified, not the expected “Grid Size (N,M) == ” or “Grid Size (X,Y) == ” then the command was ignored resulting in a model domain that was not the expected size. From Build 2020-10-AB onwards an ERROR 2482 message is generated and the simulation is halted.

## 3.6 Support for 12DA SuperTINs

Build 2020-10-AB introduces support for super tins in the 12DA file format. Super TINs comprise of a list of TINs within the file.

Prior to this build if the keyword “super\_tin {” was encountered in a .12da file, then the following error was generated and the simulation stopped:

ERROR 2417 - Unable to process a 12da Super TIN file. Please use 12da single TIN files.

The new method for reading 12da TIN files may require slightly more memory (RAM) than the previous method. It is possible to revert to the older (Method A) for reading 12da files with the .tgc command:

```
12DA Read Approach == Method A | {Method B}
```



### 3.7 New SGS Approach and High-Resolution Output (2022 Beta Feature)

#### 3.7.1 Description (2022 Beta Feature)

Build 2020-10-AB introduces a new approach for processing the sub grid sampling locations. The new approach offers some considerable benefits as follows:

- Removes the issues associated with cells that have partial coverage of data layers, with the command “[SGS Partial Grid Update Null Frac ==](#)” not needed to control the treatment of partially sampled cells.
- Includes sampling of material layers to carry out a parallel channel conveyance calculation across each cell side. This produces a more accurate low flow conveyance computation where the Manning’s n varies within a 2D cell. Depth varying Manning’s n values are also supported.
- Facilitates a new high resolution map output option where TUFLOW will directly produce grid based map outputs at a fine resolution as an alternative to post-processing using utilities like asc\_to\_asc. See Section 3.7.4 for more information.

For the previous approaches, after each elevation data layer the SGS sampled elevations within a cell and along cell sides were stored as height versus volume curves for cells and height versus flow area/width curves for faces, with the sampled elevations then discarded for memory efficiency. In the new approach the SGS sampled elevations are retained until all topographic updates are complete, then followed by the storing of the curves.

The drawbacks of this new approach are:

- The SGS sample locations must be the same for all datasets, for prior SGS sampling approaches the sample distance can be changed for different data layers.
- The XF files contain raw elevation data, not the stored SGS curves. This means after the raw elevation data in XF files are read, and other geometry commands are processed, the elevation data still needs to be converted to the SGS curves at the end of the pre-processing. This data conversion step cannot be skipped by using XF files, which means the initialisation maybe slower than that for the previous approaches. We are looking to address this in a future release.
- Method C requires more memory for pre-processing than the previous approach if the same sampling distance is used.

#### 3.7.2 .tcf Commands (2022 Beta Feature)

The new approach can be invoked with the .tcf command below. Note this is a beta feature and subject to on-going successful testing the intent is to make this approach the default if SGS is enabled for the 2022 release.

[SGS Approach ==](#) Method C

When using Method C, any occurrence of “SGS Sample Distance ==” in the .tgc file is not used, with the new .tcf command “SGS Sample Frequency ==” as described further below required. The frequency sets the number of sample points across a cell face, with the square of the number being the number of SGS sample locations per cell.

Typically .tgc commands can be repeated to modify the behaviour throughout the processing. For “SGS Sample Frequency ==” this command cannot be repeated and is therefore a .tcf command, not a .tgc command.

“SGS Sample Frequency ==” expects an odd number starting at 3 (9 cell sample locations per cell) with a maximum value of 127 (16,129 cell sample locations per cell). If an even number is specified, it will be rounded up to the next odd number and the model outputs a CHECK 3533 message. The larger the sample frequency the greater the memory requirements and slower the start up, so a pragmatic approach should be taken with consideration of the resolution of the data layers and the modelling constraints/objectives. The use of an odd number ensures that the cell corners and cell side mid-points always have an SGS sample point. The .tcf command is:

```
SGS Sample Frequency == <number_of_sample_locations_per_face>
```

For example:

```
SGS Sample Frequency == 11
```

If a Quadtree mesh is being used, different sample frequencies can be defined for each nesting level by specifying multiple frequency values after the “SGS Sample Frequency ==” command. For example, the line below sets a frequency of 21 for nesting level one, 11 for nesting level two and 5 for nesting level three.

```
SGS Sample Frequency == 21, 11, 5
```

If there are less frequency values than nesting levels, the last specified frequency is used for all additional nesting levels. For example, for a three level Quadtree mesh the line below would set both nesting level two and three to have a frequency of 11.

```
SGS Sample Frequency == 21, 11
```

### 3.7.3 Sampling of Material Manning’s n Values (2022 Beta Feature)

September 2022: Due to the issues below, for Build 2020-10-AE with SGS Approach == Method C, the default is for SGS Materials to be disabled. Build 2020-10-AE has the following changes for SGS Materials:

- A bug fix for models with SGS materials enabled and when the materials file (.tmf or .csv) was not in ascending order. This could cause the incorrect Manning’s n to be used.
- Fixes an issue with the \_uvpt check file which reported the Material at the face centre which was offset by 1 SGS sample frequency. This did not affect the computation only the check file.
- If SGS Materials == On is specified in combination with depth varying Manning’s n value then ERROR 3546 is issued and the simulation stopped. If SGS Materials is enabled without depth varying Manning’s n values then WARNING 3546 is issued.

Update June 2022: An issue has been identified which is limited to models that use the beta functionality combination of SGS Approach == Method C and SGS Materials == ON, and if the materials have depth varying Manning’s n values. As described below, SGS material sampling is turned on by default with SGS Approach == Method C. This can produce unexpected .flow results

when some of the materials have depth varying Manning's n values. The extent to which the results will be impacted will vary from model to model, with most models experiencing very limited differences. The best way to check the impact for a particular model is to simply turn off the materials sampling by adding "SGS Materials == OFF" and compare to previous results.

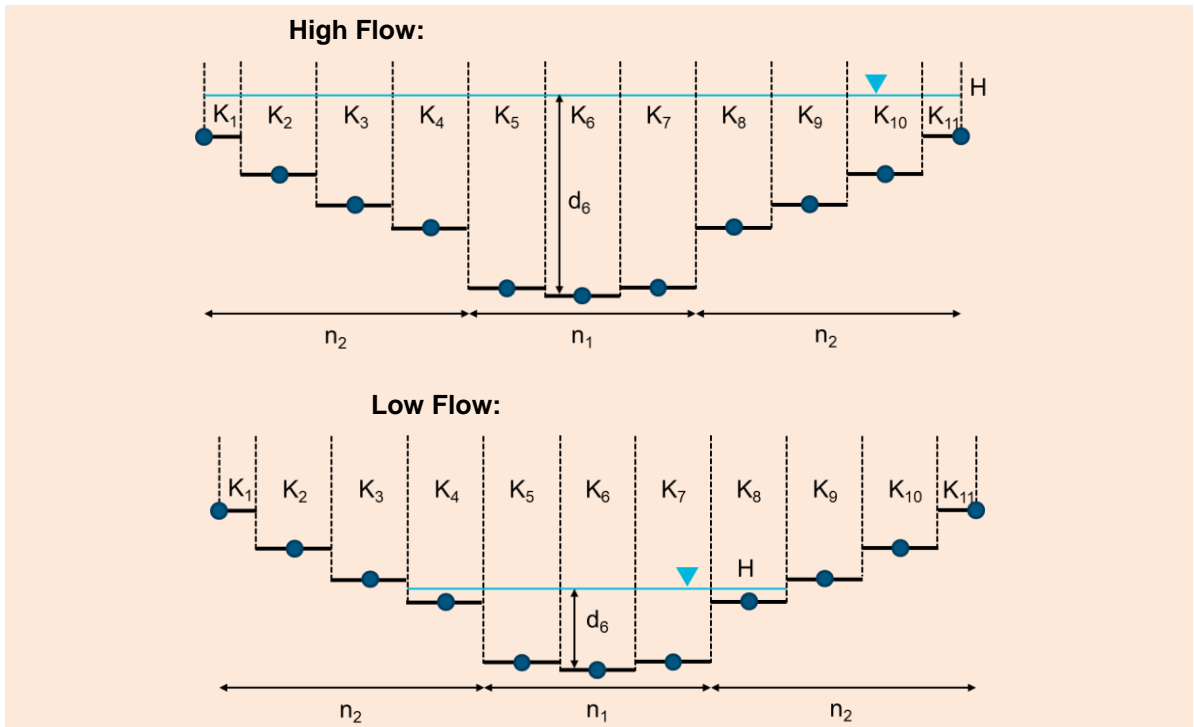
The above issue is fixed for the upcoming 2022 release (planned for second half of the year). Note that the new approach will supersede the beta functionality with no backward compatibility. In the meantime, we strongly recommend that if using 2020-10-XX releases with SGS Approach == Method C, with materials with depth varying Manning's n values, that the materials subsampling (which is on by default) is explicitly turned off with the command "SGS Materials == OFF". When developing new models with a 2020-10-XX release, and utilising "SGS Approach == Method C", and with materials using depth varying Manning's n, we recommend disabling the material sampling by adding "SGS Materials == OFF".

When using "SGS Approach == Method C", the material (land use) values can also be sampled along the cell sides or faces. This is the default if using SGS Approach == Method C, but can be turned off using the .tcf command:

```
SGS Materials == OFF | {ON}
```

The elevations and Manning's n values are sampled at the same locations along a cell face and are used to carry out a parallel channel analysis in a similar manner to that carried out for a 1D cross-section (see Section 5.10.3 in the 2018 TUFLOW manual). Any variations in Manning's n with depth are also accounted for with the depth based on the sampled elevation, not the lowest elevation as is used by the other SGS methods.

The parallel channel analysis carried out produces a curve of conveyance versus height, thereby producing a more accurate representation of the flow paths through a 2D cell. The figure and equations below are used to perform the analysis.



#### SGS Material Sampling at Cell Faces for SGS Approach == Method C

The conveyance across the cell face at given water level is:

$$K(H) = \sum K_i = \sum \frac{1}{n_i} A_i d_i^{\frac{2}{3}}$$

where  $A_i$  is the flow area and  $d_i$  is the flow depth of each sampling segment.

The equivalent Manning's  $n$  at a given water level is:

$$n(H) = \frac{1}{K(H)} A_{total} d_{face\_average}^{\frac{2}{3}}$$

where  $d_{face\_average}$  is the total flow area ( $A_{total}$ ) over the wetted width.

For the radius formulation, the resistance radius is used. Support for hydraulic radius using "SGS Radius Approach == (see Section 3.2.3) is not yet supported.

**Note:** The SGS material sampling only applies to the cell faces, i.e. for the hydraulic bed roughness calculations. For cell properties based on other material parameters such as rainfall losses and fractions impervious, these remain based on the Material ID sampled at the cell centre.

#### 3.7.4 High Resolution (HR) Grid/Raster Map Outputs (2022 Beta Feature)

When using "SGS Approach == Method C" the sampled elevations are retained including topography modifiers such as breaklines. At the end of the geometry processing this allows a high-resolution raster grid to be written and used for high resolution map outputs. Currently, only FLT and ASC raster formats are supported, with support for other grid formats planned.

To produce high-resolution output, add “HRASC” or “HRFLT” to the “Map Output Format ==” command, for example:

```
Map Output Format == XMDF FLT HRFLT
```

The map output data types and output interval can be defined separately for “HRASC” and “HRFLT” formats. For example:

```
HRFLT Map Output Data Types == h d ZPA
```

```
HRFLT Map Output Interval == 0
```

The output resolution for high resolution grid output is defined using a new .tcf command as follows:

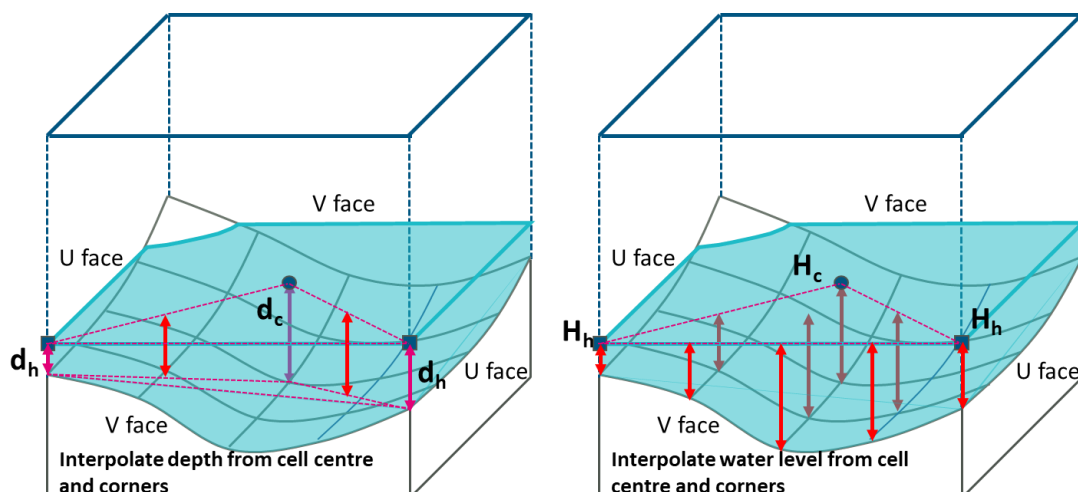
```
HR Grid Output Cell Size == 0.5
```

If this command is not defined, the default “HR Grid Output Cell Size ==” is set as the SGS sampling distance based on the “SGS Sample Frequency ==” command for HPC model. For HPC model with Quadtree refinement, the smallest SGS sampling distance is used.

**Note:** The high resolution output uses a similar approach as the remapping tool in asc\_to\_asc utility (see Section 7.6 or [https://wiki.tuflow.com/index.php?title=TUFLOW\\_Remapping](https://wiki.tuflow.com/index.php?title=TUFLOW_Remapping)). The approach is:

- The water level at each HR raster output cell is interpolated from the computed 2D water levels.
- The depth is the difference between the interpolated water level and the terrain elevation interpolated from the surrounding SGS sampled elevations.
- For any output types other than water level and depth, this function does NOT interpolate the results to a finer resolution, but only extends/reduces the output extent to the dry/wet extent. Therefore, the detail of the high-resolution hazard remains the same as the original output grid. To improve the resolution of velocity based outputs such as hazard a finer 2D cell size is needed.

The regular grid output interpolates depth from cell centres/corners with a default output resolution of half the cell size. The difference between the regular and the HR grid output is illustrated below.



**Regular Grid Depth Output (Left) and HR Grid Depth Output (Right)**

When modelling breaklines in TUFLOW, “thin” breaklines modify the cell face elevations but do not modify the cell storages. When outputting the high-resolution outputs, the user can set whether the cell face elevations are including with the .tcf command (the default is ON, to use face elevations):

```
HR Grid Output Use Face Elevations == {ON} | OFF
```

High-resolution grid outputs are also supported by HPC and Quadtree Output Zones. For example:

```
Model Output Zones == ZoneA
```

```
Define Output Zone == ZoneA
```

```
    Read GIS Output Zone == ..\model\gis\2d_oz_S05_002_R.shp
```

```
    Write Check Files INCLUDE == dem_Z
```

```
    Map Output Format == HRFLT
```

```
    HRFLT Map Output Data Types == h d ZPA
```

```
    HRFLT Map Output Interval == 0
```

```
    HR Grid Output Cell Size == 0.5
```

```
End Define
```

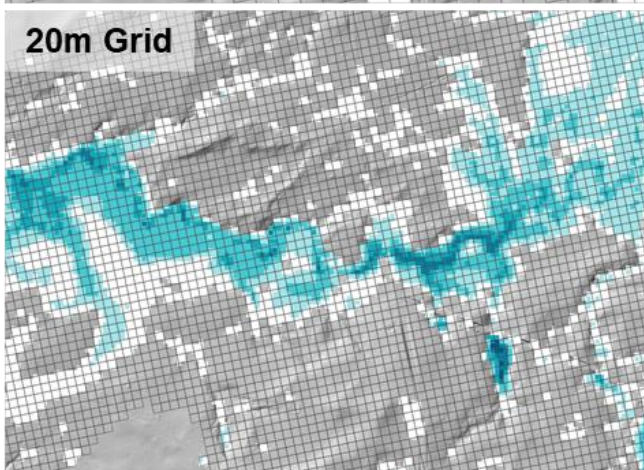
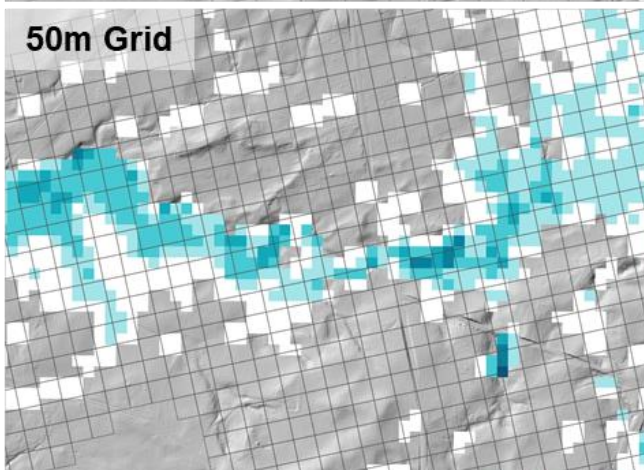
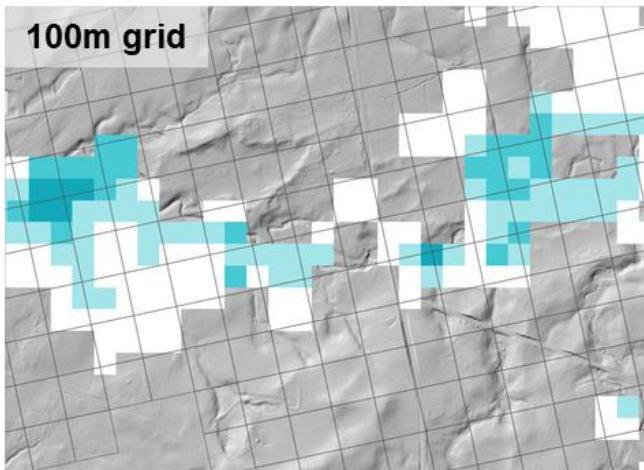
If either HRASC or HRFLT are defined in the Map Output Format, then a high-resolution DEM\_Z check file (DEM\_Z\_HR) is produced instead of the DEM\_Zmin check file for the whole model or output zone that high resolution output has been specified for.

The following figures show images from a direct rainfall SGS model demonstrating the different effects and options for mapping depths. The first figure shows the benefits of high-resolution mapping (right side images), especially where large cell sizes are used, compared to the mapping taking the exact ground elevation at the cell centre (left side images).

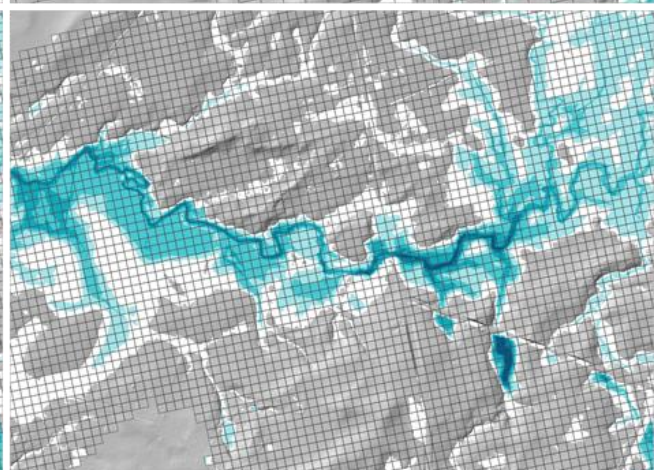
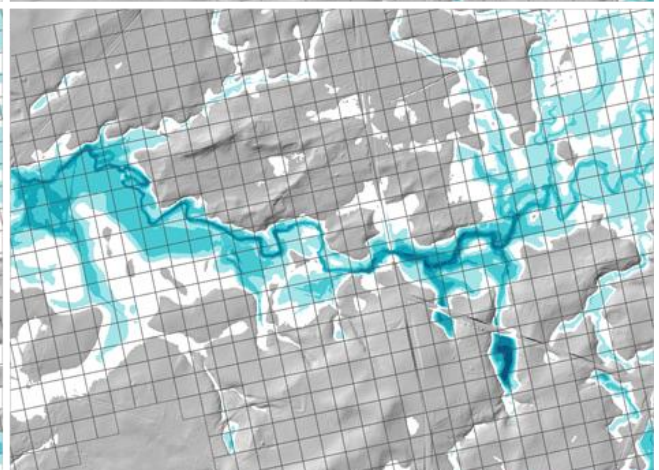
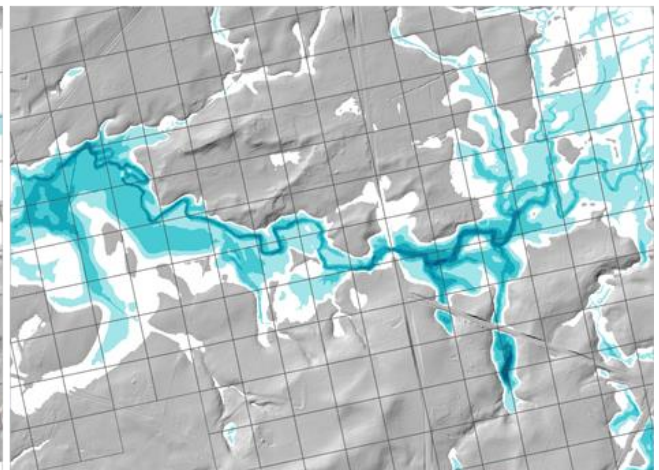
The second figure shows for the 20 m case the effect of different mapping options. A yellow shade is used to map shallow depths less (in this case 5 cm), which are often excluded using features such as “Map Cutoff Depth ==”. The top left image is for a no SGS simulation which shows significant areas being flooded to depths greater than 5 cm, when this is not the case highlighting the benefits of using SGS where cell sizes are much larger than the DEM resolution and direct rainfall is being applied. The top right image shows the exact elevation (at the cell centre) option, the bottom left the new cell averaged option and the bottom right the high-resolution option. The exact option will show disconnected flooded areas due to where the cell centre elevation sits above the partially wet cell water level. The depth averaged shows a better continuity between low flow paths but can “smooth” the depth mapping due to the averaging. The high-resolution option shows the best and clearest depth mapping, with the much larger file sizes being the primary constraint.



## FLT Output

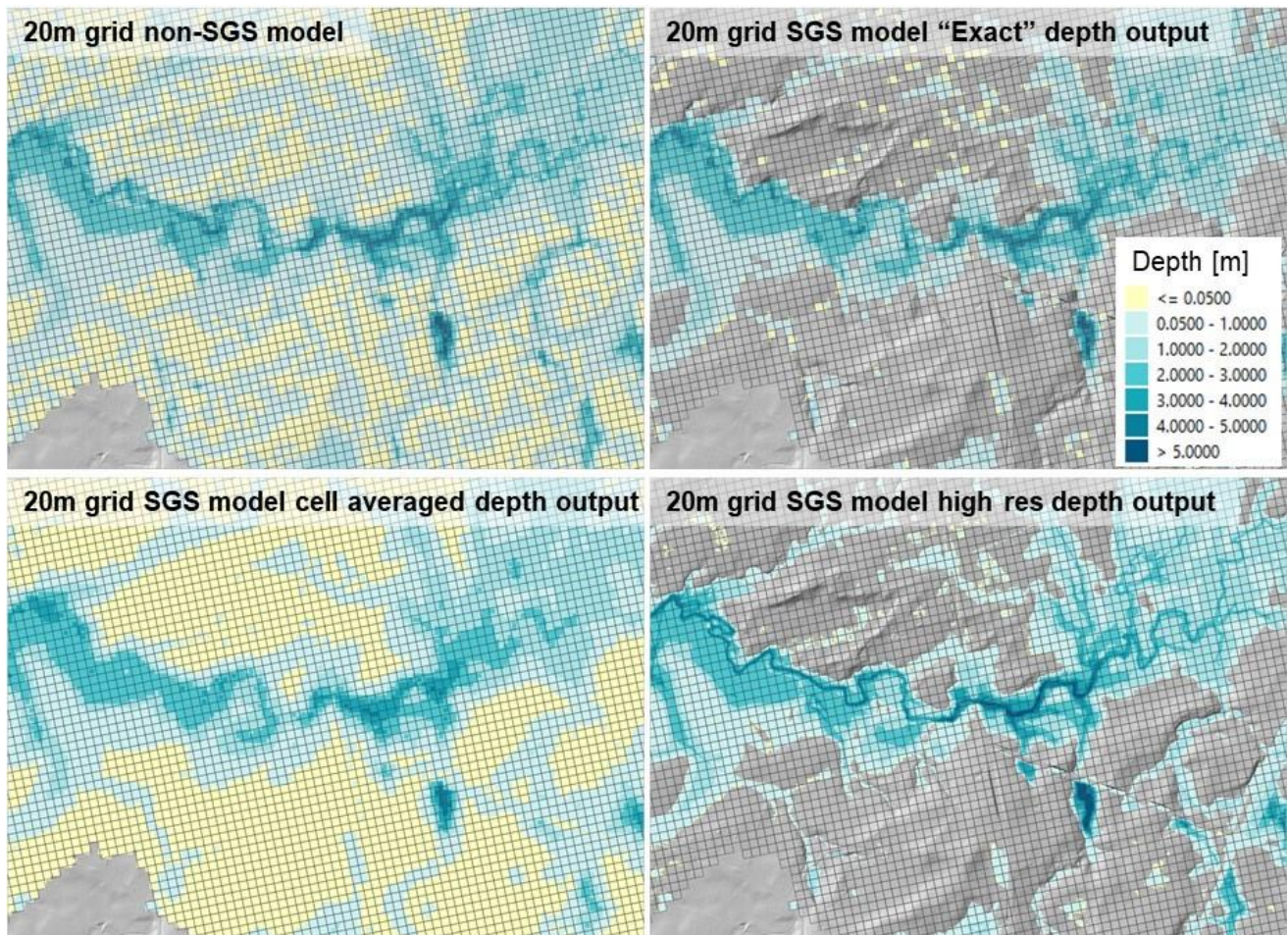


## HRFLT Output



Example of High-Resolution Depth Output for Direct Rainfall Model with Different Grid Size

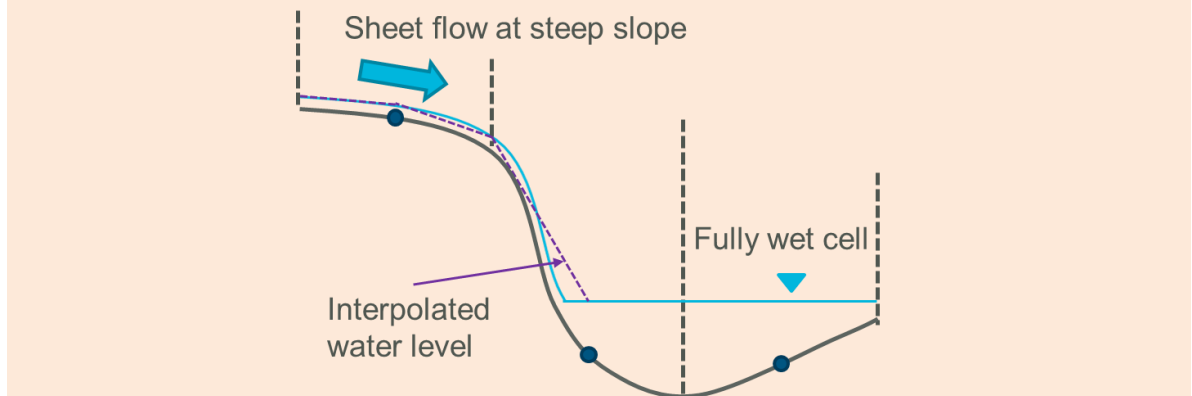




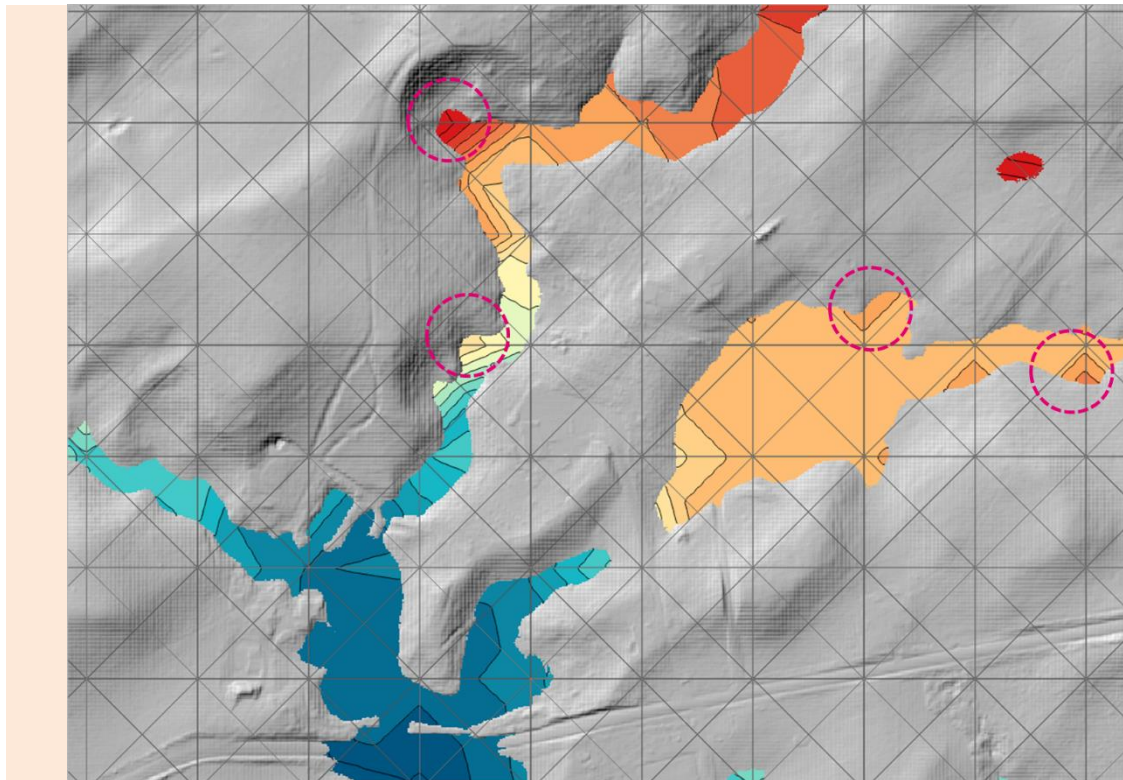
**Example of Different Depth Mapping Options when using SGS with Direct Rainfall**  
 (Yellow shade is for depths less than 5 cm)

### 3.7.5 High Resolution Output Water Level Interpolation (2022 Beta Feature)

The existing water level corner interpolation methods (i.e. the “[Map Output Corner Interpolation](#)” command in the TUFLOW manual) apply a minimum corner water level of “corner elevation + cell wet/dry depth” if all the surrounding 4 cells are wet. However, this may produce “bumpy” water level outputs along narrow streams in a direct rainfall SGS model between sheet flow cells and fully wet cells when using the new high resolution output. This is caused by applying a linear water level interpolation between SGS cells that have a non-linear minimum SGS elevation as illustrated below.



#### Linear Corner Interpolation and SGS



Example of “bumpy” or elevated water levels along narrow streams

HR Interpolation Approach == Method A

Two additional methods have been added for the corner water level interpolation for high resolution (HR) output:

`HR Interpolation Approach == {Method A} | Method B | Method C`

- Method A is the default option that applies the same water level interpolation method used for non-SGS map outputs.
- Method B performs sheet flow checks at cell faces and ignores the water level from the upstream cell.
- Method C applies the same sheet flow checks as the Method B. In addition, it also uses the number of wet SGS sampled points as a weighting that biases non-sheet flow cells that further improves the mapping to in-stream water levels.

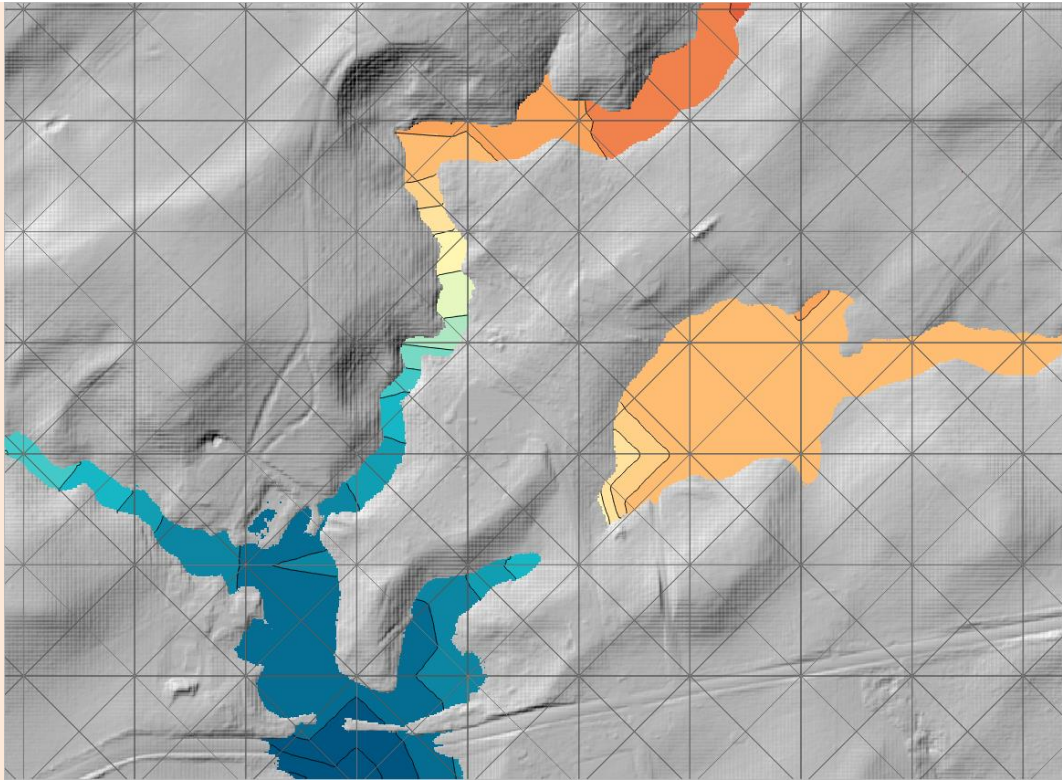
The two images below present the high-resolution water level output at the same location, but with `HR Interpolation Approach == Method B` and `Method C`. As can be seen, the water level along the narrow stream is “smoother”. Note that when these two methods are applied, the interpolated corner water level is biased to the non-sheet flow cells, and consequently, sheet flow cells may appear as “dry” cells.

**Note: Whilst Methods B and C can substantially improve the water surface mapping of SGS models using direct rainfall (rain-on-grid), there will always be inaccuracies with mapping at a higher resolution than the 2D cell resolution due to interpolation and extrapolation. Regardless of the software, the greater the ratio of 2D cell size to the high-resolution DEM cell size, the greater the potential for mapping inaccuracies. Should better mapping accuracy be required, reducing the 2D cell size to compute the spatial variation in water surface and velocities more accurately is, by far, the best course of action.**

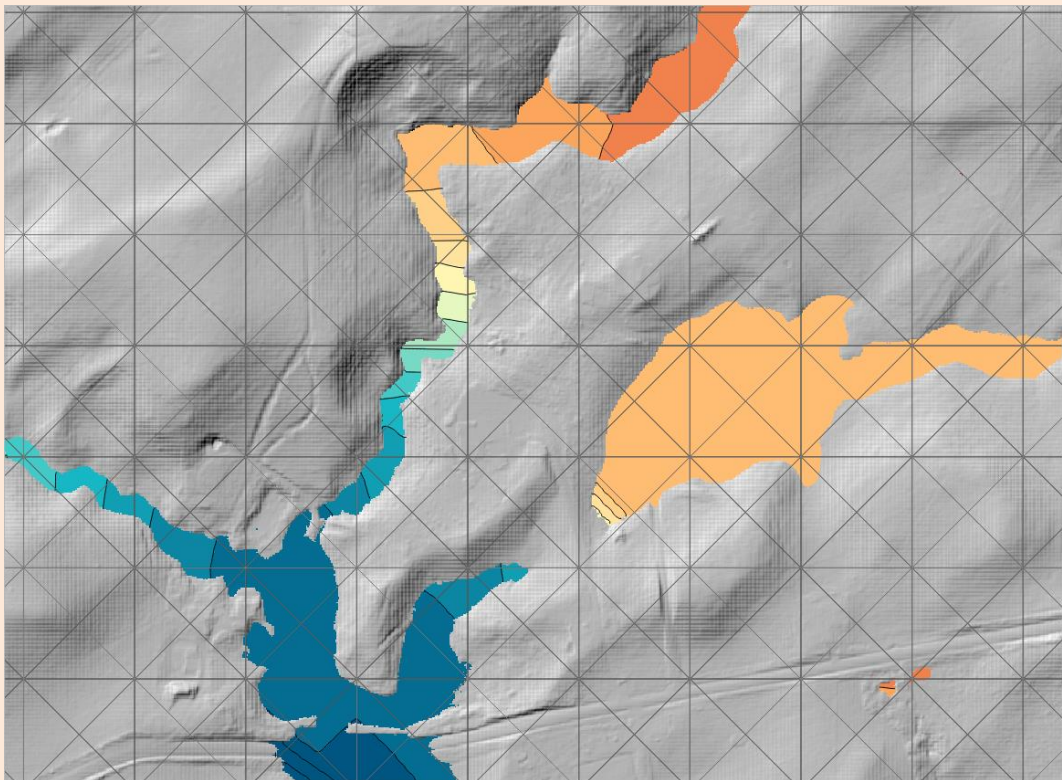
For the 2022 release, other methods are under-development to resolve this limitation. Please also note that it is not necessary to use these options for non-rainfall on grid models.

This 2022 beta feature will continue to be tested and potentially enhanced, so please send feedback to [support@tuflow.com](mailto:support@tuflow.com).





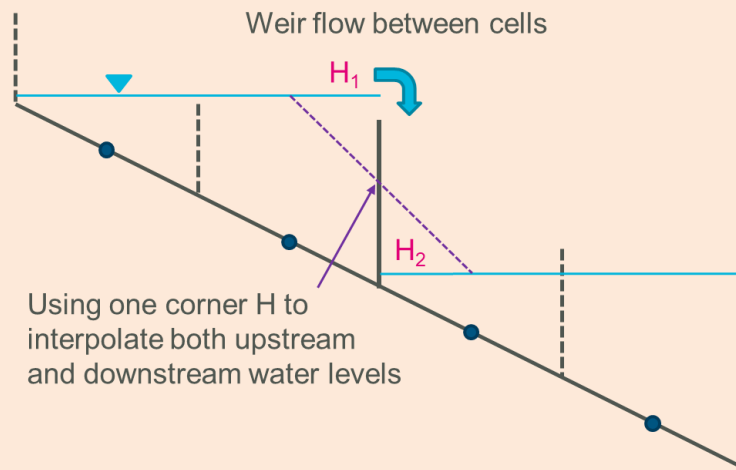
**HR Interpolation Approach == Method B**



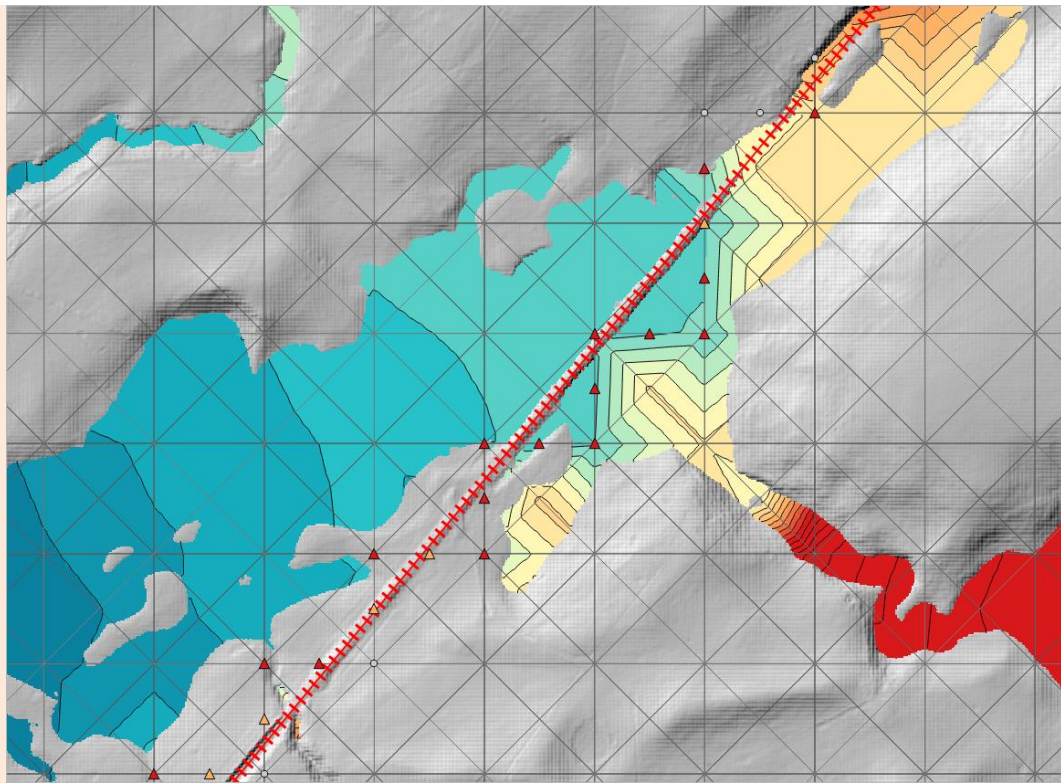
**HR Interpolation Approach == Method C**

### 3.7.6 Treatment of Thin Breakline in High Resolution Output (2022 Beta Feature)

Thin breaklines are often used to define hydraulic controls, such as levees and road embankments. These controls often experience upstream controlled weir flow, where the water levels on both sides of the breakline can be considerably different. Retaining just one water level (upstream, downstream or average) at output points along the breakline may result in unsatisfactory water level map output as illustrated in the two figures below. This issue becomes more profound the larger the 2D cell size.



**Corner Interpolation along Thin Breaklines**



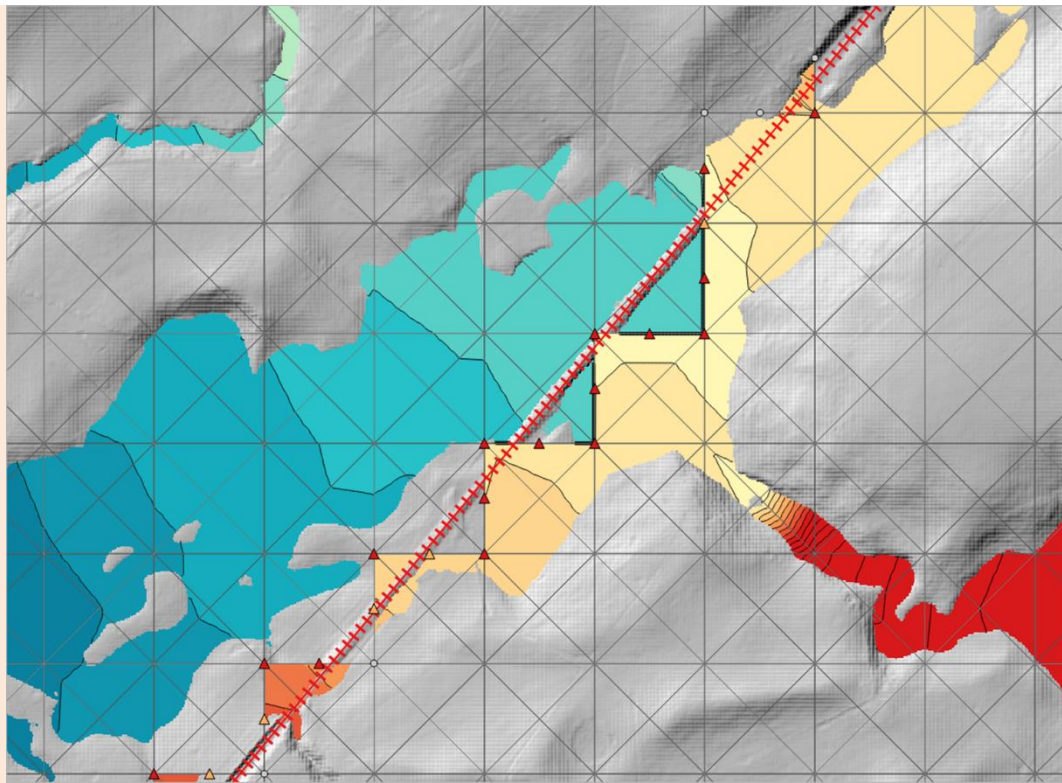
**Example of “smoothed” water level output along thin breakline**  
**HR Thin Z Line Output Adjustment == OFF**

Build 2020-10-AB introduces a new feature to treat water level along thin breakline vertexes. Note, this feature (at present) only works for thin Z lines in Z Shape (2d\_zsh) layers.

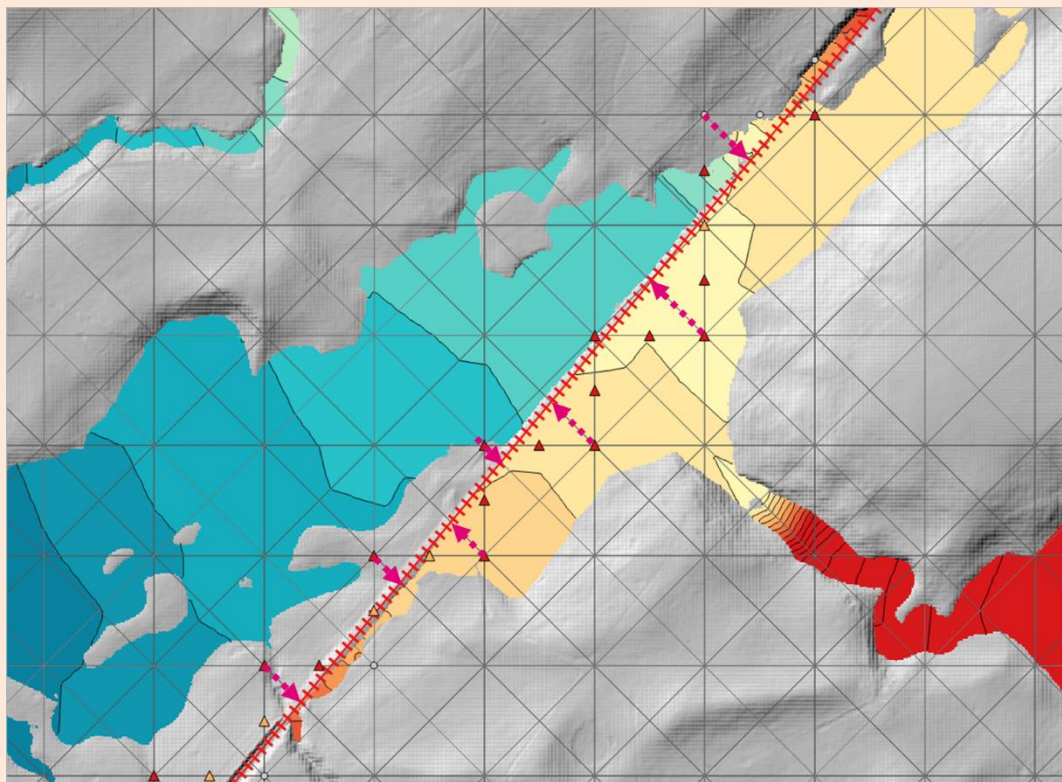
HR Thin Z Line Output Adjustment == {OFF} | ON CELL SIDES | ON ALIGNMENT

- “OFF”: the default, does not apply any water level adjustment.
- “ON CELL SIDES”: two water levels, rather than one, are stored at cell corners selected by thin breaklines. The two water levels are used to interpolate high-resolution output water levels on either side of thin breaklines. This option produces a sudden drop in water level across the breakline as illustrated in the figure below, noting the sudden drop follows the edge of the breakline along the 2D cell sides where the breakline is applied. This option is a good one for modellers to use so they can see where the drop in water levels is being computationally applied.
- “ON ALIGNMENT”: As an extension to the “ON CELL SIDES” option, this method moves and aligns the 2D cell corner vertices to the original alignment of the breakline for producing high-resolution map output. The arrows in the figure below show how vertices of the TIN used for map output are moved to align with the breakline. This can produce substantially improved high-resolution mapping along thin breaklines with substantial drops in water levels, especially for larger 2D cell sizes. Note that the cell corners are NOT shifted for the hydraulic calculations.





**HR Thin Z Line Output Adjustment == ON CELL SIDES**



**HR Thin Z Line Output Adjustment == ON ALIGNMENT**  
(Arrows show vertices that are moved in the triangulation used for HR mapping)



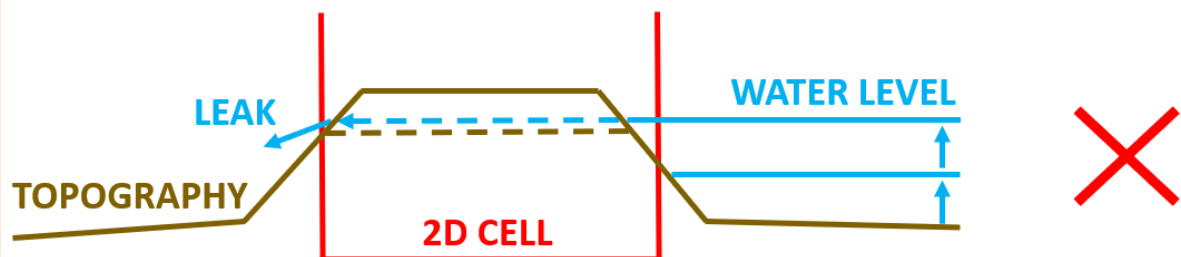
### 3.7.7 Breakline Detection (2022 Beta Feature)

When the SGS elevations are being processed, every cell face will store a curve of elevation versus flow width / flow area. Likewise, each cell has a relationship of elevation versus cell volume. To make this memory efficient TUFLOW does not store the locations in the cell or along the cell face where the low / ridge elevations occur.

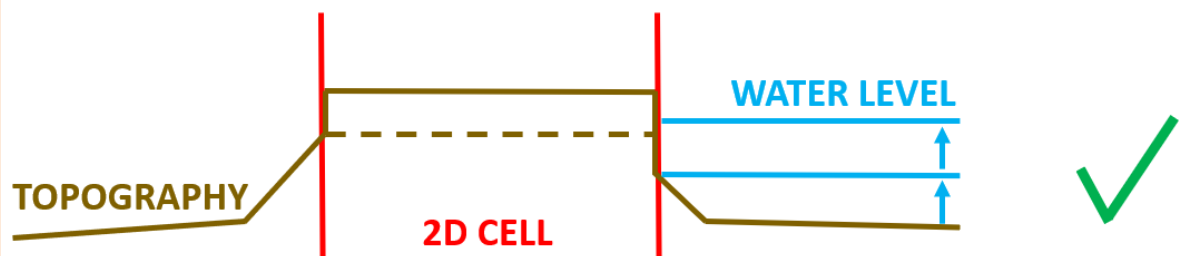
Without breaklines, SGS will be more likely to allow "leaks" through a levee or embankment compared with running the model without SGS. For example, if a 2D cell straddles a levee, SGS will sample the low elevations on either side of the levee and will flow at a lower level than if SGS was not applied and the 2D cell centre was located on top of the levee causing the 2D cell to block flow until the levee is overtopped. Therefore, whilst the need to have breaklines is paramount, regardless of whether SGS is used, it is even more important for SGS models to accurately represent hydraulic controls such as levees (artificial or natural), road/rail embankments, fences, etc.

In the figure below, the top image shows, with SGS applied and no breakline specified, how flow across the cell on the embankment occurs when the water level reaches the dashed brown line, which is below the crest. The bottom image illustrates that with a breakline specified the 2D cell's SGS curves are overwritten with the interpolation to the 2d\_zsh breakline setting the cell and cell faces to the crest level. Note that as per usual, the breakline would need to occur after the DEM or TIN in the .tgc file.

#### SGS WITHOUT 2D Z SHAPE RIDGE



#### SGS WITH 2D Z SHAPE RIDGE



SGS and Representation of Embankments

For “SGS Approach == Method C”, the SGS sampled elevations are retained during the full initialisation stage. This now allows for a check to be performed on the potential need for a breakline by comparing the final SGS sampled elevations within a cell. Build 2020-10-AB introduces the ability to perform a check on whether a 2D cell may be allowing water to cross it when a ridge exists inside the cell.

The SGS breakline detection check processes each 2D cell to identify the minimum water surface elevation needed for a continuous wet connection through the cell traversing between left to right and between top to bottom. The maximum invert elevation of the left and right faces is subtracted from the left-right minimum water level and reports it as “uDelta”. Similarly, the top to bottom value is reported as “vDelta”. uDelta and vDelta represent the depth of water over the cell face inverts by which a natural ridge (break) line would block any flow. If either uDelta or vDelta exceeds a user specified value, a spatial (GIS) CHECK 3542 message is issued at the cell location.

To activate the breakline detection check and set a minimum value for reporting the CHECK 3542 message, specify the following .tcf command.

```
SGS Breakline Detection Delta == <maximum delta value>
```

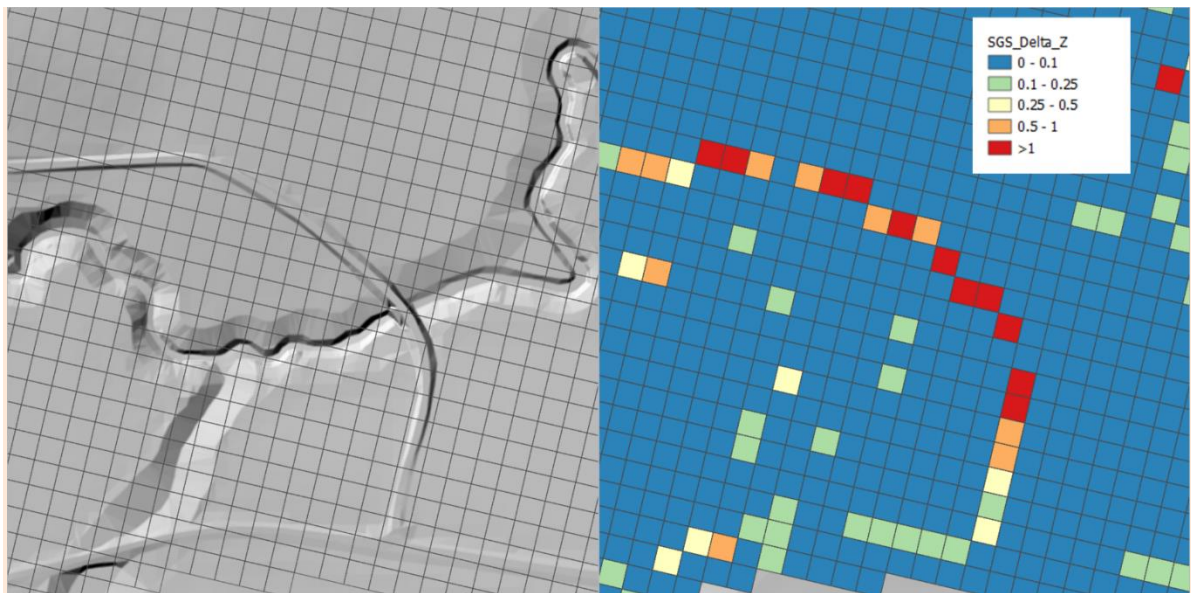
The maximum of uDelta and vDelta is output to the \_grd\_check file in the SGS\_Delta\_Z attribute.

If “SGS Breakline Detection Delta ==” is not specified, the automatic breakline detection is not performed.

**Note:** “SGS Approach == Method C” must be used to enable this feature.

**Note:** This functionality can add significant time to the initialisation process depending on the SGS sampling resolution, therefore, it is recommended to perform this check during setting up a model and to turn the feature off once the need for and creation of breaklines has been completed.

An example of using the check is illustrated in the figure below. The DEM on the left shows a road embankment and a thematic map of the \_grd\_check SGS\_Delta\_Z attribute is shown on the right. The need for a breakline is strongly evident along the top of the road as highlighted by the orange and red 2D cells.



**Example of Automatic SGS Breakline Detection Feature**

## 4 1D Solver (ESTRY)

There were no changes in Build 2020-01-AA to the 1D (ESTRY) solver.

### 4.1 Operational Spillway (SPO) Channel Bug Fixes

Build 2020-01-AB fixes a couple of bugs related to operational spillway (SPO) channels as follows and Build 2020-10-AB fixes a bug in 2020-01-AB as explained further below. If using SPO channels, it is essential to use Build 2020-10-AB or later.

- Previously the formula applied using the UASCE Hydraulic Design Criteria Sheet 312 was applied regardless of the gate setting (refer to Section 5.9.2.5 of the 2018-03 TUFLOW Manual). This formula is not applicable if the gate does not impede the flow over the spillway. Build 2020-01-AB introduces a new feature to allow for to transition between no gate-controlled flow and gate-controlled flow.
- The change made for the 2016-03 release aligning  $C_d$  values for spillways and weirs as per the note below from the manual introduced a bug in the gated flow calculations (refer to Section 5.9.2.5 of the TUFLOW Manual). This has been fixed in 2020-01-AB.

**NOTE:**  $C_d$  prior to the 2016-03 release was based on  $Q = C_d WH \sqrt{2gH}$  (as per Sheet 312), with a default  $C_d$  value of 0.5. As of the 2016-03 release, SPO channels now use the same formula as SP and weir channels and use a default value of 0.75.

**Note:** when assigning  $C_d$  and  $C_G$  values, they are to be based on the discharge formulae below which differs from the USACE HDC Sheet 312 formulae with the  $2/3$  added at the front of the formulae to be consistent with other similar structures in TUFLOW. For the formulae below refer to Section 5.9.2.5 of the TUFLOW Manual for nomenclature.

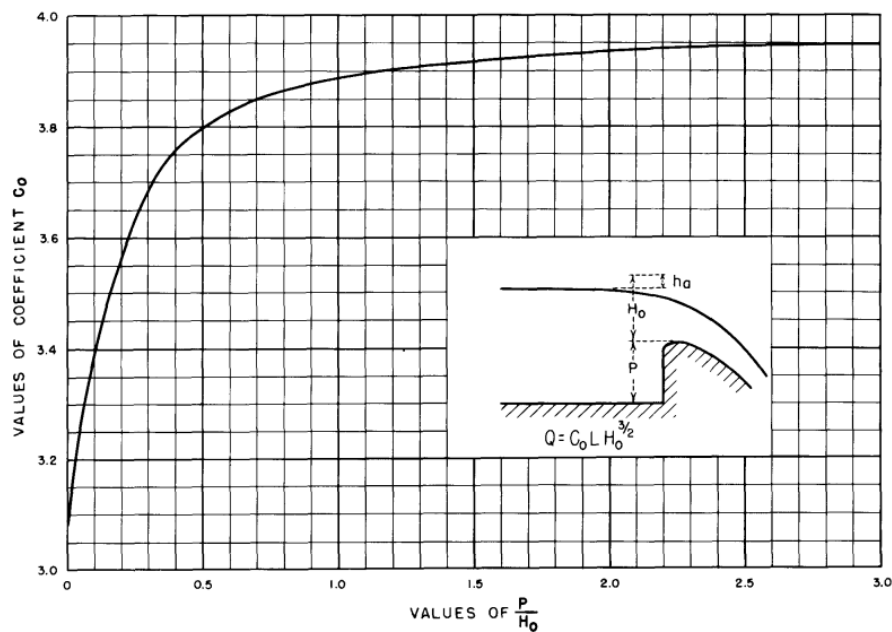
$$Q = \frac{2}{3} C_d WH \sqrt{2gH}$$

$$Q_G = \frac{2}{3} C_G WH \sqrt{2g} (H_2^{1.5} - H_1^{1.5})$$

Build 2020-10-AB fixes an issue in 2020-01-AB that “ $2/3$ ” was not correctly applied to the gate-controlled flow equation.

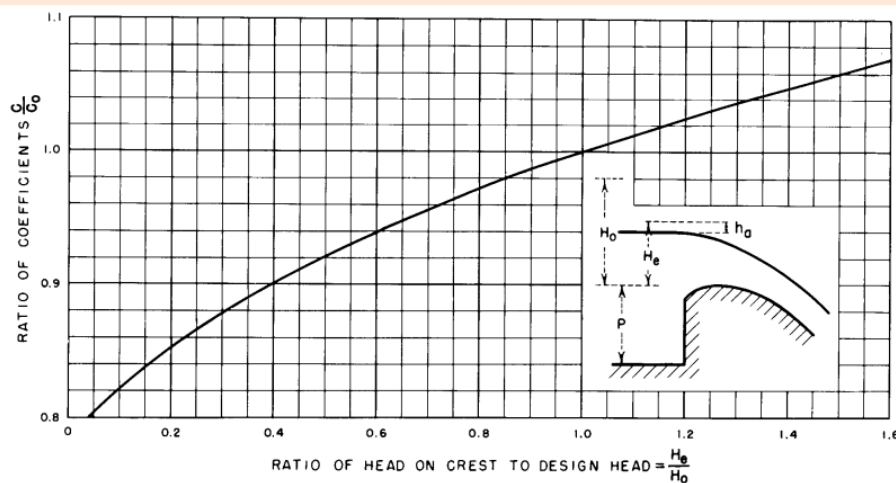
### 4.2 Change to Ogee Crest Weir (WO) Discharge Coefficient Calculation

Build 2020-10-AB has updated the discharge coefficient calculation method for the Ogee crest (WO) weir based on “Design of Small Dams” USBR (1987). “Designed Head” ( $H_0$ ) can now be specified by 1d\_nwk “HConF\_or\_WC” attribute. The ogee crest coefficient  $C_0$  (i.e. the discharge coefficient when the actual upstream head = design head) is set at the beginning of the simulation based on  $H_0$  and the height of the weir above its sill ( $P$ ) as per below Figure 9-23, USBR (1987).



**Design of Small Dams, Figure 9-23, USBR (1987)**

When the actual head ( $H_e$ ) is different from the design head ( $H_0$ ), the discharge coefficient differs from that shown on the figure above using Figure 9-24, USBR (1987). At each simulation timestep, the discharge coefficient is now adjusted based on  $H_e/H_0$  as per the chart below. The final discharge coefficient applied is  $C_0 \times C/C_0$ .



**Design of Small Dams, Figure 9-24, USBR (1987)**

Note that the relationship above is for US Customary Units, which is converted to metric if running the simulation in metric units.

The treatment of the 1d\_nwk HConF\_or\_WC attribute value is now as follows.

- If 1d\_nwk HConF\_or\_WC value is greater than zero (0), apply a constant discharge coefficient, i.e.  $C = C_0 = \text{HConF\_or\_WC}$ .

- If HConF\_or\_WC is zero (0), the actual head  $H_e$  is used to estimate  $C$  from Figure 9-23, USBR (1987), with no further adjustment based on Figure 9-24 applied. This approach should be used to estimate the discharge coefficient when  $H_0$  is not known.
- If a negative value is specified for HConF\_or\_WC, this absolute of this value will be treated as the design head  $H_0$  and the new method stated above will be applied to estimate the final discharge coefficient.

### 4.3 Enhancement to QO Channel Depth/Head Discharge Calculation

Build 2020-10-AB introduces a new .toc command for QO channels to specify how the depth/head discharge curves are calculated:

`Discharge Curve Type == {Upstream Depth} | Head Difference | Pump`

- `Upstream Depth`: (default) Uses the upstream depth above the channel invert for the depth discharge curve regardless of the downstream water level. This is the only method available for QO channels prior to Build 2020-10-AB.
- `Head Difference`: Uses the difference in upstream and downstream water levels to calculate flow rates from head discharge curves. This option is more appropriate for hydraulic structures that experience downstream controlled flow regimes.
- `Pump`: Similar to a 1D pump channel, this option uses the pumping head, i.e., downstream water level minus upstream water level to calculate flow rates from head discharge curves. The flow is always in the direction the channel is digitised. Typically, the curve has decreasing flows with increasing pumping head.

## 4.4 Minor Changes and Bug fixes

### 4.4.1 Bug Fix for Incomplete Cross-Section Data

Build 2020-10-AB fixes an issue when no data was read in from a 1D cross-section (due to an incorrect input .csv file format), if xf files were enabled (default), instead of a meaningful message TUFLOW would output:

Should not be here [jxf] - please contact [support@tuflow.com](mailto:support@tuflow.com)

and terminate the simulation. For Build 2020-10-AB, ERROR 1425 is now issued to alert the user to the problematic input file.

ERROR 1425 - End of tabular data found with no points read. Check that file is in correct format.

File = <Filename and path of .csv file>

### 4.4.2 Removes check for zero Manning's n on DF and PF channels

Build 2020-10-AE removes a check for zero Manning's  $n$  values for piping failure (PF) and dam failure (PF) 1d channels. Previously if a zero Manning's  $n$  value was input in the 1d\_nwk layer ERROR 1261 would occur even though Manning's  $n$  is not required for these channel types.

## 5 2D Solvers (Classic/HPC)

### 5.1 Overview

There are no changes in Build 2020-01-AA to the core 2D Classic solver, with the new features described in the sections below applying to the HPC 2D solver.

### 5.2 Mesh Size Insensitive Turbulence Model (HPC Only)

**Note:** This feature presently applies to TUFLOW HPC only. For TUFLOW Classic, the default approach remains the Smagorinsky method with the coefficients unchanged from prior TUFLOW releases, however, the new turbulence model discussed below may be built into the TUFLOW Classic 2D solver in a future release/update.

#### 5.2.1 Discussion on Why a New Turbulence Approach is Needed

The representation of sub-grid-scale turbulence (often referred to as eddy viscosity) has been an increasingly concerning issue as 2D cells or elements have become smaller and smaller. It is easy to demonstrate that as element size reduces, the traditional and commonly used Smagorinsky approach becomes invalid as it tends to a zero-turbulence state. The Smagorinsky approach, intended for large eddy simulation scales in coastal models, fails because it is proportional to element surface area and therefore tends to zero as the element size reduces.

The deficiencies of using Smagorinsky, especially once the cell size is smaller than the depth, has historically been accounted for in TUFLOW by using an additional constant component, the default setting in TUFLOW for many years has been to calculate the turbulence component as the addition of a Smagorinsky and a Constant eddy viscosity (rather than one or the other) as the constant component would ensure some turbulence was accounted for as cell sizes become very small. However, our research and benchmarking over the last couple of years has shown that the constant coefficient value is highly dependent on model scale, varying by several orders of magnitude from flume scale to large river scale. This constraint makes it very hard to have a default value for the constant component as the value will be cell size dependent.

As 2D solvers of any persuasion are increasingly being asked to model at smaller and smaller element sizes, it has been increasingly important to have a cell size independent approach to sub-grid turbulence averaged in the vertical for 2D schemes. This issue is even more important, if not paramount, for models that use a mesh with varying cell sizes (i.e. flexible mesh and quadtree).

Dr Greg Collecutt and Dr Shuang Gao from the TUFLOW Team have been researching and testing alternative turbulence models during 2019 and have successfully arrived at a solution. This work has been submitted as a paper for the IAHR 10<sup>th</sup> Conference on Fluvial Hydraulics (River Flow 2020) in Delft and is the default setting for the 2020-01 HPC 2D solver. TUFLOW modellers can now vary cell size downwards or across a mesh using quadtree without seeing significant changes in results due to limitations associated with turbulence scheme assumptions, especially where the flows are complex, and cell sizes are less than flow depths. Importantly, modellers can now confidently model at all scales from sub centimetre cells for a flume to tens of metres for a large river using the same turbulence parameters – experience and benchmarking for the River Flow 2020 paper has shown this to be a non-option if using Constant and/or Smagorinsky, for which calibration of the parameters for



different cell sizes is required. It is not an understatement that this research is a game-changer for 2D solution schemes and is essential as cell sizes become smaller and smaller.

Based on the success of the new turbulence scheme for HPC 2020-01, we are planning to incorporate it into the TUFLOW Classic and TUFLOW FV 2D engines (this issue does not affect TUFLOW FV in 3D mode as it uses a full 3D turbulence model).

### 5.2.2 Turbulence (Eddy Viscosity) Formulation

The TUFLOW 2020-01 HPC solver defaults to a new eddy viscosity (turbulence) model that combines both 2D and 3D turbulence effects. The model is a slightly adapted version of that described by Wu et. al. 2005<sup>1</sup>. Like the Smagorinsky eddy viscosity model, it is a zero-equation model whereby the eddy viscosity coefficient can be diagnostically computed from the mean depth and velocity fields. However, unlike the Smagorinsky model, where the turbulent length scale is related to cell size, the length scales used in the Wu model are related to water depth, and hence the computed eddy viscosity **is not related to or dependent on cell size**. This has been shown to significantly improve the cell-size convergence of model results compared to the Smagorinsky model, i.e. the results are not directly dependent on the cell size (provided there are enough cells across the waterway to adequately define the flow).

The computed eddy viscosity is the Pythagorean sum of 3D and 2D contributions:

$$v_T = \sqrt{v_{3D}^2 + v_{2D}^2}$$

The 3D contribution is derived from a dimensionless coefficient,  $C_{3D}$ , times the product of friction velocity,  $U^*$ , and a length scale,  $L_m$ :

$$v_{3D} = C_{3D} U^* L_m$$

where friction velocity is derived from depth averaged velocity, Manning's bed friction coefficient ( $n$ ), gravity ( $g$ ), and water depth ( $h$ ):

$$U^* = |U| n \frac{\sqrt{g}}{h^{1/6}}$$

The 2D contribution is derived from a dimensionless coefficient, times the product of the square of the length scale and the magnitude of the 2D velocity gradient tensor:

$$v_{2D} = C_{2D} L_m^2 |\nabla U|$$

where

$$|\nabla U| = \sqrt{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2}$$

---

<sup>1</sup> A depth-averaged two-dimensional model for flow, sediment transport, and bed topography in curved channels with riparian vegetation, Weiming Wu, F. Douglas Shields Jr., Sean J. Bennett, and Sam S. Y. Wang, WATER RESOURCES RESEARCH, VOL. 41

For both 3D and 2D components, length scale,  $L_m$ , is set as the lower of either water depth or distance to the dry boundary.

With the Wu eddy viscosity formulation, the two user definable viscosity coefficients map to  $C_{3D}$  and  $C_{2D}$  respectively.

```
Viscosity Coefficient == <C3D, C2D> ! default values are 7.0, 0.0
```

In our testing to date, we have found  $C_{3D} = 7$  and  $C_{2D} = 0$  yields results that agree well with benchmark tests and are not significantly dissimilar from those of the previous Smagorinsky method with its default coefficients, especially where the depth is not significantly greater than the cell-size. The values of  $C_{3D} = 7$  and  $C_{2D} = 0$  are the default values applied and can be changed using the command above. This effectively ignores the 2D component in the Wu model. Alternatively, to use only the 2D component of the model (and ignore the 3D component), we have found  $C_{3D} = 0$  and  $C_{2D} = 4$  to be a suitable starting point. As always, calibration remains an essential step, however, based on the testing and benchmarking thus far values significantly different to these values, provided conventional Manning's  $n$  values are used for bed friction and any hydraulic structures are appropriately represented, are likely to indicate other errors (eg. boundary values or schematisation, poor input data, etc). As always, sensitivity testing of changes in parameters on the model results should also be performed.

For backward compatibility the previous Smagorinsky approach can be specified with the command:

```
Viscosity Formulation == Smagorinsky
```

If the Smagorinsky formulation is specified, the default viscosity coefficients automatically adopted are below.

```
Viscosity Coefficients == 0.5, 0.05 ! metric values
```

### 5.2.3 Viscosity Approach

In addition to a new viscosity formulation, the TUFLOW 2020-01 HPC solver has improved dry wall treatment for the eddy viscosity based on that developed for TUFLOW Classic in 2007 (see Section 3.6 of the TUFLOW Manual). The new approach provides better representation particularly in narrow channels and is based on that that developed for TUFLOW Classic (refer Viscosity Approach == Method B in the TUFLOW manual). This enhancement brings TUFLOW Classic and HPC closer together in terms of results when using the same other settings.

For backward compatibility, the previous approach can be specified using the .tcf command :

```
Viscosity Approach == Method A
```

### 5.2.4 Q&A on Turbulence

**Q:** Why are you changing the default turbulence representation in the 2020-01 TUFLOW HPC release as this means there will be some change in results from the 2018-03 release?

**A:** Turbulence is pronounced in areas of highly transient flow (high velocities, bends, ledges, flow contraction/expansion). Where the flow is more benign and/or bed roughness is high, turbulence is not so important as it only applies where there are strong spatial velocity gradients (for example, for

uniform flow in a straight rectangular channel the turbulence term is zero as there is no spatial velocity gradient).

The problem with the Smagorinsky form of turbulence closure (which is a large scale eddy turbulence model originally developed for coastal modelling) is that it is cell size dependent (is proportional to cell surface area) and tends to zero as the cell size tends to zero – this has historically not been a major issue as cell sizes have typically been greater than the depth, however, the general recommendation in the TUFLOW manual is to be careful of using cell sizes significantly smaller than the depth based on research and knowledge at the time (see Section 1.4 of the manual). However, as cells have been becoming finer and finer with the advent of GPU models this issue has increasingly emerged and is has become particularly pertinent if using a Quadtree or flexible mesh and very small cells relative to their depths are being used.

TUFLOW, many years ago, changed from purely Constant or purely Smagorinsky to Smagorinsky plus (a small amount of) Constant. This improved absorption of eddies into the streamlines behind a bluff body ([see Section 3.4 in this paper](#)) and helped by varying degrees the modelling at finer cell sizes.

However, an improved turbulence representation is needed for 2D schemes with fine-scale cells, preferably with parameter(s) that are valid across a wide range of hydraulic scales from flume model to large river systems. This need is especially the case for our new Quadtree mesh option and for flexible meshes as these meshes often incorporate fine-scale cells in areas of high flows.

**Q:** Does this mean the Smagorinsky plus Constant turbulence model (pre TUFLOW 2020-01 default) is wrong?

**A:** The Smagorinsky/Constant turbulence combination has served the industry well and can continue to be used where the cell sizes are not significantly smaller than the depth where highly transient flows are occurring. If the model is well calibrated (using conventional parameters), continuing to use the Smagorinsky/Constant turbulence option is certainly an acceptable approach provided the model cell size is not reduced. If the model cell size is reduced in part or all of the model, it will be important to demonstrate consistent results occur compared with the coarser cell size(s). If the model is uncalibrated, the same principle applies, but the lack of calibration will imply greater uncertainty in the results.

**Q:** What was the objective of the new turbulence approach?

**A:** Our aim was to have a turbulence scheme that, with the same parameter(s) produces accurate results across a wide range of scales from flume tests to large rivers, i.e. there is no or little need to calibrate the turbulence parameters like there is at the moment. The Wu turbulence seems to achieve this which is a major step forward for the industry.

We're not aware of any 2D modelling research or other software that has addressed the issue of turbulence at fine cell sizes and that can demonstrate the same parameter(s) apply to a wide range of hydraulic scales from flume to river. 2D schemes, as far as we're aware, either omit the turbulence scheme or offer it using either the Constant or Smagorinsky approach (we believe TUFLOW is the only one that allows a combination of Constant and Smagorinsky, and now Wu).

**Q:** What is numerical dispersion and why is it a problem?

**A:** 1<sup>st</sup> order spatial schemes are known to be numerically dispersive, which means that the solution stepping forward each timestep is less accurate than a higher order solution (an analogy would be fitting a line through three points is less accurate than a polynomial). The problem with numerical dispersion is that it has a similar effect to turbulence in that it diffuses (smooths out) the numerical solution, but it is, of course, totally unrelated to the physics of turbulence. Unfortunately, though, it may give the false impression of being an alternative or substitute for representing turbulence.

1<sup>st</sup> order schemes will artificially create a steeper gradient due to the additional effects of numerical dispersion (this was observed with the first incarnation of HPC – called TUFLOW GPU – which was a 1<sup>st</sup> order spatial solution and for early calibration of the Brisbane River modelling required lower Manning's n values to calibrate compared with 2<sup>nd</sup> order schemes).

Numerical dispersion also helps stabilise a model, but for the wrong reasons. A 2<sup>nd</sup> order scheme will have little or no measurable numerical dispersion, and typically becomes unstable or “bouncy” if the turbulence scheme is turned off. So, a 1<sup>st</sup> order scheme can exhibit turbulence like effects and good stability but is not physics based and will not be as accurate as 2<sup>nd</sup> order schemes. And 2<sup>nd</sup> order schemes generally need turbulence to be stable, but the simplification of turbulence (which is extremely complex) down to a solution that is valid across a wide range of hydraulic scales has always been a challenge for 2D schemes. Of note is that 1D schemes cannot represent turbulence as they have no knowledge of flow in the 2<sup>nd</sup> direction.

**Q:** Is the best approach of currently available methods prior to the 2020-01 release to use a 'Constant' viscosity value with no Smagorinsky with calibration/validation/sensitivity testing being required to select an appropriate value for the particular system & grid size being modelled?

**A:** Constant may be able to be used provided a good calibration can be demonstrated across a wide range of flows. Prior to implementation and testing of the turbulence methods for 2020-01 release, Smagorinsky plus Constant would be the recommendation. For example, the very heavily calibrated Brisbane River model, which uses a cell size of 30 m (which is indicative of the maximum depth for major floods, so is about as fine as we'd like to go before the cell size is less than the depth effect kicks in), calibrates very well using the same combination of (conventional) Manning's n values, minor additional energy losses on sharp bends (to cater for 3D secondary currents) and standard Smagorinsky/Constant eddy viscosity coefficients (TUFLOW defaults) across a wide range of hydraulic flows from tidal to five floods varying in magnitude from the 1 in 10 to the 1 in 100.

### 5.2.5 Limiting Manning's n for Turbulence Calculation (2022 Beta Feature)

As outlined above in the equations in Section 5.2.2, when using the Wu turbulence, the  $v_{3D}$  term has the Manning's n as a multiplier, and therefore the higher the Manning's n, the higher the turbulence term. Since the timestep used in HPC and Quadtree is dependent on the turbulence, high Manning's n values, particularly in deep water, can control the timestep.

Build 2020-10-AB introduces an option to specify the upper limit for the Manning's n value used in the turbulence calculation when using the Wu formulation. This only effects the Wu turbulence calculation, the specified Manning's n is still used for the bed friction term. To specify an upper Manning's n limit, a third value is specified in the Viscosity Coefficients command:

```
Viscosity Coefficient == <C3D, C2D, Upper Limiting n>
```

For example, to use the default Wu coefficients of 7 and 0 with an upper limiting Manning's value of 0.1 the .tcf command would be:

```
Viscosity Coefficient == 7, 0, 0.1
```

If the third value is omitted, e.g. “Viscosity Coefficient == 7, 0” then no upper limit is applied.

For the 2022 release a value of 0.1 is likely to be the default setting.

### 5.3 Flow Depth Used for Bed Friction Calculation (HPC Only, incl. Quadtree excl. SGS)

An improvement to model stability has been found by upwinding the depth of flow at a face used for the bed resistance calculation in the u and v momentum equations. The previous method is still available by selecting Method A in the .tcf file with the command

```
HPC Mannings Depth Approach == Method A | {Method B}
```

Note that models utilising Sub-Grid-Sampling use a different formulation again, and this command has no effect.

### 5.4 Non-Newtonian Flow (HPC Only)

TUFLOW HPC now supports modelling of non-Newtonian fluids. High-fidelity modelling of non-Newtonian fluids is complex and 3D in nature. However, with some assumptions, it is possible to model non-Newtonian fluids reasonably well in 2D. The assumptions are:

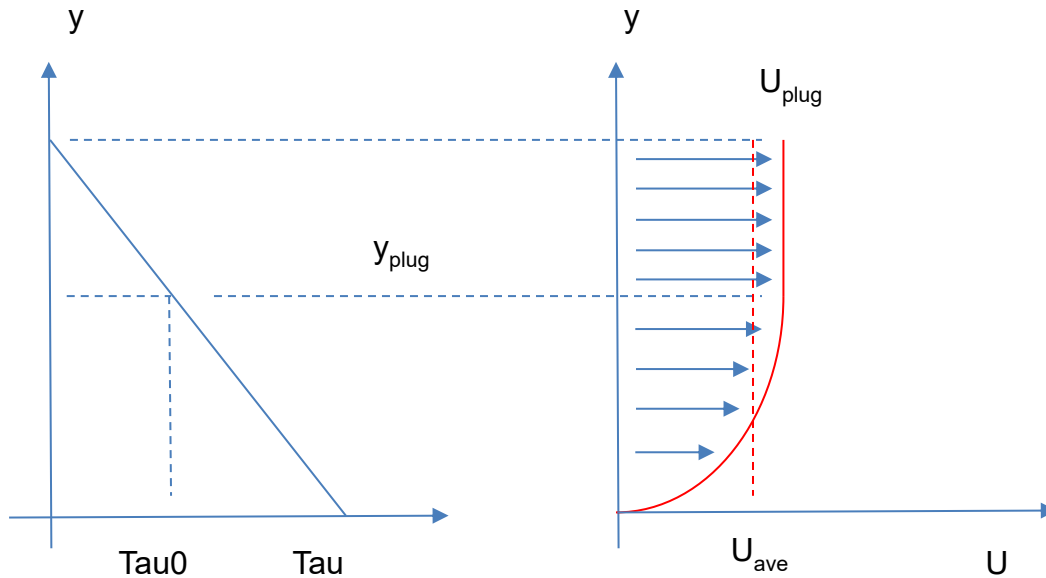
1. Turbulent eddy viscosity is not significant for non-Newtonian flows (which are usually highly viscous), and thus the non-Newtonian approach model is invoked with the “Viscosity Formulation ==” command and the 2D viscosity computed as the derivative of the power law (Hershel-Buckley) viscosity model
2. Acceleration effects are small and the fluid shear stress is linear with depth
3. The vertical velocity profile is no longer turbulent and the Manning's bed friction is no longer applicable. Instead, bed friction is computed by from the powerlaw viscosity model and the depth averaged flow velocity (see figures below).

The fluid shear stress (for flow that is shearing), is assumed to follow the power law model:

$$\tau = \tau_0 + k\dot{\gamma}^n$$

Where  $\tau_0$  is shear yield stress,  $k$  is a viscosity coefficient,  $\dot{\gamma}$  is shear strain rate, and  $n$  is shear thickening exponent, which must be non-zero and positive. Shear thinning fluids exhibit  $n < 1$ , shear thickening  $n > 1$ , and Newtonian fluids  $n = 1$ .

For flows where the bed shear stress exceeds the yield stress, a ‘plug flow’ velocity profile is computed as shown in **Figure 1**. For flows where the bed shear stress does not exceed the yield stress, the fluid is considered locked to the bed and does not flow.



**Figure 1 Non-Newtonian Plug Flow**

2D momentum diffusion is applied using an approximate viscosity coefficient computed from:

$$\mu_{2D} = kn|\dot{\gamma}|^{n-1}$$

where

$$|\dot{\gamma}| = \min\left(\frac{2|U|}{h}, |\nabla U|\right)$$

The equation for  $\mu_{2D}$  can produce unbounded results when  $n < 1$  and  $|\dot{\gamma}|$  tends to zero. Therefore, upper and lower viscosity limits,  $\mu_{low}$  and  $\mu_{high}$ , are applied. The bounded absolute viscosity is divided by water density to convert to kinematic viscosity and stored in the viscosity coefficient field, which is available for output (by including "T" in the map output data type, e.g. Map Output Data Type == h v T).

**Note:** In HPC the 2D momentum diffusion is handled explicitly and therefore can control the model timestep when the viscosity coefficient becomes large. For shear thinning models it is important to define an upper limit,  $\mu_{high}$ , that is only as large as necessary. We suggest starting with 1,000 [Pa s] and adjusting lower if the model is being strongly controlled by the diffusion control number (Nd).

**Note:** It is the user's responsibility to check whether the upper viscosity limit is influencing results in the region of interest. The lower viscosity limit,  $\mu_{low}$ , may be set to zero if desired.

Non-Newtonian related .tcf commands are:

```
Viscosity Formulation == Non-Newtonian
```

```
Viscosity Coefficients == k, n, muLow, muHigh, tau0
```

**Note:**  $k$  is absolute coefficient in  $[\text{Pa s}^n]$ ,  $\mu$  limits are in  $[\text{Pa s}]$ , and  $\tau_0$  is in  $\text{Pa}$ . As non-Newtonian fluids vary widely in coefficients there are no default settings for these parameters and they must be specified by the user based on the fluid being represented.

#### 5.4.1 Non-Newtonian Enhancements for 2020-10-AB

Build 2020-10-AB introduces an enhancement to the non-Newtonian functionality documented above. The non-Newtonian viscosity formulation was introduced in the 2020-10-AA release for modelling flows such as thick mud, and it replaced both bed friction and turbulent eddy viscosity with the Hershel-Bulkley fluid model (generalised power-law model). However, for deep fast flows of shear-thinning material, the resulting bed-friction and viscosity can become less than that expected for a water-only model, resulting in unreasonably high flow velocities. The model has subsequently been modified to:

- Add Manning's  $n$  bed friction to that computed by the non-Newtonian velocity profile solver, and
- Add the Wu 3D eddy viscosity contribution to the computed 2D viscosity using the default parameter of 7.0.

Both of these changes cause negligible differences when the mud flow is substantially more viscous than water, but provide a sensible lower bound for bed-friction and viscosity in the event that the mud appears to be behaving more like water (i.e. the flow may exhibit turbulence).

## 5.5 HPC GPU Advection-Dispersion (AD) Solver

**Note:** This new feature applies to TUFLOW HPC only and is GPU accelerated, AD is already supported in Classic using a different numerical solution.

Advection Dispersion (AD) capability is now supported for TUFLOW HPC. The AD inputs remain unchanged from those for TUFLOW Classic – refer to the Draft TUFLOW AD Manual available from the [TUFLOW Documentation page](#).

**Note:** When running AD with TUFLOW HPC there is currently a limit of one AD constituent per simulation (TUFLOW Classic allows up to 20 constituents) – this limitation will be removed in a future update/release.

The TUFLOW HPC solver tracks areal density of the passive tracer as the primary prognostic variable, with total tracer conserved to numerical precision by the finite volume scheme. The output field is then converted from areal density to volume concentration.

The diffusive fluxes of tracer across cell faces are computed using the full anisotropic diffusion matrix (Falconer et. al. as references in TUFLOW AD manual) rotated by flow direction.

$$\begin{pmatrix} \phi_{c_x} \\ \phi_{c_y} \end{pmatrix} = h \Delta x R D R^{-1} \begin{pmatrix} \partial C / \partial x \\ \partial C / \partial y \end{pmatrix}$$

$$R D R^{-1} = \begin{bmatrix} k_l \cos^2(\theta) + k_t \sin^2(\theta) & (k_l - k_t) \cos(\theta) \sin(\theta) \\ (k_l - k_t) \cos(\theta) \sin(\theta) & k_l \sin^2(\theta) + k_t \cos^2(\theta) \end{bmatrix} \frac{|U| h \sqrt{g}}{C}$$

$$\cos(\theta) = \frac{u}{|U|} \quad \sin(\theta) = \frac{v}{|U|}$$



where  $C$  is the tracer volume concentration, and  $k_l$  and  $k_t$  are the longitudinal and transverse diffusion coefficients.

### 5.5.1 Bug Fix for Advection Dispersion (AD) in HPC

Build 2020-10-AA fixes an issue with Advection Dispersion (AD) which in rare instances could allow a concentration to go negative, once this happened concentrations rapidly diverged with nearby cells getting very negative or very positive concentrations.

### 5.5.2 Support for Settling and Decay with HPC AD

Build 2020-10-AD introduces support for tracer settling and decay when running HPC, this functionality is detailed in the TUFLOW AD manual.

## 5.6 Specification of Maximum Timestep for HPC Simulations

Build 2020-10-AA introduces a maximum timestep for HPC simulations. When all cells in a HPC simulation are dry, the adaptive timestep is increased. Prior to the 2020-10-AA release of TUFLOW when running a model with significant periods in which all cells were dry, the timestep would increase to the map output interval. This could cause stability issues when the model first wets after a dry period, as the timestep has to reduce rapidly. For the 2020-10-AA release a maximum timestep can be specified with the command:

```
Timestep Maximum == {Auto} | <max timestep in seconds>
```

If not specified, or set to "Auto", the maximum timestep is based on the shallow water wave celerity where the wave speed is calculated based on  $c = \sqrt{gh}$  where the cell wet/dry depth is used as a water depth ( $h$ ). The max timestep is calculate as  $\max dt = dx/\sqrt{gh}$ , so for a 5m cell size with a 2mm wet/dry depth the maximum timestep is 35.7 seconds.

For models with infiltration and direct rainfall with a similar magnitude, it may be desirable to reduce the maximum timestep to avoid repeat timesteps.

## 5.7 2D Weir Flow Factor in HPC Solver (2022 Beta Feature)

Where flow in the 2D domain becomes upstream controlled, TUFLOW HPC triggers a weir flow approximation calculation by adjusting the water level gradient. Prior to the 2020-10-AB release, the adjustment was calibrated to match the broad-crested weir formula. Build 2020-10-AB introduced a beta feature to allow adjustment both globally and locally by specifying using the 2D weir factor in a similar manner to the WrF feature in the Classic solver. The same commands as used for TUFLOW Classic are used by the HPC solver as listed below.

.tcf command:

- "Global Weir Factor": Factor that adjusts globally. The default value is 1.

.tgc commands (to apply locally varied weir factors):

- "Set WrF": Applies the same WrF value to all cell faces.
- "Read GIS WrF": Reads the weir factor from a GIS layer in .mif or .shp format. This command can be used to adjust the weir factor locally, e.g. faces along a road embankment.
- "Read GRID WrF": Reads the weir factor from asc of flt grids.

- “`Read RowCol WrF`”: Reads the weir factor from a .mid or similarly formatted (comma delimited) file. The first three columns in the file must be “n, m, <WrF>”, where n and m are the 2D grid row, column and <WrF> is the weir factor value (not applicable for quadtree models).

**Note:** The global value (“`Global Weir Factor`”) and the spatially varying value (.tgc commands) are multiplied together (i.e. one does not replace the other).

TUFLOW Classic and HPC solvers use different approaches due to the mathematical nature of the two schemes. Therefore, the same adjustment to the WrF value in Classic and HPC can have differing effects on the results. However, for both solvers a WrF greater than 1 will decrease the flow efficiency thereby making the upstream water level higher for a given flow rate, and a value less than 1 will increase the flow efficiency.

Further research and provision of guidance of appropriate WrF values for different types of weirs is planned.

## 5.8 HPC Stability and Computational Improvements for 2020-10-AB

Build 2020-10-AB introduces the following changes which improve the stability for TUFLOW HPC simulations when using depth varying Manning’s n or Log-Law roughness.

### 5.8.1 Log-Law Roughness Improvements

The Log-Law roughness formulation becomes singular when the depth becomes less than approximately 1/11th of the roughness scale,  $K_s$ , with the resulting Manning’s n value becoming negative on the other side of the singularity. For typical values of  $K_s$ , this occurs at depths less than the default wet/dry threshold of 2 mm and does not cause an issue. However, for direct rainfall models a smaller wet/dry threshold of, for example 0.2 mm, maybe used and an instability issue may arise if using Log-Law. To solve this in a physically consistent manner, for Build 2020-10-AB onwards the calculation has been modified to limit the equivalent Log-Law Manning’s n value to a maximum of 10 as the depth approaches and goes below the singularity limit. This modification typically causes a negligible or no change in results.

### 5.8.2 Depth-Varying Manning’s n Improvements

For models that utilise depth-varying Manning’s n values, or the Log-Law roughness bed friction approach, it is possible for the model to experience rapid transitions in bed friction values from one timestep to the next. As the Wu turbulence model is dependent on bed friction, such rapid transitions have been found on occasions to cause stability issues. It has been found that limiting the relative rate of change of Manning’s n value to no more than 10% per time step avoids these stability issues in the rare situation they occur. This change was built into Build 2020-10-AB and has been found to typically cause a negligible or no change in results.

### 5.8.3 Dry Face treatment (HPC Only)

In HPC the momentum equations are evolved on any face that has at least one adjacent wet cell, which does include faces that are dry due to the water level of the wet cell being below the invert of the other adjacent cell. Under rare conditions it is possible for the momentum at a dry face to increase even though there is no actual flow across the face, leading to a runaway velocity condition that drives

the timestep down. A stability improvement was introduced in version 2020-10-AA that prevents this from happening, but it has subsequently been noticed that this fix can also cause very faint (generally sub-centimetre scale) noise in afflux differences.

An improved solution to the run-away velocity issue has been introduced in the 2020-10-AB release, as “HPC Dry Face Inertia Approach == Method B”. The improved method also brings the 2020-10-AB HPC results to be in closer agreement with the 2018-03-AE release (where “Defaults == Pre 2020” has been selected), as well as with the 2020-01-AB release. Method B is now set as the default for the dry face inertia approach, with Method A allowing backwards compatibility with the 2020-10-AA build. We strongly recommend moving to the 2020-10-AB release where afflux differences between scenarios are important.

## 6 Boundaries and Links

### 6.1 Improved HPC Water Level (H and QT) Boundaries (HPC Only)

TUFLOW HPC now defaults to a new boundary treatment for 2D HT and QT boundaries (.tcf command “HPC Boundary Approach == Method B”). Previously HT cells, or the HX cells from a QT boundary, were set to the level specified in the HT boundary data. However, this is not perfectly physical as it does not consider the dynamic head (kinetic energy of the water) and, in some situations, could lead to boundary instabilities – particularly at inflow boundaries. The new approach, which is a technique in CFD modelling, applies an energy correction during inflow to the surface elevations of the boundary cells according to the velocity head of the flow:

$$h_i = h_b(t) + \frac{\overline{U^2} - U_i^2}{2g}$$

Where  $h_i$  and  $U_i$  are the elevations and velocity magnitudes of the boundary cells,  $h_b(t)$  is the defined boundary surface elevation as a function of time, and  $\overline{U^2}$  is the average of the velocity magnitudes squared along the boundary. The energy correction is not applied for outflow, which is the same approach as used in CFD modelling.

The new approach can significantly stabilise inflow boundaries where unrealistic circulations along the boundary develop. Testing has also shown that there should be no need to use “Boundary Viscosity Factor ==” to help stabilise these boundaries. Some variation in elevation along the length of an inflow boundary may now be apparent, but the average elevation should closely match the prescribed boundary data.

If a HX cell is connected to the same 1D node, either via a QT boundary or if manually connected to the same 1D node then the new boundary approach applies. For HX boundaries that connect to more than one 1D node (eg. where a 1D channel is cut through a 2D domain), this new feature is not enabled.

For backward compatibility, specify “HPC Boundary Approach == Method A” in the .tcf file.

#### 6.1.1 Additional Approach for HPC Boundary Approach

As outlined above for “HPC Boundary Approach == Method B” the boundary energy correction applies to both HT boundaries and HX boundaries which are connected to the same 1D node (this includes QT boundaries). For Build 2020-10-AB a new method has been added (HPC Boundary Approach == Method C). For Method C the energy correction is applied for HT boundaries but not for HX cells connected to the same 1D node to provide consistency with HX cells connected to two 1D nodes.

For the 2022 release HPC Boundary Approach == Method C is likely to be set as the default.

### 6.2 Structure Loss Adjustment for SX Connected 1D Structures (Beta Function)

Build 2020-10-AA has a new beta option to extend the adjustment of contraction and expansion losses for 1D culverts and bridges to automatically adjust at 1D/2D SX connections. When there is no change in velocity magnitude and direction as water flows from/to the 2D domain into/out of a 1D structure, there should be effectively no entrance or exit losses associated with the horizontal

contraction/expansion of flow. However, prior to Build 2020-10-AA, the approaching/departure velocities in the 2D domain were not considered for adjusting the entrance or exit losses of the 1D structure requiring the user to enter appropriate (fixed) values.

Build 2020-10-AA introduces new functionality that takes the depth averaged velocities at SX cells and applies them as the approaching/departure velocities to adjust the entrance/exit loss coefficients of a 1D structure in the same manner as for a 1D/1D connection (see Section 5.7.6 Adjustment of Contraction and Expansion Losses of the TUFLOW Manual):

$$C_{entrance\_adjusted} = C_{entrance} \left(1 - \frac{V_{approach}}{V_{structure}}\right)$$

$$C_{exit\_adjusted} = C_{exit} \left(1 - \frac{V_{departure}}{V_{structure}}\right)^2$$

Where  $C_{entrance}$  and  $C_{exit}$  are entrance and exit loss coefficients specified by the “EntryC\_or\_WSa” and “EntryC\_or\_WSb” attributes in 1d\_nwk layer,  $V_{approach}$  and  $V_{departure}$  are depth averaged velocities at all connected SX cells,  $V_{structure}$  is the velocity in the 1D structure.

The new methods is available using the following new **.ecf** command (note this is an .ecf command, not a .tcf command):

```
Structure Losses SX == ADJUST | ADJUST SKEW | {FIX}
```

The “FIX” option is the default and is the approach taken in prior builds, i.e. no adjustment of the losses.

The “ADJUST” option adjusts the losses using the method described above, while the “ADJUST SKEW” option also considers the skew angle of the 2D velocity relative to the 1D structure entrance/exit. By default, the geographical angle of a 1D structure entrance/exit is automatically calculated from the 1d\_nwk GIS layer to calculate the normal velocity:

$$V_{approach} = u * \cos(\text{skew angle}) + v * \sin(\text{skew angle})$$

The skew angle can be also specified manually using the “B” attribute in a 2d\_bc SX layer with the following rules:

- If B = 0.01° ~ 360°, B defines the geometric angle of 1D structure entrance/exit.
- If B = -0.01° ~ -90°, B defines the absolute skew angle, i.e.:

$$V_{approach} = \sqrt{u^2 + v^2} * \cos(\text{skew angle})$$

- B attribute is normally left as zero, and in such case, the automatically calculated geometric angle of the connected 1D GIS network is used.
- To manually define a 0° angle, “B = 0.01°” can be specified where “0.01°” is rounded down to a geometric angle of 0°, while “B = -0.01°” is rounded up to an absolute angle of 0°. The “Zero Skew Angle” command can be used to make the cut-off value smaller than 0.01, in order to apply an even smaller user specified angle.

Subject to further testing, the “ADJUST” or “ADJUST SKEW” option will likely be set as the default in a future release, especially as Quadtree and SGS are allowing the in-bank channels to be accurately modelled in 2D removing the need for carving a 1D channel through a 2D domain, necessitating the

need for embedded 1D structures to have their losses adjusted by the approach/departure 2D velocities.

## **6.3 Bug Fixes and Minor Enhancements**

### **6.3.1 1d\_bc QT Boundary Duplication**

Fixed bug that duplicates inflows from a 1d\_bc QT region. This bug only occurs if in the rare situation that the region covers two or more nodes and the primary (first) node in the region has another QT inflow that is a point object or another QT region inflow where it is the only node within the region.

### **6.3.2 Bug Fix for HT Boundaries in HPC**

Build 2020-01-AA, HPC (including Quadtree) offered a new default HT boundary method (HPC Boundary Method == Method B) which utilised a total energy boundary during inflow conditions. The average water surface elevation would match the prescribed boundary value, but slight variations in water surface elevation along the boundary were allowed such that at each inflow cell the total energy elevation matched the prescribed boundary elevation plus the mean square velocity head along the boundary. In Build 2020-01-AA the mean square velocity head was computed over the whole boundary regardless of flow direction, whereas in Build 2020-01-AB the mean square velocity head is now computed only for regions of inflow. This has provided an improvement in stability and reduces instances of recirculation at HT boundaries. The change also applies to QT boundaries. The change only applies if using the HPC solver and should only cause minor changes for models with HT/QT boundaries with periods of mixed flow direction along the boundary.

### **6.3.3 Bug Fix for SX links in Quadtree**

For Build 2020-01-AB when running Quadtree, SX links (points, lines or regions) will now activate cells if they are inactive (code = 0) which is consistent with Classic or HPC without Quadtree. For the 2020-01-AA release, any SX links would not activate cells in Quadtree models. If an SX link was outside or snapped to the code polygon, this could cause different numbers of SX cells or TUFLOW to stop during initialisation with ERROR 2836.

### **6.3.4 Bug Fix for Storage Reduction Factor on SX links in Quadtree**

Build 2020-10-AA fixes an issue for Quadtree models with the distribution of 1D nodal storage to connected 2D SX cells. For builds prior to 2020-10-AA the SX storage factor was not being applied to the 2D cells. This could result in SX connections being slightly less stable when running Quadtree compared to the standard HPC solver.

Refer to the SX Storage Approach == command in the TUFLOW manual for further details.

### **6.3.5 Bug Fix for Global Rainfall Initial Losses in Quadtree**

Build 2020-10-AA fixes an issue which could cause the global rainfall losses in Quadtree to be incorrectly or not applied in the 2020-01 versions. The rainfall histogram in the log and check files correctly reported the histogram that was applied.

Note: see also changes outlined in Section 6.4 relating to global rainfall losses.

#### **6.3.6 Bug Fix for Quadtree Boundary Control File not stopping on ERROR**

Build 2020-10-AA now correctly stops the simulation if an ERROR is encountered when reading the boundary control file. For prior builds it was possible that an error could be output when reading in a GIS layer, however the simulation could continue. This could cause the simulation to crash later in processing or to incorrectly apply the boundary condition which generated the error.

#### **6.3.7 Bug Fix for Quadtree with Wu Turbulence**

The Wu turbulence model uses a physical length scale for calculating the viscosity factor, which is the lesser of water depth or horizontal distance to dry boundary. The Quadtree engine had a minor error regarding computing distance to the nearest dry boundary. This primarily affects models with fast flowing water against deep boundaries to dry cells. The error has been rectified in Build 2020-10-AA and back testing has shown negligible differences against the 2020-01-AB Build for most models. No backward compatibility has been provided for this change.

#### **6.3.8 Bug Fix for Small Timestepping in HPC (including Quadtree)**

For both the HPC and Quadtree engines improvements have been made in timestep control. Under certain flow conditions (involving a nearly dry face connecting a comparatively deep wet cell to a dry cell of higher elevation) it was possible for the face velocity to become unphysically high. This instability does not affect the accuracy of results in anyway, since the computed face flux remained at zero, but could cause a simulation to take significantly longer than necessary to compute due to the courant number control limit. The issue has been resolved in 2020-10-AA. The change may cause different timestepping for some models, with an associated, albeit very small, change in results. There is no backward compatible setting for this change.

#### **6.3.9 Bug Fix for NetCDF Rainfall Times**

Build 2020-10-AA fixes an issue which could cause a rainfall input grid in the NetCDF file format not to be updated for each rainfall grid if the smallest map output interval was greater than the rainfall grid interval.

#### **6.3.10 Bug Fix for NetCDF Rainfall Grids and Output Drive**

Build 2020-10-AA fixes an issue if using a rainfall control file, to apply gridded rainfalls using the NetCDF output format when also using an Output Drive == <drive letter> command. Prior to the 2020-10-AA release this would cause an ERROR 2720 when creating the NetCDF file and would stop the simulation.

#### **6.3.11 Bug Fix for Quadtree HX Links with CN Weighting Factors**

When running Quadtree, Build 2020-10-AA now supports HX links with multiple CN connections snapped to the same vertex (provided the "f" weighting attribute sum to 1.0).

Prior to Build 2020-10-AA, TUFLOW would report ERROR 2061 if HX boundaries in Quadtree had more than one CN connection snapped to the same vertex on the HX line. Note that due to the bug fix in Section 6.3.6 the 2020-01 version of TUFLOW may have continued to run after reporting ERROR 2061.



#### **6.3.12 Quadtree Support for SA Points**

Build 2020-10-AA now supports GIS point objects for 2d Source Area (2d\_sa) inputs for Quadtree meshes.

#### **6.3.13 Bug Fix for Quadtree QT Boundary with no Elevation Change**

Build 2020-10-AA fixes an issue with the nodal area storage for the hidden 1D nodes created at a QT boundary in a Quadtree mesh if all elevations along the boundary were within 0.1m. For the 2020-01 release the “A” GIS attribute, which is a storage multiplier (and defaults to a value of 5.0), was not included when creating the NA table. This means the hidden 1D node would have less storage than it should and could cause instabilities at the boundary.

This typically only affected flume or benchmark models as real-world model boundaries have elevations that vary.

#### **6.3.14 H2D Trigger Supported in Quadtree for Operational Structures**

Build 2020-10-AA now supports the 2D water level trigger (H2D) in Quadtree for operational structures. Previously this was unsupported for a Quadtree mesh, but did not necessarily produce an ERROR. An ERROR may have occurred if the trigger was in a refined region ‘ERROR 1521 – Could not identify 2D active cell in any 2D domain at coordinates <X>, <Y>’, but even if an ERROR did not occur TUFLOW may have been selecting an incorrect cell.

#### **6.3.15 Bug Fix for Reading RAFTS loc and tot with over 10,000 Timesteps**

Build 2020-10-AA fixes a bug that would cause RAFTS output files \*.loc and \*.tot to be read incorrectly when the number of timesteps in the RAFTS simulation was equal to, or more than, 10,000 timesteps.

#### **6.3.16 Bug Fix for HQ Boundary Types with Log-Law Material Approach**

Build 2020-10-AA fixes a bug that would cause an incorrect HQ curve calculation in TUFLOW Classic if the boundary selected any cells using the Log-Law material approach. This only affected the automatic approach when the user specified a slope in the ‘B’ attribute field. This bug only affected TUFLOW Classic as HPC prior to the 2020-10-AA build did not support the total HQ curve approach.

#### **6.3.17 Boundary Lines Creating an Invalid Quadtree Mesh**

Build 2020-10-AB fixes an issue that could occur along boundary lines that would cause the creation of an invalid Quadtree mesh. This situation would produce a ‘Should not be here’ error during initialisation causing model start-up to fail.

#### **6.3.18 HQ Boundaries Missing Cells**

Build 2020-10-AB fixes an issue along HQ boundary lines that would sometimes miss cells in areas of refinement transition. This would produce a ‘Should not be here’ error during initialisation causing model start-up to fail.

#### **6.3.19 Small SA Regions in Quadtree Meshes**

Build 2020-10-AB provides consistency for Quadtree meshes with fixed grids where no cell centres are within an SA region. For builds prior to 2020-10-AB an ERROR 2836 would be generated for a

Quadtree mesh and the simulation halted. For 2020-10-AB onwards the centroid of the SA region is used to select the Quadtree cell the point falls within. This is consistent with fixed grids. If the SA region selects no cells and the centroid of the SA region is not within an active cell then ERROR 2836 is still issued.

#### **6.3.20 Support for Streams in Quadtree Meshes**

Build 2020-10-AB now supports Read GIS Streams in conjunction with source area (2d\_sa) inflows for Quadtree meshes. Prior to this build streamlines were ignored and default approach of the lowest cell was used.

#### **6.3.21 Bug Fix for HPC with SA ALL and Streams**

Build 2020-10-AB fixes an issue with HPC for a model that has both Read GIS SA ALL in conjunction with streams defined with Read GIS Stream. For HPC builds prior to 2020-10-AB the inflow would be applied using the default SA approach (lowest cell, then all wet cells) rather the split between all cells. If the flag “Stream Ignore” had been set (e.g. `Read GIS SA ALL Stream Ignore ==`), then the correct SA approach was applied. See also 6.3.22 below.

#### **6.3.22 Update to Message 2208**

For both Classic and HPC when Read GIS SA ALL was used in a model with streams defined TUFLOW was reporting the message below:

CHECK 2208 - No active cells along streams within SA inflow <boundary name>

CHECK / WARNING 2208 messages now include the geographic coordinates.

See also 6.3.21 above.

#### **6.3.23 Improved Messaging for ERROR 2048**

Build 2020-10-AB provides improved messaging if a boundary object does select any 2D cells. Previously it was possible that ERROR 0115 (columns not in ascending order) would be spuriously generated instead of ERROR 2048. This change allows the cause of the error to more easily be identified.

#### **6.3.24 Boundary Updates to Ensure Reproducibility of Results on HPC and Quadtree**

Build 2020-10-AB introduces very minor changes to the calculation for all boundary types to ensure that these provide reproducible results for repeated simulations. Previously, in rare occasions, differences in results could be seen between repeat simulations, this behaviour was occurring due to floating point precision calculations. For example, at a HQ boundary the water level applied is based on the total flow leaving the boundary. When summing flow across boundary cells the order in which the numbers are summed can slightly change the result due to numerical rounding. Due to the highly parallelised nature of GPU compute, this could result in typically very subtle differences in water levels applied at the boundary. Build 2020-10-AB introduces a more robust method for these summations to ensure the pattern of the summations are in a repeatable order.

#### **6.3.25 Improved Error Message if 1D Boundary is Read into 2D Boundary Control File**

Build 2020-10-AB provides better ERROR messaging if a 1d\_bc GIS layer is read into the 2D boundary control file (.tbc file). In this instance ERROR 2340 will be output and the simulation will terminate. Previously TUFLOW would produce ERROR 2041 which is less clear as to what is causing the issue.

#### **6.3.26 SX Cell Elevation Check Not Conducted for 2d\_bc SX lines with Multiple Segments**

Build 2020-10-AB fixes an issue with 2d\_bc SX lines with the “Z” flag. Prior to this build, 2d\_bc SX lines with multiple segments could cause a malfunction of the “SXZ” flag that lowered the 2D cell centre elevation to match the invert.

#### **6.3.27 Bug fix for SX Cell Elevation Check when using 1d\_nwk nodes**

Build 2020-10-AB fixes an issue with the SX cell elevation check in Quadtree when using SX connections defined on 1D network nodes. Previously, this check could fail to identify cells where the 2D elevation is above the 1D node and no ERROR 2050 was issued. This change was made after the Beta3 version of 2020-10-AB.

#### **6.3.28 Bug fixes to SX Sag connections when more than one 2D cell used**

Build 2020-10-AB fixes two related issues with Sag type SX connections (SXS) if using Classic or HPC solvers (excluding Quadtree). The fix only applies to SX connections at non-pit nodes in the 1d\_nwk layer, or 1D pits with SXS specified in the “Conn\_1D\_2D” attribute for pit nodes, and only applies to SX sag connections that select more than one 2D cell.

Prior to this build:

- When selecting additional cells, the lowest cell from all 8 surrounding cells (including diagonals) was used, this should have been the lowest of the 4 cells that share a face (diagonals not used).
- The cell centre (ZC) elevation for additional cells was being lowered to the 1D bed level if the cell elevation was above this.

For Build 2020-10-AB the behaviour should now be the same between Classic and HPC (both fixed grid and Quadtree) solvers. This change was made after the Beta3 version of 2020-10-AB and may cause a slight change in results. Method A can be used for backward compatibility.

SX Node Cell Selection == [ METHOD A | {METHOD B} ]

#### **6.3.29 Speed Improvements for Quadtree Boundaries**

Build 2020-10-AB introduces speed improvements for Quadtree models with numerous inflow boundaries (SA and RF). Previously, for Quadtree models the number of inflow boundaries was dynamically determined, meaning there was no limit to the number of boundaries applied. However, this approach was found to be significantly slower than the approach for fixed grid models where large numbers of inflow boundary locations occurred. For the 2020-10-AB version, for Quadtree models the total number of inflow boundaries is now limited to 65,535 with no more than 4 overlapping on a single cell. This is the same approach and limits as used by the HPC solver for fixed grids.

#### **6.3.30 Cell Face Elevation Lowered by SX boundary Z Flag**

Prior to 2020-10-AC, SX polylines and regions with a Z flag lower cell centre elevations only. This creates lowered SX cells with higher, unchanged cell side elevations that can potentially act like “buckets” and prevent water from flowing in/out of the SX cells until the cell sides are overtopped. 2020-10-AC introduces a minor change that SX regions with a Z flag also lower the cell face elevations. This change can improve the flow of water between SX cells at low flows, especially when multiple rows of SX cells are selected. Note that this function only applies to SX regions using the Z flag, while SX lines with the Z flag only modify the cell centre elevations.

#### **6.3.31 SA Triggers Fixed in HPC**

Build 2020-10-AD fixes a bug in SA triggers for HPC. Previously SA triggers would not trigger and no warning or error message was produced.

#### **6.3.32 Bug fix for QT boundary for Quadtree models**

Build 2020-10-AD fixes a rare issue with QT boundaries in HPC models using Quadtree mesh. The bug would cause the TUFLOW simulation to stop during initialisation and report that the boundary line was not supported.

#### **6.3.33 Fixed elevation check for Quadtree Models with HX links**

Build 2020-10-AD correctly checks that all cells along HX links are above the elevation of the 1D node bed, if a HX cell elevation is below the 1D level an ERROR 2044 message was generated. Prior to Build 2020-10-AD, this error was not generated and could cause water to immediately start flowing from the HX cell, this would show in the output files as an unacceptably high mass balance error.

#### **6.3.34 Bug fix for Flood-Modeller with HX links and Quadtree Solver**

Build 2020-10-AD introduces a fix for Flood Modeller-TUFLOW models which utilise a Quadtree mesh and include HX type links. When assigning flows from HX cells to the Flood Modeller unit, the upstream unit is now assigned the 1D/2D flow exchange as required by the Flood Modeller 1D solution. For Quadtree simulations prior to 2020-10-AD the closest unit was assigned rather than the upstream unit. The volume transferred is preserved in either scenario, however, the fix typically increases the stability of the 1D/2D link as well as align with the mechanics of the Flood Modeller scheme. No backward compatibility has been provided for this fix. This change does not affect Classic and HPC non-Quadtree models, which correctly assign the flow to the upstream unit.

#### **6.3.35 Significantly Improved Runtimes of 1D/2D Models Using Quadtree Solver**

Build 2020-10-AD improves the speed performance of Quadtree models when linked to 1D via either SX or HX connections. This includes TUFLOW 1D (ESTRY) models as well as external 1D engines. No change in results should be evident from the speed improvement. For models with a small number of connections (e.g. less than 100), the speedup is typically less than a factor 2, however, the benefit improves the more 1D/2D connections are present. For example, a model with ~2,200 1D channels and ~6,500 2D HX or SX cells, the runtime reduced from 85min to 10min.

### 6.3.36 Improved Messaging for WARNING 0109

2020-10-AE improves messaging for a 'Should not be here' when reading from boundary XF files. Previously a 'Should not be here' was encountered if the previous model run generated XF files while reading boundary data before TUFLOW exited with ERROR 2048. The 'Should not be here' message has been replaced by WARNING 0109 which is the same message generated if not reading XF files.

## 6.4 Changes for Global Rainfall Losses and Material Rainfall Losses

Build 2020-10-AA fixes a discrepancy between how rainfall losses are applied to global rainfall boundaries between Classic and HPC.

As per the TUFLOW manual Global Rainfall losses (.tbc commands Global Rainfall Initial Loss == and Global Rainfall Continuing Loss ==) are only applied to Global Rainfall Boundaries for both Classic and HPC. These commands must occur before a Global Rainfall BC is applied.

Prior to the 2020-10-AA Build, TUFLOW Classic did not apply rainfall losses in the materials file to any global rainfall boundaries, this is the behaviour documented in the TUFLOW manual. However, TUFLOW HPC did apply these losses. A summary of the loss application for both Classic and HPC is outlined in the table below for Builds prior to 2020-10-AA:

	Solution Scheme	
	Classic	HPC
Global Rainfall Losses	Applied	Applied
Material Rainfall Losses	Not Applied	Applied
Both Global and Materials Losses	Only Global Applied	Both losses applied

For the 2020-10-AA Build, if rainfall losses have been specified in the materials file these are applied to global rainfall and any Global Rainfall losses are ignored. A WARNING 2244 is output if Global losses are also specified. This is summarised in the table below for builds 2020-10-AA onwards

	Solution Scheme	
	Classic	HPC
Global Rainfall Losses	Applied	Applied
Material Rainfall Losses	Applied	Applied
Both Global and Materials Losses	WARNING 2244, Material Losses Applied	WARNING 2244, Material Losses Applied

A new .tcf command for TUFLOW Classic simulations can be used to control if the materials losses are used.

```
Global Rainfall Use Material Loss == OFF | {ON}
```

If using Defaults == Pre 2020-10 with TUFLOW Classic, this is reverted to OFF. Refer Section 12 for details on backward compatibility.

#### 6.4.1 Bug Fix for Global Losses and HPC

Build 2020-10-AB fixes an issue that could cause WARNING 2244 to be issued and global rainfall losses to be ignored for HPC simulations (including Quadtree) even when there were no rainfall losses assigned to materials.

#### 6.5 HPC New Drying approach with Soil Infiltration

Build 2020-01-AB introduces a new approach for handling drying when soil infiltration is present. This aims to avoid cyclical wetting and drying when the rainfall boundary rate is less the infiltration rate.

Infiltration is only computed for wet cells, consequently if the applied rainfall rate is less than the infiltration rate then the cell will repeatedly dry then wet. This can cause time output results to appear erratic and in rare cases has resolved stability issues if SGS was implemented. Currently infiltration rate is limited to the available water within a cell plus a precision adjustment, allowing the cell to dry. An improved approach is to remove the precision adjustment for cells that have positive rainfall, so they remain “wet” but with near zero available water, thus eliminating the cyclical drying and wetting. The new option is available as “`HPC Infiltration Drying Approach == Method B`”. The previous approach is available as Method A.

For models with SGS on the default approach is to use Method B, while for Non SGS models the default approach is to use Method A for backward compatibility. Method A is consistent with the 2020-01-AA and 2018-03 releases of TUFLOW, but for non-SGS models there maybe be benefits in switching to Method B noting that this will infinitesimally reduce the amount of infiltration and therefore infinitesimally increase the amount of surface water. For SGS activated models Method B is recommended for improved stability in rare cases and because the differences between methods is even less due to the much smaller wetted area in the cell at the drying point.

`HPC Infiltration Drying Approach == Method A | Method B`



## 6.6 HPC (including Quadtree) support Virtual Pipes

For Build 2020-10-AA virtual pipe functionality is included for HPC grids and Quadtree meshes.

### 6.6.1 ESTRY Support Virtual Pipes Outlets

For Build 2020-10-AA virtual pipe outlets (VPO) can be connected to, and discharge into, the upstream or downstream end of a channel in ESTRY for HPC including Quadtree meshes.

TUFLOW will connect the virtual pipe outlet to ESTRY if a VPO pit is snapped to the upstream or downstream end of a channel. A VPO pit cannot be snapped in the same location as a user defined node, another pit (including 'Q' type), or manhole.

VPO types are not yet supported in TUFLOW Classic and therefore this feature is not yet supported in TUFLOW Classic.

## 6.7 HPC (including Quadtree) Support Classic Style HQ Curve

For Build 2020-10-AA the TUFLOW Classic style HQ boundary is now supported in HPC including Quadtree.

The Classic style HQ boundary computes the flux across the entire boundary line and uses an automatically generated rating curve to apply a water level to the model. As with Classic, users can specify a slope (m/m) in the 'B' attribute field and TUFLOW will automatically generate a rating curve based on the Manning's equation. Alternatively, users can specify a rating curve by specifying a boundary name in the 'Name' field (the 'B' attribute must be set to zero). For more information on Classic style HQ boundaries see the TUFLOW Manual (Section 7.4 of the 2018-03 Manual).

The rating curve generated by HPC will be identical to that produced by Classic when SGS is **not** on as SGS is not supported in Classic. Rating curves generated by Quadtree can differ slightly, even when calculated using an identical mesh to that of Classic and HPC. This is caused by slightly different calculation of length needed due to the possibility of refinement areas. If SGS is on, the generated rating curve can be different from that generated with SGS off.

The 'S' flag (fitting a cubic spline to the data) is not yet supported in HPC or Quadtree, nor is it recommended.

For the 2020-10-AA release of TUFLOW, the default approach is to use a total or Classic style HQ boundary, rather than the cell by cell approach in prior builds.

Backward compatibility can be achieved by using 'CELL' in the following command to invoke the previous HPC approach of applying the water surface slope on a cell by cell basis:

```
HPC HQ Boundary Approach == {TOTAL} | CELL
```

### 6.7.1 Bug Fix for Quadtree if HQ Boundary Selects Only a Single Cell

Build 2020-10-AB fixes an issue with HQ boundaries if the HQ boundary only selects a single Quadtree cell. The occurrence of this issue is very rare, but the fix may increase the water level near single cell HQ boundaries in Quadtree mesh models.



## 6.8 HPC Support for Reducing Groundwater Level

Build 2020-10-AB optionally allows for soil moisture to be reduced when applying a negative rainfall boundary. Prior to this build, a negative rainfall boundary only applies to water above the ground, it does not change the soil moisture / groundwater level. Therefore, when a cell dries out the cumulative infiltration and soil moisture remain constant.

For Build 2020-10-AB, the option to allow negative rainfall to apply to groundwater has been added, for example, to represent a gradual loss of groundwater. To enable this, the following commands can be used:

```
Soil Negative Rainfall Approach == {NONE} | FACTOR
```

```
Soil Negative Rainfall Factor == <multiplication factor> {1.0}
```

If “Soil Negative Rainfall Approach ==” is set to FACTOR then when a cell becomes dry the negative rainfall boundary will remove groundwater. The multiplication factor can be used to apply the boundary at a different rate for surface water and groundwater. For example, if the “Soil Negative Rainfall Factor ==” is set to 0.5 then when negative rainfall is applied to a dry cell the groundwater volume removed is 50% of the rate of that applied to a wet cell (i.e. surface water). When a cell is wet, the boundary applies to the surface water area, noting that if SGS is applied the partial surface water area is used if the cell is not fully wet. The default factor is 1.0 meaning the same boundary volumes are applied to surface water and ground water.

The default approach of NONE does not apply negative rainfall to water that has infiltrated into the soil groundwater store.

## 7 Outputs and Check Files

### 7.1 Time-Series Output in NetCDF format

1D and 2D timeseries based output (1D results, 2d\_po, etc) can now be directed to a NetCDF file using “Time Series Output Format == {csv} | NC” in the .tcf. If desired, both .csv and NetCDF can be specified using this command.

One of the advantages of NetCDF is all the timeseries output is in a single compressed .nc file, rather than multiple uncompressed .csv files. This can be useful for large 1D models or large amounts of 2d\_po data.

The feature was developed primarily for integration with FEWS, but will also appeal to users who like to perform customised post-processing using Python, Matlab or other platforms that recognise the NetCDF format.

The QGIS TUFLOW Viewer has been updated to support the NetCDF format as an alternative to the .csv files.

#### 7.1.1 Bug Fix for NetCDF Time-Series and Quadtree

Build 2020-01-AB fixes a bug which caused a blank NetCDF time-series output to be written for Quadtree models, Classic and HPC simulations were not affected. CSV files for Classic, HPC and Quadtree were unaffected.

### 7.2 Extra Plot Outputs

Timeseries of 1D node mass error, node flow regimes, channel flow regimes, and structure losses are now output with the .csv or .nc plot data in the results\plot\ folder. This data can now be read into the GIS viewing platforms. Previously this data was only output to the ESTRY Output File (.eof) and to the GIS output files \_TSF or \_TSL. The new output csv files are listed in the table below (and are replicated in the .nc output).

results\plot\csv\	Result Type
_1d_NF.csv	Node flow regime
_1d_TSMB.csv	Mass balance error
_1d_CF.csv	Channel flow regime
_1d_L.csv	Channel losses

#### 7.2.1 Flow Regimes

Flow regime in “\_1d\_NF.csv” and “\_1d\_CF.csv” are reported as the regime flag(s) as described in the TUFLOW Manual Table 12-1.

#### 7.2.2 Mass Balance Error

The “\_1d\_TSMB.csv” reports the mass error in m<sup>3</sup>/s for all plot output timesteps for each node. The output is similar to the “\_1d\_MB.csv” output which is written to the 1D output folder.

### 7.2.3 Channel Losses

The “\_1d\_L.csv” reports the form losses being applied at each channel. The number of loss columns output for each channel is dependent on the channel type. For example, ‘C’ and ‘R’ type channels will output three loss columns (Entry Loss, Additional Form Loss, Exit Loss) and ‘BB’ bridges will output two loss columns as per Section 5.7.2.4 of the TUFLOW Manual.

## 7.3 NetCDF Map Output

NetCDF Map Output is now synchronised at each map output interval so it can be accessed as the simulation progresses. This enhancement is for both “NC” north-south aligned raster and “CC” cell centred NetCDF outputs.

### 7.3.1 NetCDF Raster Grid Output (NC) Enhancement and Bug Fix

Build 2020-01-AB writes additional projection information to the ‘NC’ grid output format to enhance the reading and viewing of the output in GIS platforms when the `SHP Projection ==` command is specified in the TCF.

Build 2020-01-AB also fixes an issue in the ‘NC’ grid output where the X-coordinate variable standard name was being misspelt.

### 7.3.2 NetCDF Raster Grid FEWS Configuration .xml File

Build 2020-01-AB can write out a FEWS regular grid configuration file (.xml). To do this, the geodatum string must be specified, via the .tcf command:

```
FEWS Geodatum == <geodatum string>
```

e.g.

```
FEWS Geodatum == GDA94 MGA Zone 56
```

This .xml file is written if the FEWS Geodatum is specified and the Map Output Format includes the “NC” format. The .xml file is written to same results folder as the grid .nc, and has the file name “Grids.xml”. This follows the FEWS documentation here:

<https://publicwiki.deltares.nl/display/FEWSDOC/06+Grids#id-06Grids-regular>

## 7.4 Quadtree Models – Map Output Considerations

### 7.4.1 Overview

In a Quadtree model with varying cell sizes, care needs to be taken over interpreting the grid map output result. By default, the smallest Quadtree 2D cell size is used to set the grid output cell size. This may generate excessive number of output grid cells inside the largest Quadtree 2D cells. For example, with 4 levels of refinement, the largest 2D cell will have  $8^2$  (64) output grid cells. TUFLOW uses a TIN interpolation using the output values at the 2D cell centres and corners. However, this may cause a “saw tooth” effect for large Quadtree cells with large differences in values at the cell centres and corners. The following command is provided for the user to change the interpolation method for generating grid outputs.

```
GRID Output Interpolation == CC EXCLUDE H | CC ALL | {TIN}
```

- CC EXCLUDE H uses the 2D cell centre value for all grid output cells within the 2D cell, except for the water level output (h) which still uses the TIN option below so that the water level grid output slopes according to the TIN water surface and can be used for post-processing of results.
- CC ALL applies the 2D cell centre value for all grid output cells within the 2D cell for all outputs including water level.
- The default setting, TIN, interpolates from the TIN generated using the cell centre and cell corner values, ie. the same TIN as used for X MDF output if SMS TRIANGLES is specified.

## **7.5 Sub-Grid Sampling (SGS) – Map Output Considerations**

### **7.5.1 Overview**

If using Sub-Grid Sampling (SGS – see Section 3.2), care needs to be taken over interpreting the results, particularly the map output. SGS will produce partially wet cells along the wet/dry boundaries and potentially create new flow paths that were previously blocked. Therefore, the number of wet cells is usually more than for a no SGS run, all else being the same – a cell will be flagged as wet in the TUFLOW output if only partially wet. The greater the cell size, relative to the resolution of the terrain data, the more pronounced these effects are.

One of the significant benefits is the water surface level (h) output will now extend into dry ground as the modelled flood extent now includes all the partially wet cells around the wet/dry perimeter. The benefit is that it is no longer needed to buffer the water surface output if generating a high-resolution depth output by subtracting the DEM from the water surface.

The images in Figure 2 show the depth results for two cell sizes of 20 and 60 m for a small area within a catchment wide direct rainfall model. Without SGS water is trapped at choke points where the 2D cells are too coarse for the primary flow path to be reasonably represented and flow hydrographs are attenuated, more so for the larger 60 m cell size. With SGS turned on water is no longer trapped producing much improved conveyance and substantially less attenuation of the flood wave.

As shown in Figure 2 the SGS results show a greater coverage of wet cells due to the presence of partially wet cells. As the cell size increases the coverage increases as can be seen between the 20 and 60 m resolutions. For the 60 m case, nearly all cells shown as wet would be partially wet due to the coarseness of the 60 m cell size. New commands are provided for the user to exercise greater control over the outputs as discussed in the following sections.



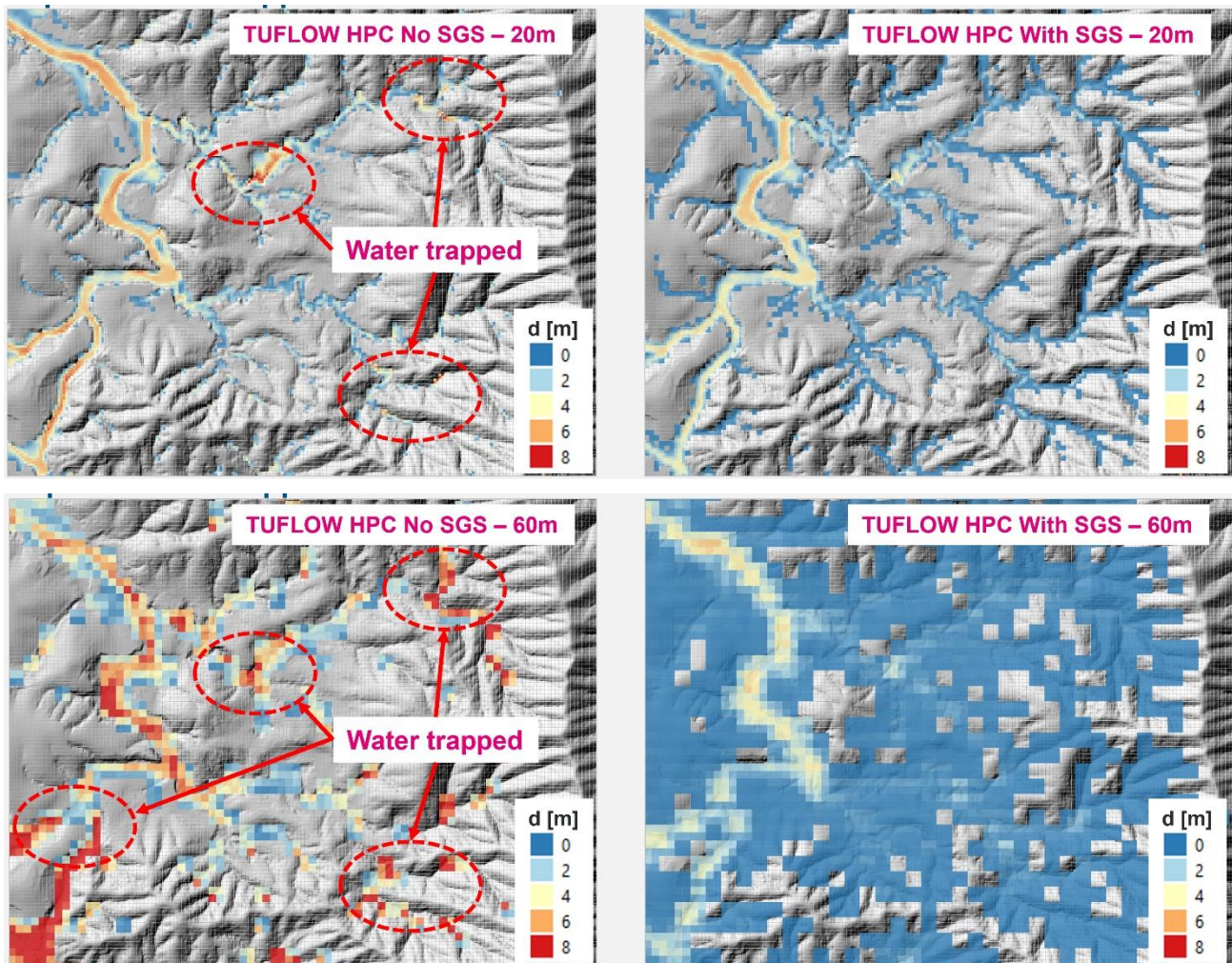


Figure 2 Effect of Using SGS on a Catchment Wide Model

### 7.5.2 Map Cutoff SGS == Commands

**Note:** To ensure consistency between SGS map cutoff depth and SGS depth interpolation method, “Map Cutoff SGS” and “SGS Depth Interpolation Approach” commands have been merged into a single command “SGS Depth Output” for Build 2020-10-AB. The “Map Cutoff SGS” command is still supported in Build 2020-10-AB but will be retired in a future release. Please use the “SGS Depth Output” command as documented below to set the cutoff option, and use “Map Cutoff Depth” command to set the cutoff value in a similar manner to non-SGS models.

If Sub-Grid Sampling (SGS) is applied the command “Map Cutoff SGS == ” can be used to control the elevation below which cells are shown as “dry” in the map output. The options for “Map Cutoff SGS” are:

Map Cutoff SGS == Average | {Exact} | Median | Minimum | Percentile

- Average – Uses the mean elevation for the cell.

- Exact – Uses the elevation at the exact location of the cell centre. I.e. the elevation that would be sampled if SGS was not used. This option was added as the default setting for the 2020-01-AB release.
- Median – uses the median (50<sup>th</sup> percentile) elevation for the cell.
- Minimum – uses the minimum elevation for the cell. This was the default for the 2020-01-AA release.
- Percentile – Define the cutoff elevation based on a user specified set percentile. This requires a second argument, separated by vertical bar “|”, which is the percentile to use. For example, “Map Cutoff SGS == Percentile | 25” will use the 25th percentile of the SGS elevations sampled within the cell.

For the average, exact, median, and minimum options, a optional second argument can be specified which is the depth above the datum. For example, “Map Cutoff SGS == Average | 0.05” will use an elevation 0.05m above the average SGS elevation sampled for the cell.

A combination of “Map Cutoff Depth ==” and “Map Cutoff SGS ==” can be used in conjunction, with the higher elevation used. When SGS is on, the Map Cutoff Depth refers to a depth above the cell minimum elevation. For example:

```
Map Cutoff Depth == 0.05
```

```
Map Cutoff SGS == Percentile | 25
```

Will use the maximum of 0.05m above the lowest SGS sampled elevation, or the 25th percentile of the SGS elevations sampled within the cell.

The elevation used for setting whether a cell is wet or dry in the map output, is reported in the \_grd\_check file as the “Z\_Map\_Cutoff”. This is the SGS Depth Output elevation for each cell plus the map cutoff depth.

**Note: That this can be controlled based on the map output data type, see Section 7.5.4.**

### 7.5.3 SGS Interpolation Commands for Map Output

The following .tcf commands can be used to control how map output values that use water depth are calculated for the mesh-based outputs (e.g. DAT, XMDf) if SGS is used. Note, these commands only affect depth dependent outputs (e.g. d, q, hazard outputs, BSS, SP). Outputs such as water surface level and velocity are not affected.

```
SGS Depth Interpolation Approach == {EXACT} | AVERAGE | MEDIAN | MINIMUM | PERCENTILE
```

**Note: As of Build 2020-10-AB this command has been changed to**

```
SGS Depth Output == {EXACT} | CELL AVERAGE | MEAN | MEDIAN | MINIMUM | PERCENTILE
```

“SGS Depth Interpolation Approach” command is still supported by Build 2020-10-AB, but it will be retired in a future release.

The available options are:

- EXACT (the default option) calculates the depth using the ZC and ZH elevations that would be sampled at their respective locations if SGS was not used, i.e. the elevations that would be sampled exactly at the cell centre (ZC) and cell corner (ZH) locations if SGS was not applied. This is the default setting.
- CELL AVERAGE: please refer to Section 7.5.6
- AVERAGE (**Note: “AVERAGE” has been replaced by “MEAN” in Build 2020-10-AB**) uses the average Z value assigned to the cell centre and the cell corners from the SGS sampling, i.e. the map output shows the average depth within a cell / around a cell corner (also see [SGS ZH Sample Ratio](#) below). This option was added for the 2020-01-AB update.
- MEDIAN uses the median (50th percentile) elevation for the cell. This option was added for the 2020-01-AB update.
- MINIMUM uses the minimum Z value, i.e. the map output shows the maximum depth within a cell / around a cell corner (also see [SGS ZH Sample Ratio](#) below).
- PERCENTILE uses an elevation based on the specified set percentile. This requires a second argument, separated by vertical bar “|”, which is the percentile to use. For example, “[SGS Depth Interpolation Approach](#) == Percentile | 25” will use the 25th percentile of the SGS elevations sampled within the cell. This option was added for the 2020-01-AB update.

Note that the [SGS Depth Interpolation Approach](#) and [Map Cutoff SGS](#) should generally be kept to the same datum, for example if Map Cutoff SGS is set to minimum, but SGS Depth Interpolation approach is set to Average, then there may be areas which show as “wet” in the depth output, but have a depth of 0.

The elevations used for the SGS Depth Output are output in the `_zpt_check` file as the “ZOut” attribute.

`SGS ZH Sample Ratio == <ratio>`

The above command can be used to control the area used for SGS at ZH locations. The area is set to `<ratio>*cell_area` to sample and generate Z values around a cell corner (the sampled elevations are not used in the hydraulic computations, only for map output). The default setting for this command is 1.0. The corner elevations are also used to enable the `_DEM_Zmin` check file (see 3.2.7) to be written. Prior to the 2020-01-AB release the SGS ZH Ratio was dependent on the SGS Depth Interpolation Approach.

#### 7.5.4 SGS Mapped Extent Options

Further to the above command “Map Cutoff SGS”, Build 2020-01-AB, allows the user to control which map data types are present as “wet” for partially inundated cells and which data types are trimmed based on the Map Cutoff Commands (see above). The following `.tcf` command can be used control, If set to full, partially wet cells are shown as wetted in the map output



```
SGS Map Extent <Full or Trim> == <map data types or "All">
```

For example “SGS Map Extent Trim == All” will trim all mapped data types the Map Cutoff extents.

And “SGS Map Extent Full == h v hazard” will set the water level, velocity and any hazard outputs to the full extent, while all other data types will be trimmed.

The default is for water level output to show the full extent, but for all other data types to show the extent trimmed, i.e. `SGS Map Extent Full == h`.

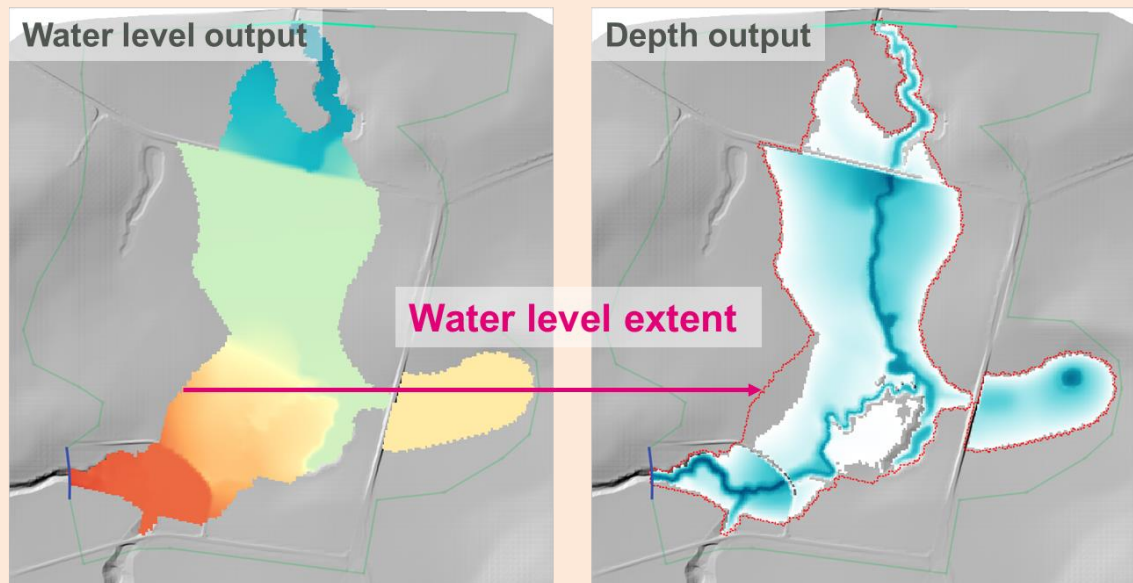
Build 2020-10-AB fixes an issue that “SGS Map Extent Full” option was not applied properly if Map Cutoff is also used. Outputs are now set to the full extent regardless of the Map Cutoff Depth value. Note that this command does not apply to the high-resolution output (Section 3.7.4).

### 7.5.5 SGS Corner Interpolation Enhancement at Wet/Dry Interface

Build 2020-10-AA fixes a corner interpolation issue at the Wet/Dry Interface. When only some cells around a corner are wet, TUFLOW needs to determine how to average the wet cells’ values to interpolate a cell corner value of the dry cell. It has been found that the minimum SGS cell elevation at dry cells which was used for the corner water level interpolation under certain situations, leads to an inconsistency between the cell corner and cell centre water level output values. The issue has been fixed in Build 2020-10-AA.

### 7.5.6 SGS Cell Averaged Depth Output (2022 Beta Feature)

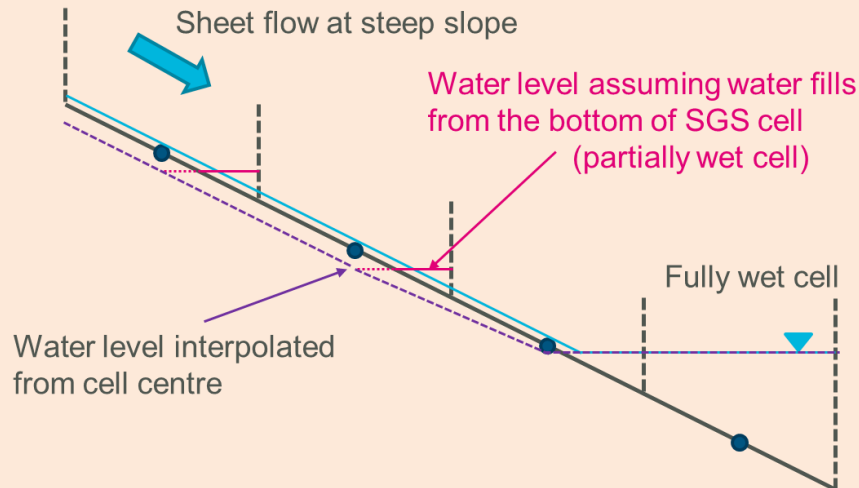
With the default SGS depth output option, the user may get a mismatch of the water level and depth output extent as shown in Figure 3, especially in a direct rainfall model.



**Figure 3 Example of Water Level and Depth Output from SGS Models**

This mismatch occurs at sheet flow cells, where thin layers of water flow over a steep slope. The water depth is set as “volume / cell area” in the hydraulic calculations (blue water level line in Figure 4), but TUFLOW output assumes the water is filled from the lowest SGS sampled elevation which

occurs at the bottom of the sloping cell (pink water level lines). Since the water level is interpreted from the cell centre, this may generate a water level below the ground elevation at the cell centre, therefore producing a negative depth for map output.

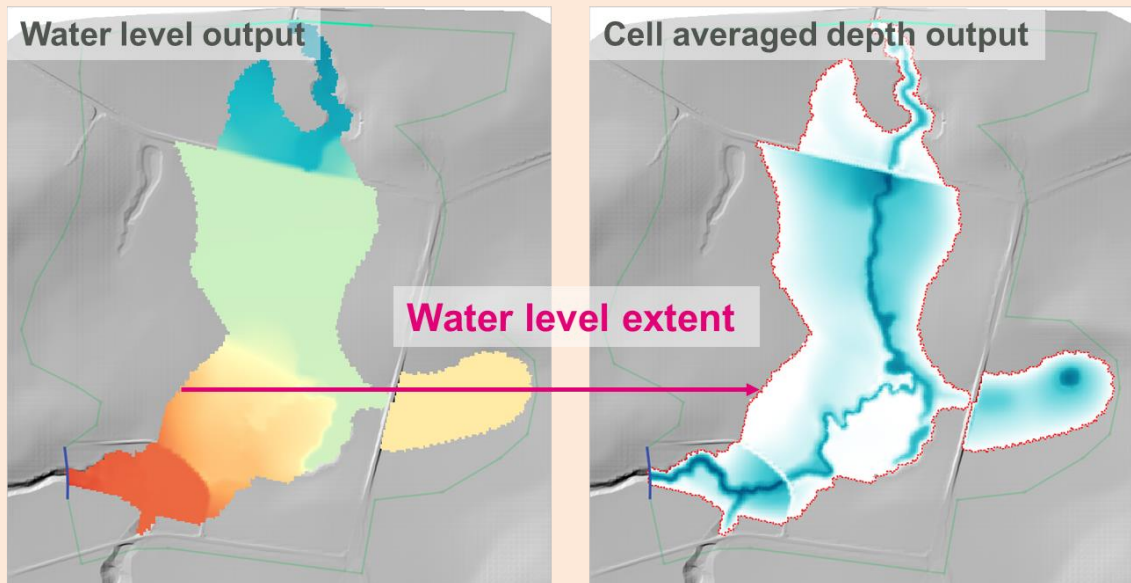


**Figure 4 Effect of Using SGS on a Catchment Wide Model**

Build 2020-10-AB implements a new depth output option (Section 7.5.3) to output the cell averaged depth (i.e. “volume / cell area”) for SGS models by specifying:

`SGS Depth Output == CELL AVERAGE`

As shown in Figure 5, the water level and depth extents are matched well using this option, and superior depth mapping will result, especially for direct rainfall models.



**Figure 5 Water Level and Cell Averaged Depth Output**

The corner interpolation for cell averaged depth is slightly different from when using other types of depth output options. The cell averaged depth at the cell corner is interpolated from the cell averaged depth from the surrounding cell centres, instead of subtracting the cell corner elevation from the

interpolated cell corner water level. In addition, ridge breaklines read by “`Read GIS Z Shape`” command are also considered in the corner depth interpolation at the cell corners selected by the breaklines. Note that the “`Read GIS Z Line`” command is not yet supported for this feature.

**Note:** Cell averaged depth output is not used for the new high resolution raster outputs (see Section 3.7), i.e. the high resolution depth output is still the difference of the interpolated water level and the SGS sampled bed/ground elevation.

### 7.5.7 SGS Velocity Based Output Depth

For output data types that use a combination of depth and velocity in the calculation, Build 2020-10-AB introduces a change in how the depth is determined where SGS is applied. The velocity used for these outputs should usually be the cell average velocity, therefore, for consistency a cell average depth is applied for these output types as of Build 2020-10-AB regardless of the “SGS Depth Output ==” setting (see Section 7.5.6 above). However, if desired, this setting can be changed with the new .tcf command:

```
SGS Velocity Based Outputs == {CELL AVERAGE DEPTH} | SGS DEPTH OUTPUT
```

If set to “SGS DEPTH OUTPUT” the depth used is that set by the “SGS Depth Output” command and is primarily provided for backward compatibility.

Output data types affected by this change:

- Unit flow
- Froude Number
- Bed Shear Stress and Stream Power
- Hazard Outputs

Where SGS is not applied there is no change. This modification was made after the Beta3 version of 2020-10-AB.

Please note that prior to Build 2020-10-AB, an SGS model always uses “water level – minimum elevation within SGS cells” to track the maximum hazard outputs, regardless of the output depth specified by the “`SGS Depth Output ==`” command (or “`SGS Depth Interpolation Approach ==`” prior to Build 2020-10-AB). This issue could cause the maximum hazard to become larger than that estimated from the maximum depth and velocity.

## 7.6 SGS – Remapping Model Outputs to a Finer Resolution DEM

**Note:** For Build 2020-10-AB, if using the new SGS Approach == Method C, it is possible to output high resolution rasters directly from TUFLOW. Refer to Section 3.7 for details.

The remapping of map output if using SGS to a fine resolution DEM is being built into TUFLOW as a direct output option. In the meantime, remapping can be carried out using the `asc_to_asc` utility using ASC and FLT formats with support for NetCDF to be built in for a future update. The workflow to carry this out is provided below and you will need to use Build 2020-05-AA or later of

asc\_to\_asc.exe available from:

<http://www.tuflow.com/Tuflow%20Utilities.aspx>.

The -remap option in asc\_to\_asc has been enhanced to facilitate the mapping of models using SGS to a finer resolution DEM, thereby considering the partially wet nature of cells due to SGS. The options for -remap and examples are provided further below.

**Note:** If using SGS it is generally not necessary to use the buffer distance (-bd) option (unless using the additional freeboard, -fb, option) as SGS models will, by default, automatically buffer the water level grid with the partially wet cells around the wet/dry interface. For this reason, a TIN (triangular interpolation) method is used by default as it is significantly faster to process and gives almost identical results to the IDW method.

**Note:** Whilst SGS provides high definition topographic representation within a cell resulting in excellent cell size convergence and no mesh orientation effects, it still only produces a single water level calculation at the mid-point of each 2D cell. In steep areas, or where there are sudden changes in topography the interpolation of water levels will still be subject to the coarseness of the 2D cell sizes. Should the remapping show discontinuities between surface water or unrealistic dry/wet areas, the only solution to improve upon this is to use a higher resolution hydraulic computation in these areas, usually by using the Quadtree option. The third set of example images further below highlight this situation for depicting flow over a road embankment.

**Note:** Additional functionality is presently being built into asc\_to\_asc to provide a workflow for remapping other types of map outputs to a fine resolution. This workflow will be added to this section of the release notes once available.

#### Options:

```
"-remap" remap a water level grid to a higher resolution DEM.  
  "-wl" sets the lower coarser resolution water level grid.  
  "-dem" sets the higher finer resolution DEM.  
  The default interpolation method is TIN (triangular interpolation).  
  "-idw_npt<number of points>" change the interpolation method to  
  IDW and set the number of points used for IDW interpolation  
  (default is 12).  
  "-fb<freeboard>" add an user specified freeboard to the original water  
  level grid.  
  "-bd<buffer distance>" set the maximum extent flags based on buffer  
  distance.  
  IDW interpolation is applied if "-bd" or "-fb" options are used.
```

And examples are:

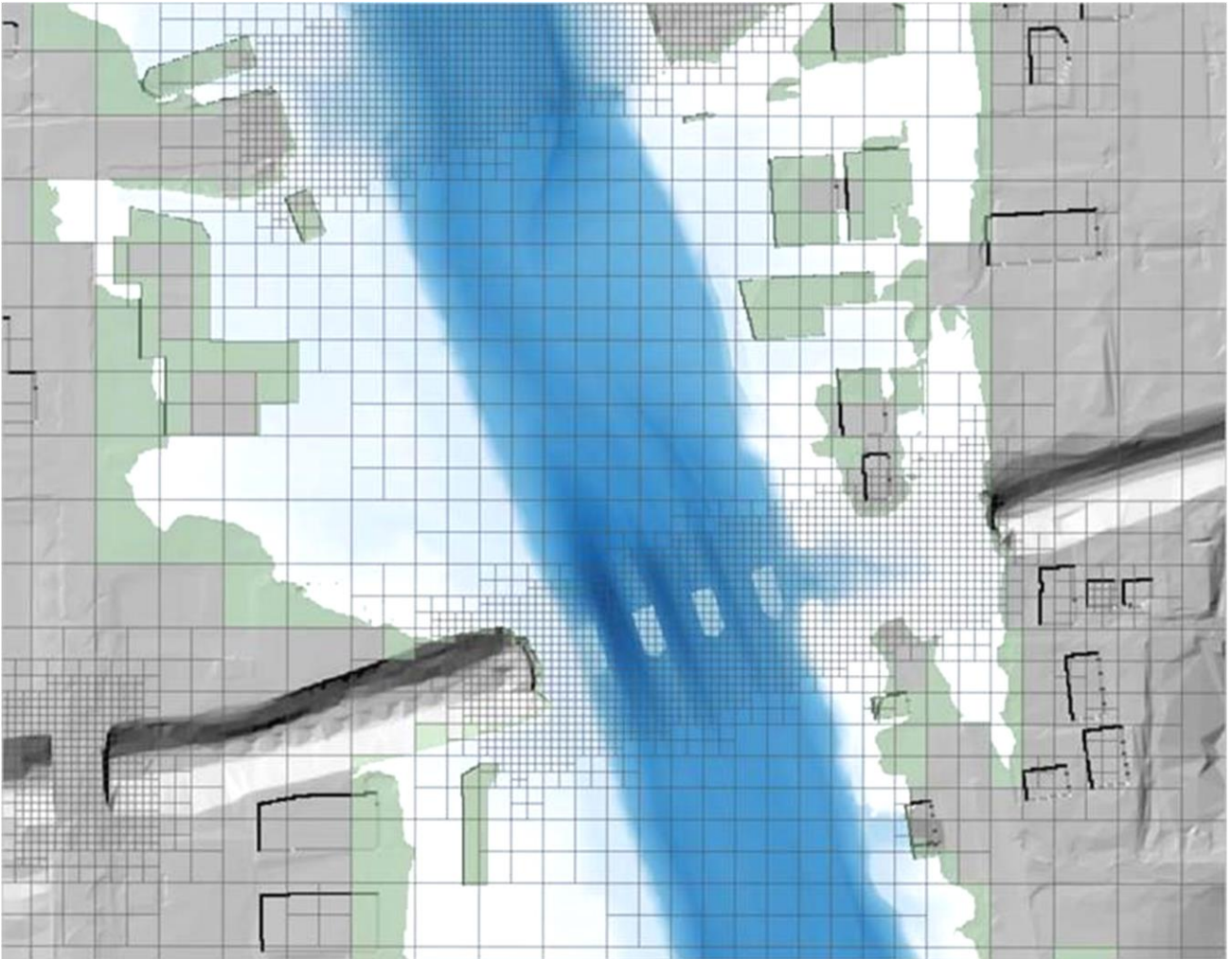
**Example 1** - Remaps a water level grid to a finer resolution DEM using TIN interpolation.

```
asc_to_asc.exe -remap -wl lowres_h.asc -dem DEM_highres.asc
```

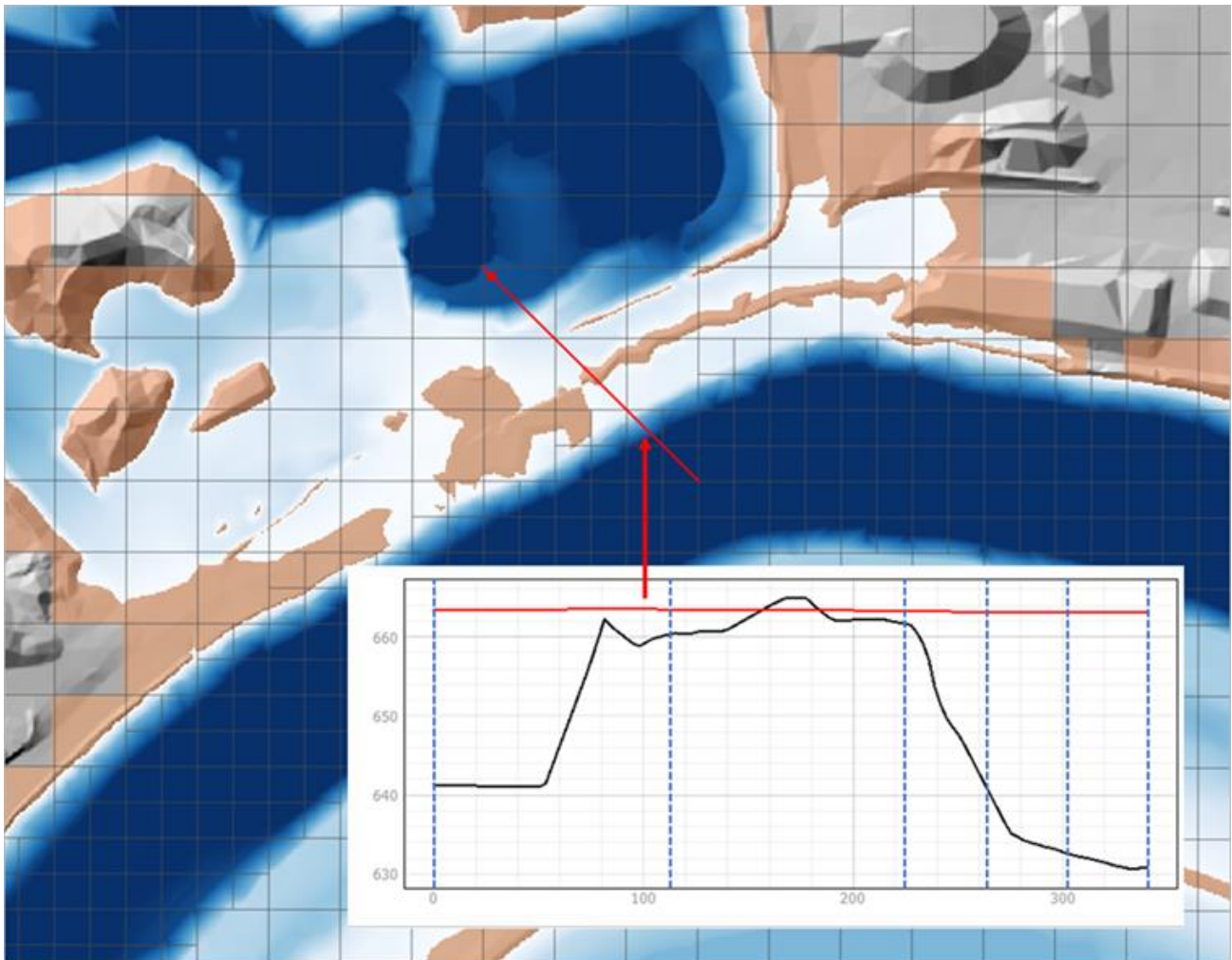
**Example 2** - Remaps a water level grid to a finer resolution DEM using IDW interpolation. Applies a buffer distance of 20m and adds a freeboard of 0.2m to the original water level grid.

```
asc_to_asc.exe -remap -idw_npt12 -fb0.2 -bd20 -wl lowres_h.asc -dem  
DEM_highres.asc
```



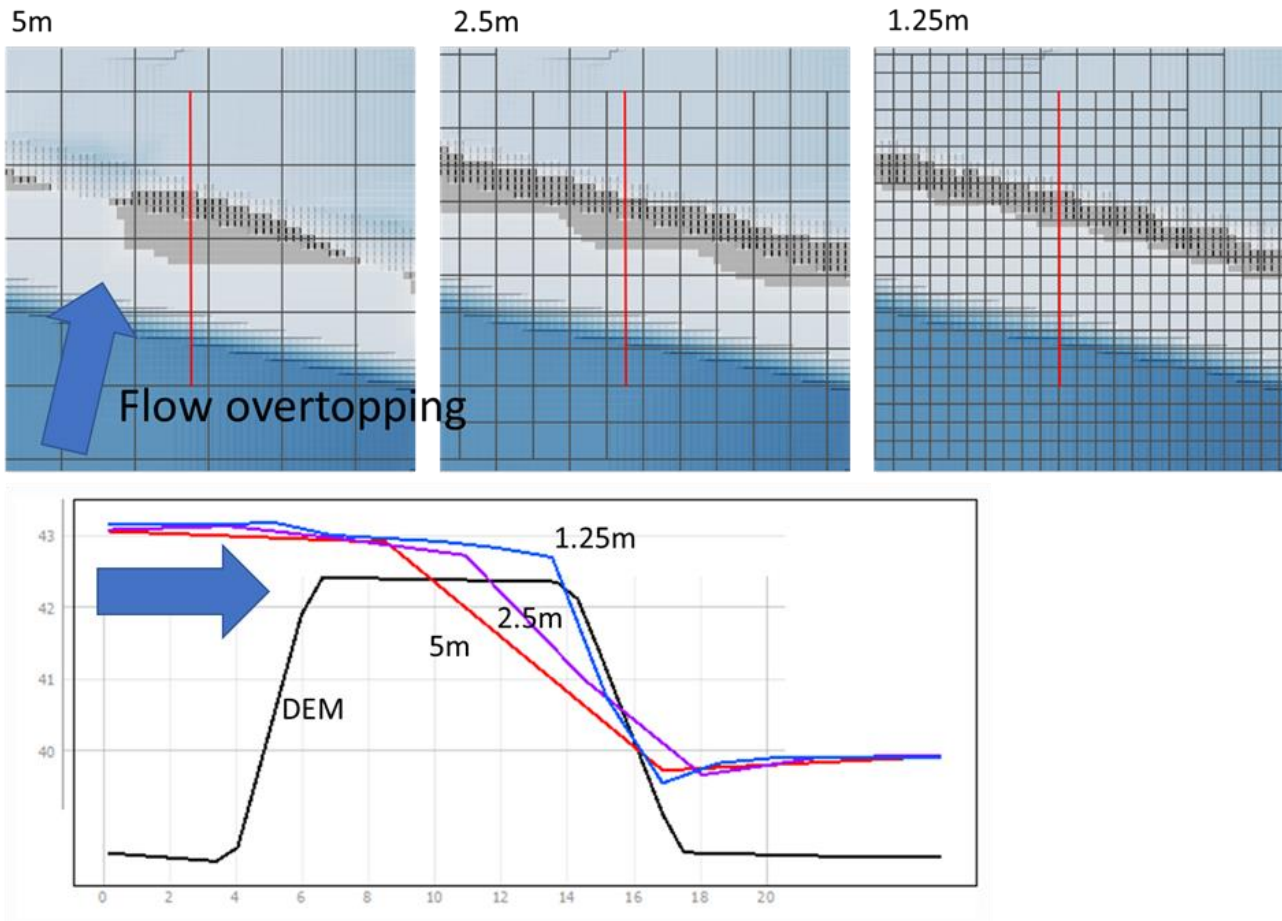


*Image above shows the water depths remapped from a Quadtree/SGS model using the water level grid output remapped against the fine resolution DEM. The green shaded cells show the partially wet SGS cells around the edge of the flood extent.*



*The images above show the water depths remapped from a Quadtree/SGS model and a profile of the water surface (red line) and DEM (black line) across the levee. Importantly with SGS being used the cells along the levee are partially wet (without SGS these cells are dry), thereby producing superior results by not blocking flow between river and floodplain, and the ability to produce high resolution mapping using the -remap option with no need to buffer the results.*





The top images above show the water depths remapped from a Quadtree/SGS model with varying cell size resolutions for the road embankment, whilst the lower image presents the water level profiles across the embankment for the different cell sizes. Whilst the different cell size scenarios produce near identical results upstream and downstream of the embankment, a much-improved water level profile is achieved across the embankment by using finer 2D cell resolutions resulting in improved remapping of results as shown in the top images. Note that though there is water flowing over the embankment, even the finest cell size of 1.25 m shows a thin strip of dry land on the downstream face of the embankment due to the interpolation of the computed water surface as can be seen in the profile.

## 7.7 Maximum Velocity Cutoff Depth now supported in HPC

For Build 2020-01-AB HPC (including Quadtree simulations) support the Maximum Velocity Cutoff Depth == functionality. This can be used to set a cutoff depth when tracking velocity maximums, below which the maximum velocity is the velocity at the peak water level, above this depth the maximum velocity is the tracked maximum velocity. This can be useful. To be consistent with TUFLOW classic, the default value is 0.1m.

If the “Defaults == Pre 2020-01” (or earlier) then this depth is set to 0.0 so that the maximum velocity is always the tracked maximum. To enable consistency with the 2020-01-AA release “Maximum Velocity Cutoff Depth == 0.” is required in the .tcf file.

This command is fully documented in the TUFLOW manual.

## 7.8 New Map Output Data Types

### 7.8.1 New Hazard Output – Port Macquarie-Hastings Council

Build 2020-01-AB introduces a new hazard category based on Table 3 in the Port Macquarie-Hastings Council Flood Policy (2018). This document is available via the following link:

<https://www.pmhc.nsw.gov.au/files/assets/public/aboutus/publications/registers/code-and-policy-register/policy-port-macquarie-hastings-council-flood-adopted-2018-12-12.pdf>

This is specified with the “ZPMH” map output data type. The categories are defined as:

$D < 0.4$  and  $V < 0.5$  and  $D*V < 0.2$ : Category 1 (Low Risk)

$D < 0.8$  and  $V < 2.0$  and  $D*V < 0.5$ : Category 2 (Medium Risk)

$D < 1.8$  and  $V < 2.0$  and  $D*V < 1.5$ : Category 3 (High Risk)

$V < 4.0$  and  $D*V < 2.5$ : Category 4 (Very High Risk)

Otherwise Category 5 (Extreme Risk)

Dry points are assigned Category 0.

### 7.8.2 New Hazard Output – Hamilton City Council NZ

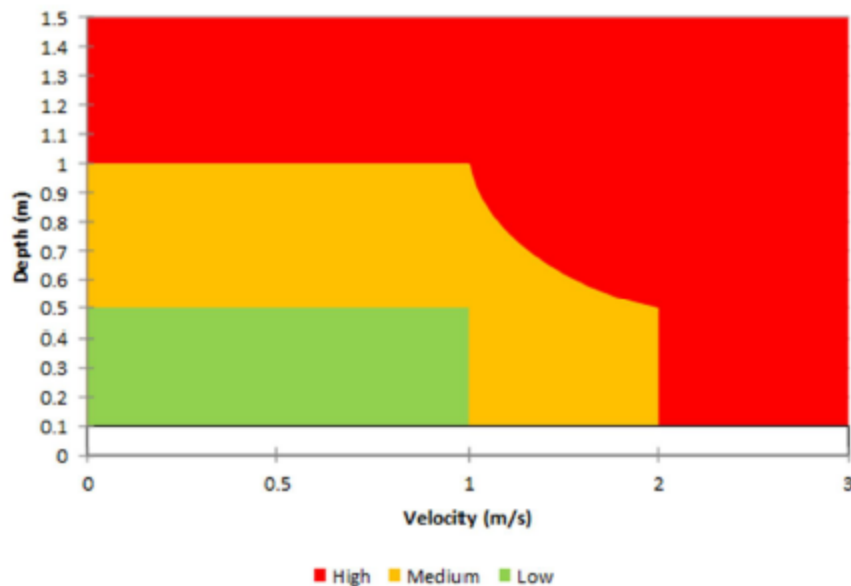
Build 2020-10-AB introduces a new hazard category based on the Hamilton City Council Flood Hazard Report (2012). This document is available via the following link:

<https://www.hamilton.govt.nz/our-council/council-publications/districtplans/flood/Documents/Flood%20Hazard%20Report%20-%2029%20Oct%202012%20-%20Final.pdf>

**Table 10: Flood Hazard Classification**

<b>Insignificant</b>	Applies when the depth of water is less than 100 mm, whatever the water velocity. Note that 100 mm of water in a property will cause damage, and even water of very shallow depth should be treated with caution.
<b>Low</b>	Applies when the depth is in the range of 100-500 mm and the velocity is less than 1 m/s.
<b>Medium</b>	Applies for three conditions: Depth is in the range of 500-1000 mm and the velocity is less than 1 m/s; or Depth is in the range of 100-500 mm and the velocity is in the range of 1-2 m/s; or Depth is in the range of 500-1000 mm and the velocity is in the range of 1-2 m/s and $\text{depth} \times \text{velocity} < 1$ .
<b>High</b>	Applies for three conditions: Depth is greater than 1000 mm; or Velocity is greater than 2 m/s; or Depth is in the range of 500-1000 mm and the velocity is in the range of 1-2 m/s and $\text{depth} \times \text{velocity} > 1$ .

**Figure 3 Depth – Velocity Criteria for Hazard Classification**



This is specified with the “ZNZ1” map output data type. The categories are defined as:

$D < 0.1$ : Category 1 (Insignificant)

$D < 0.5$  and  $V < 1.0$ : Category 2 (Low)

$D < 1.0$  and  $V < 2.0$  and  $D \times V < 1.0$ : Category 3 (Medium)

Otherwise Category 4 (High)

Dry points are assigned Category 0.

### 7.8.3 New User Defined Hazard Output

Build 2020-10-AB introduces a new ability for the user to define a number of depth, velocity and depth-velocity product thresholds to create custom hazard categories. Like other hazards these are assessed at each computational timestep and the maximum is tracked every timestep. To specify a user defined hazard, add output type “ZUD1” in the “Map Output Data Types” command, e.g.:

```
Map Output Data Types == h d V ZUD1
```

The thresholds are defined through reading a .csv file using the following command:

```
Read Hazard File == <file_name>.csv
```

The csv file should contain three (3) columns defining the thresholds for depth, velocity, and depth-velocity product respectively. The example below produces the same output as for “ZNZ1” introduced above.

	A	B	C
1	D	V	D*V
2	0.1	99999	99999
3	0.5	1	99999
4	1	2	1

Category 1:  $D < 0.1$ . Note: the velocity, and depth-velocity product thresholds are effectively not applied by using a very large threshold value of 99999.

Category 2:  $D < 0.5$  and  $V < 1.0$ . Note: the depth-velocity product threshold is effectively not applied by using a very large threshold value of 99999.

Category 3:  $D < 1.0$  and  $V < 2.0$  and  $D*V < 1.0$

Category 4: Otherwise

Dry points are assigned Category 0.

A maximum of 10 categories can be specified by the .csv file, but just one User Defined Hazard is allowed per simulation. This functionality is compatible with both the Classic and HPC solvers.

## **7.9 1D External Scheme WLL Temporal Map Output Bug for HPC Linked Models**

### **7.9.1 Build 2020-01-AB**

Build 2020-01-AB includes a WARNING 2465 that the temporal WLL map output for TUFLOW HPC linked models is suppressed due to a bug. Note that:

- The bug does not affect the maximums WLL map output from HPC linked models.
- TUFLOW Classic linked models are unaffected, only TUFLOW HPC models.
- The bug has no influence on TUFLOW 1D (ESTRY) WLL map output.

### **7.9.2 Build 2020-10-AA Onwards**

The bug described above has been fixed in the 2020-10-AA Build. As there are changes also required to the code base in the external 1D scheme, Build 2020-10-AA will need to utilise a compatible version of external 1D engine.

## **7.10 Output Changes for 2020-10-AA**

### **7.10.1 Reporting Location Supported for HPC including Quadtree**

For Build 2020-10-AA, Reporting Locations are now supported for the HPC solver, including Quadtree meshes.

### **7.10.2 Structure Groups Supported for HPC including Quadtree**

For Build 2020-10-AA, Structure Group outputs are now supported for the HPC solver, including Quadtree meshes.

Build 2020-10-AC fixes an output bug in the structure group reporting for HPC that could cause the structure group's upstream and downstream levels to only report zeros. This would also affect the \_SHmx.csv output when it occurred.

### **7.10.3 Evacuation Routes Now Supported for HPC including Quadtree**

Build 2020-10-AA now supports Evacuation Route outputs for the HPC solver, including Quadtree meshes.

Build 2020-10-AC fixes a bug in HPC where if using cut-off depths of zero this would trigger the evacuation route as being cut from the first timestep, not from when the route was overtopped. Build 2020-10-AC also supports Evacuation Routes in SGS, but the SGS cells/faces selected by Evacuation Routes layers are assumed as flat (i.e. the single ZU or ZV elevation is used as per a non-SGS model).

2020-10-AD fixes a rare memory allocation issue for Evacuation Routes for models with SGS activated during the model pre-processing.

### **7.10.4 GIS Workspaces for QGIS and MapInfo Support Relative Filepaths**

For Build 2020-10-AA workspaces output for QGIS (\*.qgs) and Mapinfo (\*.wor) are now written with relative references. This can be switched back to absolute references with the following command in the TCF.

GIS Project Path Format == Absolute | {Relative}

This command is set to absolute if running with the Defaults == Pre 2020-10, refer to Section 12 for further details.

Build 2020-10-AB sets the default GIS Project Path Format to 'Relative'. Previously this was incorrectly set to 'Absolute' in the 2020-10-AA release.

#### 7.10.5 12D TMO Map Output Format Supported for Quadtree Simulations

For Build 2020-10-AA, when running Quadtree, map output in the 12D TMO format is now supported. This will require 12D Model version 15 or higher.

#### 7.10.6 12D TGO Velocity Angle Map Output in Radians

Build 2020-10-AA now outputs the TGO velocity angle in radians. Previously this was output in degrees. This change applies to both Classic and HPC.

#### 7.10.7 New Plot Output (PO) Data Types

Build 2020-10-AA now supports the following PO data types and additions.

Flag	Description	
Point, Line, or Polyline		
D_	Depth	<p><b>Point:</b> Depth of the nearest cell</p> <p><b>Line or Polyline:</b> The average depth of all wet cells along the line. Note: In Classic if a polyline is used, the average water level along each line segment is output, therefore, use of polylines is not recommended for this output type at present. HPC, including Quadtree, will take the average depth along entire polyline.</p>
H_	Water Level (Head)	<p><b>2020-10-AA now supports line objects in HPC including Quadtree (previously in HPC this would return the water level at the first vertex along the line)</b></p> <p><b>Line or Polyline:</b> The average water level of all wet cells along the line. If all cells are dry, the lowest cell's ground level (ZC) is output. Note: For polylines, unlike Classic, HPC will average water level along entire polyline.</p>
V_	Velocity	<p><b>2020-10-AA now supports line objects in Classic and HPC including Quadtree</b></p> <p><b>Line or Polyline:</b> The average velocity along the line. The average velocity is calculated as Q_ / QA.</p>



Flag	Description	
HU HD	Downstream and Upstream Structure Water Levels	<p><b>2020-10-AA now supports HU and HD types in HPC including Quadtree</b></p> <p><b>Point, Line or Polyline:</b> Used to define the upstream (HU) and downstream (HD) water levels of a QS structure. The 2D water level is used to populate the upstream and downstream water level data in the structure output feature. For a point the water level at the nearest 2D cell centre is used. For a line or polyline, the average water level along the entire line is calculated. Note, for polylines the water level being the average along the entire line.</p> <p>To associate the HU and HD objects with the QS line, all three (QS, HU and HD) must have the same ID for the 2d_po Label attribute Label. Note that the water levels over time are output to a 2D PO .csv file and the summary information at the flood peak to the _SHmx.csv file. If the point or line is completely dry, -99999 is output to the .csv files.</p> <p><b>2020-10-AA now supports writing out _SHmx.csv file for 2D only models.</b></p>
QS	Structure Flow	<p><b>2020-10-AA now supports QS type in HPC including Quadtree</b></p> <p><b>Point:</b> N/A (zero flow results).</p> <p><b>Line or Polyline:</b> Same as Q_, but also used to set up a 2D structure output that will include in addition to the 2D flow any flows from intersected 1D structures and the split between below and above deck flows. Note the flow output to the 2D PO .csv files is only the 2D flow, while that to the _SQ.csv file is the combined 1D/2D structure flow.</p> <p><b>2020-10-AA now supports writing out _SHmx.csv file for 2D only models.</b></p>

#### 7.10.8 Additional Fields in the \_ccA GIS Output File

Build 2020-10-AA includes additional reporting fields in the 1D \_ccA GIS output. The additional fields pertain to any surcharge occurring at closed channels e.g. underground pipes. A pipe is considered surcharging if it is connected to a surcharging pit (a pit experiencing negative flow) and the depth of water above the upstream pit invert is greater than a user defined value (default is 0 m). The user defined cut-off depth is a global value.

```
Pit Surcharge ccA Cutoff Depth == {0} | <Value>
```

Additional fields:

- Sur\_CD = Surcharge cut-off depth. Depth above pit upstream invert for a pit to be considered surcharging.
- Dur\_Sur = Duration of surcharge. Total duration (hrs) that the pipe is considered surcharging.
- pTime\_Sur = Percent time that the pipe is surcharging relative to the time the pipe is running at 100% full

- TFirst\_Sur = Time of first surcharge (hrs).

#### **7.10.9 Improved High Control Number Reporting**

Build 2020-10-AA now reports the location of High Control Numbers in the messages GIS layer. Prior to 2020-10-AA only locations of Not A Number (NaN) instabilities was reported. From Build 2020-10-AA onwards, both NaN and High Control Number locations are reported with WARNING 2550. For models with repeat timesteps due to High Control Numbers, the messages layer can be used to locate where these are occurring.

### **7.11 Bug fixes and Minor Enhancements for 2020-10-AA**

#### **7.11.1 Bug Fix for 2d\_po Lines Outside of Quadtree Mesh**

Build 2020-10-AA fixes an issue with Plot Output lines when they were located completely outside of a quadtree mesh. For 2020-01 versions incorrect data was written to the output files. This typically would include invalid numbers (e.g \*\*\*\*) and / or dummy and incomplete entries in the \_PO.csv output.

#### **7.11.2 Bug Fix for 2d\_po Regions Havg Type in Quadtree**

Build 2020-10-AA fixes an issue with the Havg PO type for regions that would not correctly average the water level when the polygon intersected mesh refinement areas in a Quadtree mesh.

#### **7.11.3 Bug Fix for 1d\_chan.csv**

Build 2020-10-AA fixes an issue in the 1d\_chan.csv written to the plot/csv folder that would incorrectly output open channel left and right bank elevations.

#### **7.11.4 Bug Fix in Check File if using Read CnM Mannings**

Build 2020-10-AA fixes two check file issues when specifying bed roughness directly in the .tgc file via CnM (Set CnM, Read GIS CnM or Read Grid CnM) rather than using a materials file and Mat commands (Set CnM, Read GIS CnM or Read Grid CnM). The two fixes are:

- The DEM\_M (raster output of Material ID) is suppressed as it is not valid. This applies to Classic, HPC and Quadtree simulations.
- For Quadtree the \_uvpt check file are now correctly reported, previously these were being output as Not a Number (NaN). This does not affect Classic or HPC check files.

This does not affect the computations, the roughness values being used were correct, only the check files have been changed.

#### **7.11.5 Increase Maximum Length of ID field in \_TS GIS output layers**

Build 2020-10-AA allows extra width for the timeseries name attribute in the \_TS GIS output layer this has been increased to 34 characters, which allows for the 32 character name in plot output and reporting locations plus 2 characters for the output data type. Prior to the 2020-10-AA release this was limited to 14 characters based on the 12 character 1D channel name plus 2 characters for the output data type. For Builds prior to 2020-10-AA the name could be truncated in the \_TS GIS output layer – but not in the output .csv files.

Build 2020-10-AB fixes alignment issues in the fixed field .eof (ESTRY Output File) associated with this change.

#### **7.11.6 Extra Precision in \_grd Check Files for SRF Value**

Build 2020-10-AA allows extra width for Storage Reduction Values in the grid check file (\_grd\_check). Prior to this, values of less than -9.999 would be written as \*\*\* in the check file, causing it to show up as blank or NaN when viewed in GIS. This change only affects the values written in the check file and not the SRF values used in the engine.

#### **7.11.7 Extra Precision in \_grd Check Files for Z\_Map\_Cutoff in Quadtree (SGS only)**

Build 2020-10-AA allows extra width for sgs \_Cutoff values in the grid check file written when running with Quadtree (\_grd\_check). This field is only output if SGS is enabled. Prior to this, values of greater than 99.99 would be written as \*\*\*\* in the check file, causing it to show up as blank or NaN when viewed in GIS. This change only affects the values written in the check file and not the values used in the engine.

#### **7.11.8 Extra Precision in \_lfcsh\_uvpt for Form Loss Values**

Build 2020-10-AA allows extra width for form loss values in the lfcsh\_uvpt check file, this has increased from three to four decimal places. The form loss values are stored as floating point numbers, this change only affects the value written to the check files.

#### **7.11.9 Fix an Issue with Multiple Evacuation Routes with the Same Name**

Build 2020-10-AA fixes an issue with the evacuation routes. When multiple GIS inputs had the same name this was causing "ERROR 2409 - Cut off types must also match when extending route with matching name" to be spuriously output. If this issue occurred the simulation was halted.

#### **7.11.10 Bug Fix for SAC Check File**

Build 2020-10-AA fixes an issue with 2d\_sac MIF check files when SA pits and normal SA polygons were used in the same model. Previously in Classic and HPC the 2d\_sac could be incomplete, in Quadtree the user may get the following message:

```
Should not be here [gis0_%ir(i) <= gis0_%ir(i-1)] - please contact support@tuflow.com.
```

#### **7.11.11 Fix an Issue with Holes in Raster Outputs**

Build 2020-10-AA fixes an issue which could cause holes or gaps in raster outputs if the model had a very slight rotation (e.g 0.1 degrees or 44.9 degrees). Due to a precision issue, when an output raster pixel was on top of the cell edge, it could be missed.

This issue was more likely to occur in models with very fine cell size / output grid size.

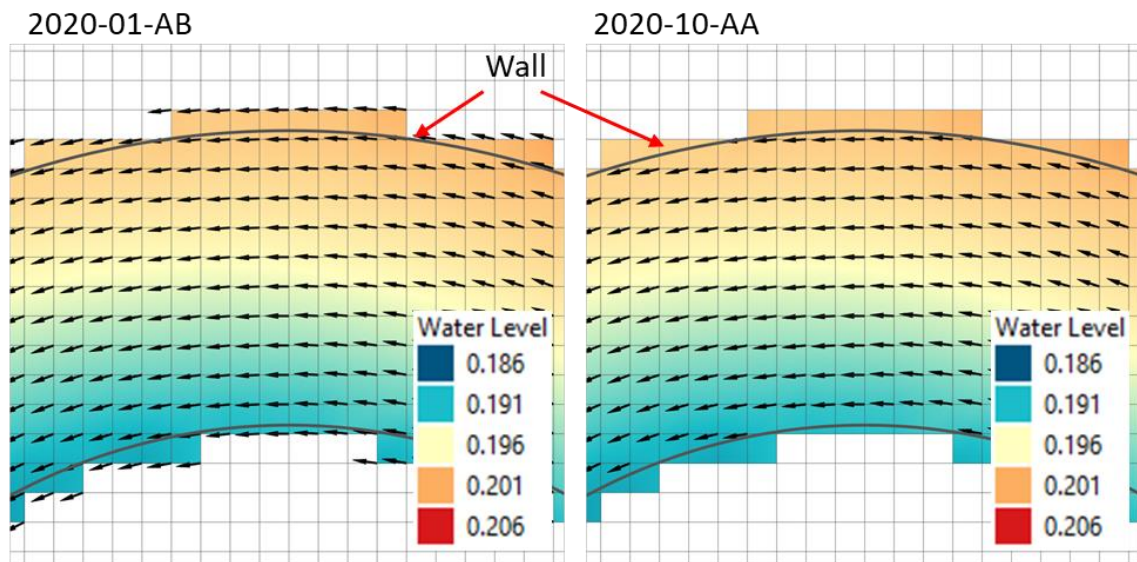
#### **7.11.12 Vector Output at Dry Zpts**

For mesh-based output files (XMDF, DAT), vector output is normally displayed at a mesh vertex if one of the surrounding mesh elements is wet. If using SGS, this will display vectors at nodes in partially wet cells that are dry. As illustrated in the figure below, velocity vectors are displayed at

mesh vertices outside the vertical wall despite those cell corners being dry. Build 2020-10-AA allows the user to control whether the vectors are outputted at dry Zpts or not.

`Map Output Vector Dry Zpts == ON | {OFF}`

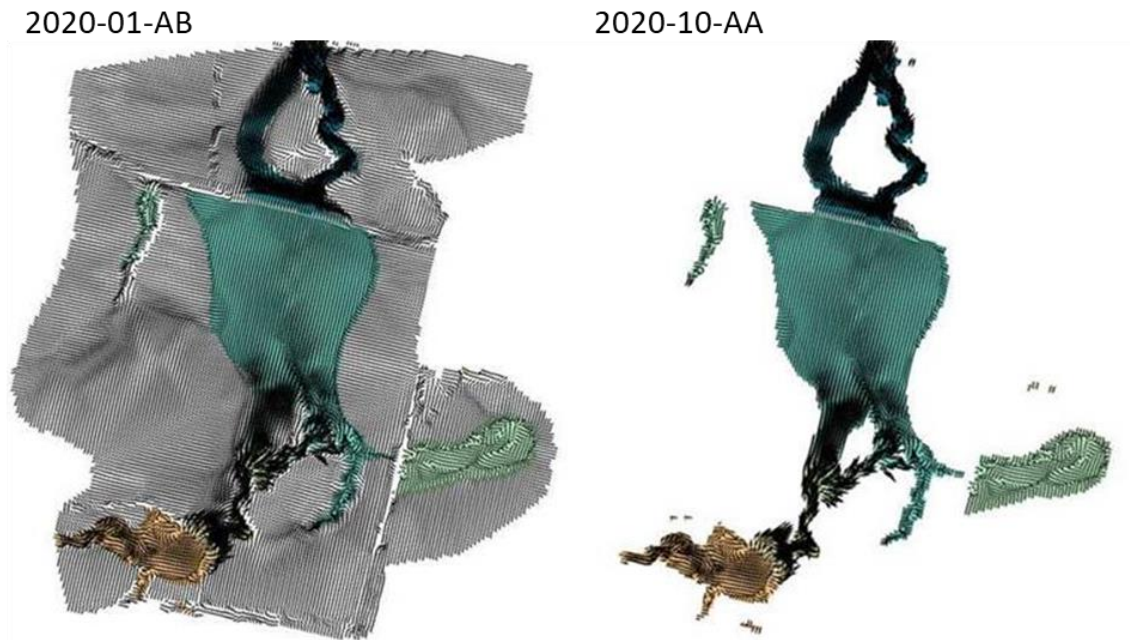
The “OFF” option is the default, which will disable the vector outputs at dry Zpts. “ON” option can be used for backward compatibility.



Options for vector output at dry Zpts

### 7.11.13 Map Cutoff Depth and Vector Output

When using “Map Cutoff Depth” with mesh-based output files (XMDF, DAT), the display of scalar and vector outputs at each mesh is controlled by a 0/1 flag, which is set as 0 if the local depth is below the cutoff depth. This flag disables the vector display in SMS however does not in the current QGIS release (version 3.14 at time of publishing), and thus the velocity vectors are display everywhere for a global rainfall model even if the “Map Cutoff Depth” command is specified, as illustrated below.



Options for vector output cutoff

Build 2020-10-AA includes a new command below to set the vector output values as zero in the mesh output files at 2D cell corners and centres that are below the “Map Cutoff Depth”.

```
Map Cutoff Vector == {ON} | OFF
```

The “ON” option, the default for Build 2020-10-AA, enables the map cutoff depth and sets the vector output value as zero if the cell water depth is below the cutoff depth. “OFF” option can be used for backward compatibility.

## 7.12 Output Changes for 2020-10-AB

### 7.12.1 Output Zones are Now Supported for Quadtree Meshes

For Build 2020-10-AB output zones are now supported for Quadtree meshes for all map output data types that are supported by Quadtree (at present, this excludes Cell Centred NetCDF, Blue Kenue and WaterRIDE outputs). Refer to the TUFLOW manual for more information on output zones.

### 7.12.2 Time Output Cutoff Hazard Support in HPC

Build 2020-10-AB has introduced support for “Time Output Cutoff” to hazard categories for the HPC solver, as per TUFLOW Classic. Previously for HPC (including Quadtree) depth was the only criteria



available. This has been extended to all hazard categories, refer to “Time Output Cutoff” command in the TUFLOW manual for further details.

### 7.12.3 TMO Update for Quadtree

The TMO map output used by 12D has been updated to v3.0 for Quadtree results in Build 2020-10-AB. The output has been updated to include results at all 9 cell points (cell centre, cell sides, cell corners). Results at cell corners are interpolated from surrounding cells, similar to .xmdf and .dat results. Results at cell sides use a mixed interpolation from cell corners at refinement and wet-dry boundaries, and from cell centre results in all other cases.

HPC (non-Quadtree) and Classic will continue to use TMO output v2 which contains only cell centre results. This change was made after the Beta3 version of 2020-10-AB.

Minor fix in 2020-10-AC, that now correctly outputs the result version number in the .tmo header for results from Classic / HPC (not Quadtree) that was broken in the 2020-10-AB build.

## 7.13 Bug fixes and Minor Changes for Build 2020-10-AB

### 7.13.1 HPC PO Online Causing Zero Values in Results

Build 2020-10-AB fixes a bug that could cause 1D outputs to show zero values when write PO Online was on in HPC. This bug only affects the output writing, and the computations were not affected.

**Note:** PO Online is not currently supported for Quadtree Solver, if specified WARNING 2815 is issued.

### 7.13.2 \_TS.mif not Loading in Mapinfo

Build 2020-10-AB fixes a bug that was introduced in 2020-10-AA that could incorrectly set styling for the \_TS.mif output for point objects. This would cause the layer to produce an error when imported into Mapinfo.

### 7.13.3 Q\_ / QS PO Point Output Causing “Should not be here” Error

Build 2020-10-AB fixes an issue that would cause a ‘Should not be here’ if the user specified a Q\_ or QS point object for a HPC simulation. Point objects do not support flow outputs and a warning is now produced and the output type will be ignored, and the simulation will continue.

### 7.13.4 D\_ PO Point Output in HPC Causing “Should not be here” Error

Build 2020-10-AB fixes an issue that could cause a ‘Should not be here’ if the user specified a D\_ point output while using the HPC solver.

### 7.13.5 PO Character Limit Causing “Should not be here” Error

Build 2020-10-AB fixes an issue that would cause a “Should not be here” error when the user ran into the character limit in the PO GIS layer that caused incomplete PO type names for certain line PO types. For example, if a ‘H\_’ PO type name was cut short due to the character limit for a PO line and only the ‘H’ was recorded.



#### 7.13.6 Bug Fix for QS PO Outputs in Models with no 1D Elements

Build 2020-10-AB fixes an issue with HPC simulations which had QS (structure flow) PO types but no 1D elements anywhere in the model. This could occasionally cause a simulation to crash. In these circumstances, the \_SQ.csv output (the total structure flow for both 1D and 2D) will not be written. The \_2D\_QS.csv (the 2D structure flow) output would still be written.

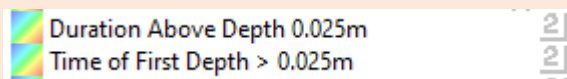
#### 7.13.7 Increase Decimal Places for Time Output Cutoff Depths

Build 2020-10-AB increases the number of significant figures for the labels of time output cutoff depths output labels. Previously the output label was restricted to two decimal places, this has now been increased to three.

For example:

`Time Output Cutoff Depths == 0.025`

will now correctly show the 3 significant figures, for example, a legend like below.



#### 7.13.8 “Should not be here” While Writing 1D Check Files

Build 2020-10-AB fixes an issue that would cause a “Should not be here [dbfdataline]” error while writing 1D check files if the user had included quotes in the Type attribute field in a 1d\_nwk layer (e.g. “W” instead of just W).

#### 7.13.9 Bug Fix for dGW Output in HPC Solver

Build 2020-10-AB fixes an issue with the Depth to Ground Water (dGW) map output for HPC fixed grid simulations. This was showing the infiltrated water causing the ground water level to rise too quickly. This did not affect the hydraulic calculations of infiltration and groundwater level; this issue was with the dGW map output only. Classic and HPC Quadtree simulations were not impacted.

#### 7.13.10 Bug Fix for HX Flags in the 1d\_to\_2d Check File

Build 2020-10-AB fixes an issue with the “S” flag being output to the 1d\_to\_2d check file for HX cells. This does not affect the calculations only the check file. This flag is only used if the “Adjust Head at ESTRY Interface ==” is enabled, which is not the default and is no longer recommended.

#### 7.13.11 \_zpt\_check File Update for SGS Models

For Build 2020-10-AB when writing the \_zpt check file with SGS enabled the fourth attribute is output with the name “Zmin” rather than elevation, as it refers to the lowest elevation.

#### 7.13.12 Create TIN Zpts not Writing to sh\_obj\_R Check File

Build 2020-10-AB fixes an issue that would not write to the sh\_obj\_R check file for the ‘Create TIN Zpts’ command if the command did not include ‘Write TIN’ (i.e. ‘Create TIN Zpts Write TIN ==’) and ‘XF Files == OFF’ was specified in the .tcf file.

#### **7.13.13 sh\_obj\_R Check File Not Writing when using XF Files**

Build 2020-10-AB will now write to the sh\_obj\_R check file when reading a previously generated TIN from xf files from mands “Create TIN Zpts ==” or “Read GIS Z Shape ==”. Previously TUFLOW would only write to the sh\_obj\_R check file while generating the TIN.

#### **7.13.14 Quadtree Orientation Angle output in the .tlf File Now in Degrees**

For Build 2020-10-AB the Orientation Angle output in the .tlf when running Quadtree mesh is now in degrees which is consistent with HPC / Classic fixed grid simulations. Before this version, the angle was output in radians when running with a Quadtree mesh.

#### **7.13.15 zsh\_zpt Check Points at Deactivated Quadtree Cell Corners**

Build 2020-10-AB removes the zsh\_zpt check points written at deactivated Quadtree cell corners.

#### **7.13.16 DEM\_Z Check File and SXL/SXZ Flag**

Build 2020-10-AB fixes an issue that the elevation update performed by an SXL/SXZ flag were not reflected in the DEM\_Z check file. This only impacts the model output, not the hydraulic calculations.

#### **7.13.17 Water “Puddles” in Dry Quadtree Cell**

Build 2020-10-AB fixes an issue that zero (0) depth water “puddles” were written in the map output before the Quadtree cells became wet. This is an output issue, it did not impact the actual water volume in a cell, nor affect the hydraulic calculations.

#### **7.13.18 Number of Irregular Culvert Barrels in hydprop\_check File**

Build 2020-10-AB fixes an issue in the hydprop\_check file that would only report the flow area and conveyance for a single barrel for ‘I’ (irregular) type pipes regardless of the input ‘number of’ attribute. Note, the correct area was being used in the calculations, this was only an issue with the hydprop\_check file.

#### **7.13.19 Extra precision in .hpc.dt.csv**

Build 2020-10-AB changes the precision in the .hpc.dt.csv file for the “tEnd” output column, which contains the simulation time in seconds at the end of the timestep. Previously, as the number got larger, then the number of significant figures after the decimal was reduced and for long simulations this could be rounded to the nearest second. For 2020-10-AB, this is written with a consistent number of significant figures (3).

#### **7.13.20 Bug Fix for PO Region Outputs**

Build 2020-10-AB fixes a bug that “Flow Out of Region” and “Volume” were not written to the time-series output. This change was incorporated into the Beta4 version.

#### **7.13.21 Bug Fix for Model with Record Gauge Data**

Build 2020-10-AB fixes an issue where the simulation exits without any error message when “`Read GIS Object RECORD GAUGE DATA`” function is used.

#### **7.13.22 Advanced Weir “number of” attribute in Check File**

Build 2020-10-AB correctly reports number of channels in the nwk\_C\_check output as one for advanced weir types (WW, WB, WC etc.). Previously this was being reported incorrectly, however this was a check file bug only and did not have any bearing on the computations.

### **7.14 Bug Fixes and Minor Changes for Build 2020-10-AC**

#### **7.14.1 Bug Fix for PO Output in Quadtree Model**

2020-10-AC fixes an issue that some Quadtree models with both PO points and lines could cause a memory access violation depending on the combination of output types.

#### **7.14.2 Bug Fix for HR Output in double precision**

2020-10-AC fixes an issue where double precision models with HR (high resolution) output crash during the pre-processing due to an issue with memory allocation.

#### **7.14.3 Bug Fix for HR Output in Output Zone**

2020-10-AC fixes an issue where models with HR (high resolution) output crash when HR output is used in output zones, but HR output is turned off for the entire domain.

#### **7.14.4 PO Header Change for Quadtree**

2020-10-AC changes the output header order for Quadtree outputs to be the same as Classic / HPC fixed grid PO outputs. Previously the first two header lines in the ‘\_PO.csv’ were incorrectly switched for Quadtree results. This only affected the header in the ‘\_PO.csv’ and not any of the csv outputs in the **plot/csv** folder.

### **7.15 Bug Fixes and Minor Changes for Build 2020-10-AD**

#### **7.15.1 Bug Fix for PO Output in HPC**

2020-10-AD fixes an issue for PO line types in HPC where if a line segment only selected a single cell a ‘Should not be here’ message was produced.

### **7.16 Bug Fixes and Minor Changes for Build 2020-10-AE**

#### **7.16.1 PO Regions in HPC causing a should not be here**

2020-10-AE fixes an issue that could occur in HPC with PO region objects causing a ‘Should not be here’ to occur. This issue only occurred in HPC single domain models (i.e. not Quadtree) and was caused by a negative memory assignment causing an integer overflow. This could occur when the total number of cells selected by all the PO regions was significant. For example, if the model had numerous small PO regions that selected many cells, or a few large PO regions that selected many cells.

#### **7.16.2 QS Flow Output Missing**

2020-10-AE fixes a bug that was introduced in the 2020-10-AD build that could cause structure flow outputs (‘QS’ PO type) to not be written to either the PO.csv or in the ‘plot/csv’ output folder.

#### **7.16.3 Quadtree dt min Output Not Writing**

2020-10-AE fixes a bug in the dt min output in Quadtree that was not writing when there was no temporal output, and not writing the final values correctly if the simulation didn't stop on a map output interval.

#### **7.16.4 Incorrect Maximum Value Written for the dt min XMDF Output**

2020-10-AE fixes a bug in the XMDF output for HPC and Quadtree for the dt min type. Previously the maximum value from a dataset could be incorrectly written to the XMDF which caused result viewers to incorrectly display default min/max values.

#### **7.16.5 Quadtree Maximums Showing Dry Cells as Wet**

2020-10-AE fixes a bug in Quadtree that could sometimes flag a cell as wet due to approximations in floating point binary operations when calculating depth on the post-processing side. This could manifest in null values being reported to outputs because actual values weren't available (e.g. dt min grid output would contain cells with values up to  $9 \times 10^{39}$ ).

#### **7.16.6 Quadtree Model with Variable Z Shape**

Build 2020-10-AE fixes an issue with variable z shapes in Quadtree with map outputs. If there were any zpts selected by variable z shapes where the final elevation was identical to the initial elevation, then Not a Number (NaN) values could occur in the output. This did not affect the hydraulic computations, just the map outputs in these locations.

#### **7.16.7 HPC ZH Output**

Build 2020-10-AE fixes an issue with HPC (not Quadtree) ZH map output type. The elevations reported to the output would toggle between negative and positive values on subsequent map output writes along the left edge of the 2D domain. This would only affect models with active cells in the first two columns of the domain.

#### **7.16.8 Pit check file Width attribute**

Build 2020-10-AE fixes an issue with the "Width" attribute reported in the \_pit\_A check file. Previously it was reporting the "conn\_width" attribute.

#### **7.16.9 Empty File Attribute Changes for 2d\_lfcsh**

Build 2020-10-AE changes the attribute names for the 2d\_lfcsh empty file to be consistent with the release notes. Attribute 10 has changes to "L2\_or\_L23\_FLC" and attribute 13 has changed to "L3\_FLC\_or\_fIP".

## 8 Simulation Control

### 8.1 Default Simulation Log Folder Changed

The simulation log folder has changed to "C:\ProgramData\TUFLOW\log". Previously, "C:\BMT\_WBM\log\\_ All TUFLOW Simulations.log" was used, however, this can cause issues if access to C: drive is blocked by your IT administrator. Writing to the simulation log folder can still be suppressed with the .tcf command "Simulation Log Folder == Do Not Use".

#### 8.1.1 Include Username in Simulation Log Folder

A number of users reported permissions issues with TUFLOW 2020-01-AA when using the default simulations log folder "C:\ProgramData\TUFLOW\log\\_ All TUFLOW Simulations.log". The issue generally occurred if multiple users were running simulations on the same computer, the first user created the file and then subsequent users may not have had permissions to write to the file.

For the 2020-01-AB the username is added to the default simulations log folder "C:\ProgramData\TUFLOW\<username>\log\\_ All TUFLOW Simulations.log". This folder location can be set with the "Simulation Log Folder == " command in the .tcf or the licence control file (see point below).

#### 8.1.2 Extra Licence Control File Location

Build 2020-01-AB allows the user to specify the licence control file in "C:\ProgramData\TUFLOW\TUFLOW\_Licence\_Settings.lcf". The search locations in decreasing order of priority are:

1. TUFLOW\_Licence\_Settings.lcf in the same location as the TUFLOW executable
2. C:\ProgramData\TUFLOW\TUFLOW\_Licence\_Settings.lcf
3. C:\BMT\_WBM\TUFLOW\_Licence\_Settings.lcf
4. C:\BMT\_WBM\TUFLOW\_Dongle\_Settings.dcf

#### 8.1.3 Bug Fix for Simulation Log Folder

Build 2020-01-AB fixes the Simulation Log Folder reporting in the .tlf.

### 8.2 Package Model (-pm) Enhancements

The package model (-pm) that rapidly packages up multiple model configurations has been enhanced to (hopefully) now capture all combinations and permutations of model inputs. If the -pm feature misses a file, please email [support@tuflow.com](mailto:support@tuflow.com).

#### 8.2.1 Package Model Fixes 2020-10-AA

Build 2020-10-AA includes the following fixes for the package model functionality

- The following file types are now skipped if they are missing. Previously it would give an error if the following files were missing. All these files can contain additional files that may need copying, e.g. a material.csv file can have a .csv file containing depth varying Manning's n data.
  - .tgc, .tbc, .trd, .adcf, .trfc, .tef, .qcf, .ecf

- 1d\_nwk
- 1d\_tab (1d\_xs, 1d\_hw\_1d\_bg)
- Database files (BC, AD, or pit inlet)
- .tmf materials files

- Fix issue that would fail to copy materials file if a global multiplier had been used. E.g. in the following .tcf command line a global multiplier of 1.2 is applied.

```
Read Materials File == materials_001.tmf | 1.2
```

Previously the materials\_001.tmf would have been missed. This can apply to .tmf or csv files.

- Issue a warning if package model attempts to create a folder with more than 248 characters (folder length excluding filename) or more than 260 characters (including filename). In this case the folder creation or file copy is likely to fail due to limitations in Windows. TUFLOW still attempts to create the folder or copy the file, however, if it fails it is clearer why this is failing.
- Package model will now copy all files if referencing MID files e.g. Read MID Zpts ==
- Package model will now treat file path references in 'Read Files' (.trd) correctly (consistent with model simulation).
- Package model will now copy the full GIS set of files if Read RowCol is specified:

For .mid input (e.g Read RowCol Mat == grid\_mat.mid) previously only the .mid file was copied, now the .mid, .mif, .dat, .id, .map and .tab files will be copied if present.

For .dbf input (e.g Read RowCol Mat == grid\_mat.dbf) previously only the .dbf file was copied, now the .dbf, .prj, .shp, .shx and .qpf files will be copied if present.

## 8.2.2 Package Model Fixes 2020-10-AB

Build 2020-10-AB includes the following fixes for the package model functionality:

- Package model will now correctly copy XF files when the following .tcf command:  

```
XF Files Include in Filename == <text>
```
- Package model will now copy rainfall grids referenced in the CSV file when using the following .tcf command:  

```
Read Grid RF == <CSV File>
```
- Package model will now correctly copy the inputs reference with the .tgc command:  

```
Create TIN Zpts == <GIS input>
```
- Package model will now copy .prj file if the "SHP Projection == " command references a .shp file and not a .prj file.

## 8.2.3 Package Model Change in Build 2020-10-AD

In Build 2020-10-AD, package model 'all' (-pmALL) no longer copies all raster files in the same folder that have the same name but different extensions. Previously this functionality would copy all files with the same name that could, as an example, include such files as vertical mapper '.grd' files, which are not used by TUFLOW and could significantly increase the size of the package model output.



### 8.3 Version (-version) Option

Specifying the new -version run-time option will return the TUFLOW Build. This can be useful in python or batch file scripts to cross-check or output the TUFLOW Build.

### 8.4 Optimising Multi-GPU Performance (HPC Only)

**Note: Multi-GPU is not yet supported for Quadtree meshes, so these enhancements are only applicable to non-Quadtree models.**

If a model is simulated across multiple GPU devices, one of the devices (usually the one with the most wet cells) will be controlling the speed of the simulation and the other devices will be under-utilised. By default, TUFLOW HPC divides a model equally over multiple GPU devices. However, for real-world models, it is usual for the GPUs to have an un-equal amount of workload due to the number of active cells and number of wet cells, and this can change throughout the simulation as the model wets and dries.

The TUFLOW 2020-01 release allows the user to distribute the workload unequally to the GPU devices. During a simulation the workload efficiency of each GPU is output to the console and to the .tlf file with a suggested distribution provided at the end of the simulation. A number of iterations may be required to fully optimise the distribution.

For example, a model simulated across four GPU devices reported at the end of the simulation in the the .tlf file:

- Relative device loads: 60.3% 100.0% 83.7% 53.2%
- HPC Suggested workload balance HPC Device Split == 1.23, 0.74, 0.89, 1.40

The command “HPC Device Split == 1.23, 0.74, 0.89, 1.40” was added to the .tcf file for the next simulation producing the improved device workload efficiencies below and a 20% faster run time!

- Relative device loads: 100.0% 96.1% 96.8% 94.6%

**Note:** The benefit depends on the model, but if you have a significant variation in workload efficiencies between GPU devices this feature should provide a noticeable decrease in run times.

### 8.5 NVLink – Multi-GPU Performance (HPC Only)

NVLink is a new connection (cable) for Nvidia GPU devices providing peer to peer (direct) access between GPUs, rather than having to communicate via the CPU, giving faster communication. For example, data transfer rate for the RTX 2080 Ti is 32GB/s on the CPU PCIe and 100 GB/s via an NVLink.

The TUFLOW 2020-01 release automatically recognises and utilises any peer to peer (p2p) access between GPUs that is possible according to the hardware setup. Peer to peer access typically requires an NVLink connector between GPUS, or in some cases peer to peer access can occur via the PCI bus if all cards are placed into TCC driver mode.

Whilst testing thus far has not produced a huge jump in performance, it is expected greater gains in the future will arise as it is increasingly possible to connect to large numbers of GPU devices, especially Cloud based instances.

### 8.5.1 New Option to Control Peer to Peer GPU Access

For Build 2020-10-AB, when running a HPC simulation across multiple GPUs, the user can choose to disable peer to peer GPU access by specifying the .tcf command:

```
GPU Peer to Peer Access == DISABLED | {ENABLED IF AVAILABLE}
```

This can also be specified using the command line argument -p2p0 to disable, or -p2p1 to enable (if available). The default is to use peer to peer access if available. Peer to peer access in rare occasions has failed to work when a peer connection has been reported as possible by the nVidia console or API.

## 8.6 Shapefile Projection Changes

### 8.6.1 Projection Check

Build 2020-01-AB introduces a more robust processing of the shapefile format projection (.prj) file to ensure that an input GIS file has the same projection as the model projection. Prior to 2020-01-AB this file was read as a character string and when a projection check was performed a string compare was performed to check that the input file projection matched the projection defined for the model (with the SHP Projection == command). This could cause issues if moving between GIS platforms, e.g. between ArcMap and QGIS as these could write the .prj files slightly differently, particularly with a different number of decimal places. For example, consider the two extracts from .prj files; "PARAMETER["latitude\_of\_origin",0]" and "PARAMETER["latitude\_of\_origin",0.0]". A string compare would highlight these as not matching. Build 2020-01-AB parses the parameter data into number values and compares each of these numbers to check they are consistent. This should provide a more robust check less prone to falsely flagging the projections as different. A new .tcf command has been added to allow setting to the previous "Simple" string compare:

```
SHP Projection Check Method == SIMPLE | {PARSED}
```

The consistency check for GIS projections can be turned from an ERROR (default) to a WARNING with the command: `GIS Projection Check == {ERROR} | WARNING`

Build 2020-10-AA fixes an issue with the parsing of the projection file when there were duplicate parameter entries with the same name. This could occur if the .prj file also defined a Vertical Coordinate System (vertcs). In this case, the 2020-01-AB Build would cause an ERROR 0184 – Unable to allocate memory for projection parameters. This issue only occurred if using the default

```
SHP Projection Check Method == PARSED
```

Build 2020-10-AB increases the allowed tolerance between projection attributes when doing a compare when the SHP Projection Check Method == PARSED. For example, the case below will now be considered the same.

```
...],PARAMETER["false_easting",6561666.666666666],PARAMETER["False_Easting",6561666.666666667],PARAMETER
```

## 8.7 Ability to Pause a Simulation at Set Time / Interval

The 2020-10-AA Build introduces the ability to pause a TUFLOW simulation at set intervals during the simulation. This will create a Console pause requiring a carriage return (enter key) to be pressed before the simulation will continue. This feature has gone in at a user's request for coupling of TUFLOW with a groundwater model, with the aim to pause the TUFLOW simulation, allow the groundwater model to run and update rainfall grids to apply fluxes in the TUFLOW model.

```
Simulation Pause Start == <Model time in hours>
```

```
Simulation Pause Interval == <Time interval in hours>
```

For example, the following commands will cause the simulation to pause at 96 hours, and then every 24 hours for the remainder of the simulation.

```
Simulation Pause Start == 96
```

```
Simulation Pause Interval == 24
```

## 8.8 Copy Model Functionality

### 8.8.1 Copy Model Fixes for 2020-10-AB

Build 2020-10-AB includes the following fixes for the copy model functionality:

- Copy model will now copy rainfall grids referenced in the CSV file when using the following .tcf command:  

```
Read Grid RF == <CSV File>
```
- Copy model will now copy the .tcf file into the correct location. Previously when using a relative reference in the batch file, e.g. `tufLOW.exe -c my_model.tcf`, the .tcf file would sometimes be copied outside of the directory created for the rest of the model.

### 8.8.2 Copy Model Change for 2020-10-AD

In Build 2020-10-AD, copy model 'all' (-ca) will no longer copy all raster files in the same folder that have the same name but different extensions. Previously this functionality would copy all files with same name that could, as an example, include such files as vertical mapper '.grd' files, which are not used by TUFLOW and could significantly increase the size of the copy model output.

## 8.9 Backwards Compatibility "Defaults" command

The HPC Solver was first introduced in the 2017-09 release of TUFLOW and the Quadtree solver was introduced in the 2020-01 release. To prevent incorrect use of "Defaults ==", TUFLOW Build 2020-10-AC issues ERROR 2484 and halts the simulation for:

- HPC simulations with "Defaults == Pre 2017" or earlier; and
- Quadtree simulations with "Defaults == Pre 2020-01" or earlier.

## 9 Minor Enhancements and Bug Fixes for 2018-03 Release

### 9.1 Bug Fix for FC Shape Where Negative Depth Caused NaN

2020-01 fixes a bug which caused a Not a Number (NaN) to be generated for a cell side that experienced a negative depth if a Flow Constriction Shape was applied.

### 9.2 Bug Fix for SA streams

Fixed a bug that could cause a Not a Number (NaN) to be generated if a SA inflow region was being applied to both stream and wet cells. This would occur, if all the streams cells became dry whilst there were still wet cells within the region, this caused a divide by zero and a NaN was generated.

### 9.3 Bug Fix for PLOT\_R

The PLOT\_R file written to results\plot\gis\ was not being written correctly. This would occur for the .shp file and for complex region objects for the .mif file (i.e. internal holes existed in the polygon). This has now been fixed.

### 9.4 Enhancement to the QGS File

The .qgs file written by TUFLOW (runs\log\ ) now includes projections. When opened in QGIS version 3.0 or later it should now correctly set the project projection and projection of all vector layers.

### 9.5 Bug Fix for Folder Creation with UNC Paths

Build 2020-01-AB fixes an issue that occurred in 2020-01-AA when TUFLOW was creating a folder that included a Universal Naming Convention (UNC) path for example:

Output Folder == \\server\share\project\TUFLOW\results\

For the 2020-01-AA version, TUFLOW would stop with an error saying unable to create folder.

### 9.6 Bug Fix for Default == Pre 2013

Build 2020-01-AB fixes an issue which would cause the 2020-01-AA release to stop with an error when run with Defaults == Pre 2013.

### 9.7 Message Box Changes

Build 2020-01-AB changes the message box behaviour, for the 2020-01-AA version of TUFLOW messages boxes created by TUFLOW could lock other windows until the message had been dismissed.

### 9.8 Minor Changes for 2020-01-AB

Build 2020-01-AB has the following minor changes:

- Additional empty (template) GIS files are created when running with "Write Empty GIS Files == " command.

## 9.9 Minor Changes for Build 2020-10-AA

Build 2020-10-AA has the following minor changes:

- When running Quadtree or SGS models, at the end of the simulation allocated memory is released. This prevents TUFLOW simulations linked with external 1D engines from failing to initialise if running multiple events.
- A WARNING 2468 message is output for any cells that have no active cell faces. This can occur if a 1D/2D connection is activating a 2D cell surrounded by null cells.
- An ERROR 2469 message is output, and the simulation halted if an output zone is defined without a Read GIS Output Zone command. Prior to the 2020-10-AA release this could cause a simulation to crash when writing results.
- Fixed instances which would cause WARNING 0255 (unclosed GIS layers) to occur at the end of the simulation. This did not affect the integrity of the GIS files output from TUFLOW.
- The default restart file version when writing restart files is version 2 (Write Restart File Version == 2). Prior to Build 2020-10-AA this was set to version 1 for Classic simulations and version 2 for HPC simulations.
- A check is performed on the length of the absolute file path to the .tcf. If this is more than 260 characters WARNING 0624 is issued prior to opening the file. In this situation the .tcf may fail to open, so this WARNING makes it clearer that the issue may relate to the path length.
- An ERROR 1423 is generated when a virtual pit outlet (VPO) is connected to a channel that already has another user defined node, pit, or manhole in the same location (See Section 6.6.1)
- Build 2020-10-AA replaces ERROR 2217 in Quadtree with WARNING 2215. This error would occur when using a pit (1d\_nwk or 1d\_pit) to connect 1D and 2D domains with an SX connection that tried to connect more than 10 cells. Quadtree will now limit the number of connected cells to 10 and issue a warning and continue. This is consistent with TUFLOW Classic and HPC.
- Build 2020-10-AA will simplify the mesh extent polygon written as part of the 2d\_dom\_check file in Quadtree if the number of points in the region exceed TUFLOW's upper limit by removing all internal regions.

## 9.10 Bug Fix for the Temporal WLL Map Output of an External 1D Scheme's Results

Use of the 2020-10-AA Build with one of the TUFLOW linked external 1D schemes, i.e. 12D DDA, Flood Modeller or XP-SWMM, will require an update from these software due to a change in the calling routines between external 1D schemes and TUFLOW. This is in relation to a bug fix for the temporal WLL map output of the external 1D scheme's results for linked TUFLOW HPC models only. Note that the bug fix:

- Does not affect the hydraulic calculations and does not affect the external 1D scheme's WLL map output for maximums (it only affects the temporal WLL map output).
- Only applies to TUFLOW HPC externally linked models – it does not affect TUFLOW Classic linked models (but the updated external software that works with 2020-10-AA or later is required irrespective of whether Classic or HPC solver is used).
- Has no influence on TUFLOW 1D (ESTRY) WLL map output.

For Flood Modeller users this will require an update to version 4.6 once released.

For the 2020-01-AB Build issued on May 5, 2020, a WARNING 2465 is issued if external 1D WLLs are present in the map output.

Note: 2020-10-AA will only work with the updated external scheme software once available, and 2020-01-AB or earlier will only work with current and past releases of the external 1D scheme software.

## 9.11 Minor Changes for Build 2020-10-AB

Build 2020-10-AB has the following minor changes:

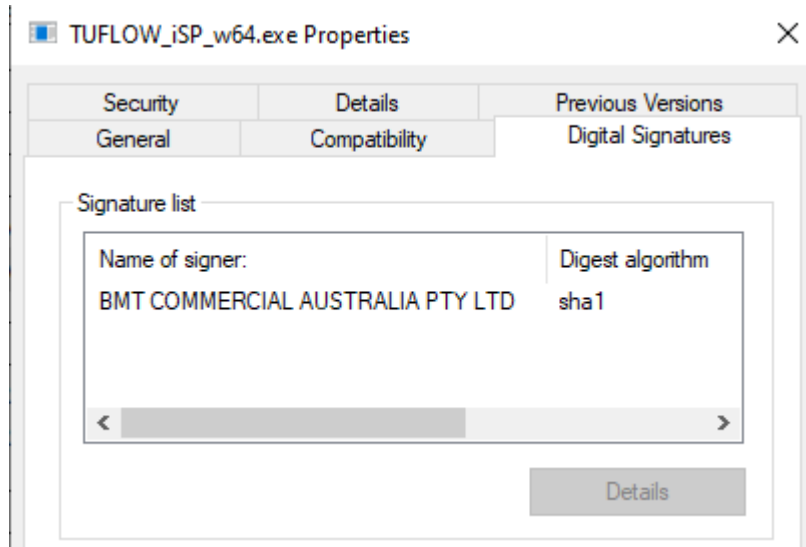
- When using "Create TIN Zpts Write TIN ===" the .tin file created now includes the end of tin (ENDT) flag at the end of the file. This was previously missing, but did not limit the ability for the file to be read into either TUFLOW or SMS.
- When running TUFLOW HPC simulations in test model (with the -t command line argument) the simulation progresses further with the HPC solver being initialised, i.e. the data being loaded onto GPU. Previously TUFLOW would halt after the check files had been written.
- When running on GPU, extra output is written to the screen and log file prior to interrogating compatible GPU. If there is an issue with the querying the GPU (for example due to a driver issue), then the messaging helps to diagnose the issue.



## 10 Licensing and Installing

### 10.1 Security Certificate

The TUFLOW 2020-01 release is digitally signed, this can be checked by right clicking on the .exe file and selecting properties, under the “Digital Signatures” tab the following should be present.



### 10.2 Bug fixes and Minor Enhancements

#### 10.2.1 Excess Licences Released for Quadtree Simulations

When running Quadtree simulations with the 2020 releases of TUFLOW only a single compute device (GPU or CPU) can be used. In the instance that multiple devices are specified TUFLOW will issue a WARNING 2803. Prior to Build 2020-10-AA, the licences remained in use until the end of the simulation. For Build 2020-10-AA onwards any excess licences are released prior to the commencement of the simulation.

#### 10.2.2 Support Period Now Displayed

From Build 2020-10-AA onwards the support and maintenance year the version of TUFLOW falls in is now displayed to the screen and log file. For 2020-10-AA this is July 2020 - June 2021.

#### 10.2.3 Command Line Argument -nwk (Network only) Now Works for CodeMeter Licences

For Build 2020-10-AB the command line argument -nwk is now supported for CodeMeter licences. This forces TUFLOW to only look for a network licence, i.e. skip the search for a local licence. Prior to this build the -nwk option only worked for older Softlok licences.

#### 10.2.4 Check on the GPU .ptx File Version

For Build 2020-10-AB a check is performed so that the version of the .ptx files matches the expected version. This aims to prevent these files being incorrectly copied / moved. It is strongly recommended to always keep all files in the TUFLOW download together.

to this build the -nwk option only worked for older Softlok licences.

#### **10.2.5 Updated licencing when running from 12D Model**

For Build 2020-10-AC Quadtree module licencing is supported when running from 12D. Note that this will require 12D Model 15 or newer.

## 11 Backward Compatibility to the 2018-03 Release

The software compilers used for TUFLOW Classic/HPC have been updated to their latest versions for the TUFLOW 2020-01 release. This may cause slight differences in results due to changes in compiler code optimisation (all other settings being the same). Therefore, it is not possible to provide bitwise identical results to the 2018-03 release using the 2018-03 default settings. Overall, differences will be fractions of a mm, although around the wet/dry boundary some greater differences may exist.

For the TUFLOW Classic 2D solver and the 1D (ESTRY) solver the results should be the same or very similar to the 2018-03-AE release as the default settings for these solvers have not changed.

For the TUFLOW HPC 2D solver, the new turbulence scheme (Section 5.1), improved treatment of inflows along HT and QT boundaries (Section 6.1) and enhanced treatment of the turbulence term along the wet/dry interface (Section 5.2.3) have been set as the new defaults as these features offer substantial benefits, especially the new turbulence scheme. The results will differ by varying degrees depending on the type of model (for example, friction dominated models will change significantly less than low friction, highly transient or complex flow models, especially where the cell size is less than the depth or if modelling at flume-scale). To revert to the same settings as for the 2018-03 release the following commands will be required in the .tcf file:

```
Viscosity Formulation == Smagorinsky
Viscosity Approach == Method A
HPC Mannings Depth Approach == Method A
HPC Boundary Approach == Method A
```

The above can all be invoked with the .tcf command:

```
Defaults == Pre 2020-01
```

As always, it is recommended when switching to a new build with an established model that test runs be carried out and comparisons made between the old and new builds (subtracting the two maximum h data sets and reviewing any differences is an easy way to do this). If you have any queries on the comparison outcomes, or require clarification or more detail on any of the points below, please email [support@tuflow.com](mailto:support@tuflow.com).

## 12 Backward Compatibility to the 2020-01 Release

For the October 2020-10-AA release of TUFLOW, the same software compilers as the January 2020-01 versions of TUFLOW have been used. For TUFLOW Classic the results should be identical with the 2020-01-AB release of TUFLOW noting the Global Rainfall and materials change described further below.

For HPC Grid and Quadtree simulations the following bug fixes may cause subtly different results:

- Timestepping change outlined in Section 6.3.8 – HPC including Quadtree
- Quadtree only Wu turbulence bug fix outlined in Section 6.3.7.
- Quadtree only cell selection at changes in resolution, see Section 3.1.8

As well as the bug fixes above for which there are no backwards compatible options, the following changes may also change results between the 2020-10-AA version and the 2020-01-AB version.

```
Map Output Vector Dry Zpts == ON
Map Cutoff Vector == OFF
HPC HQ Boundary Approach == CELL
GIS Project Path Format == ABSOLUTE
```

For TUFLOW Classic simulations with Global Rainfall Boundaries and Rainfall losses in the materials file, the following command will be required. Refer Section 6.4.

```
Global Rainfall Use Material Loss == OFF
```

The above can all be invoked with the .tcf command:

```
Defaults == Pre 2020-10
```

As always, it is recommended when switching to a new Build with an established model that test runs be carried out and comparisons made between the old and new Builds (subtracting the two maximum h data sets and reviewing any differences is an easy way to do this). If you have any queries on the comparison outcomes, or require clarification or more detail on any of the points below, please email [support@tuflow.com](mailto:support@tuflow.com).