



TUFLOW Classic and HPC 2020-01 Release Notes

Document Updates and Important Notices

(in reverse 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.

All users of the 2018-03 and prior releases, especially if using the HPC solver, are recommended to utilise the superior functionality of the 2020-01 release.

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 and with updates to the 2020-01 release.

Summary

The TUFLOW Classic/HPC 2020-01 release represents 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 release is, 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 or later 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 tutorial and demo models, or if running in free demo mode, no licence is required. For any licensing enquiries please contact sales@tuflow.com, or for general support support@tuflow.com. Use of the TUFLOW software in any mode is bound by the [TUFLOW Products Licence Agreement](#).

The 2020-01-AA release includes 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.

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

For the 2020-01 release, 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

Document Updates and Important Notices	1
Summary	2
Licensing and Executable Versions	3
1 2020-01 Release Overview	6
2 1D Domain Construction	7
2.1 Faster Start-Up Times for Large 1D Networks	7
3 2D Domain Construction	8
3.1 Quadtree Mesh (HPC Only).....	8
3.2 Sub-Grid Sampling, SGS (HPC Only, including Quadtree)	15
3.3 Non-Linear Failure of Variable Z Shapes	23
3.4 2D Bridge Decks	24
4 1D Solver (ESTRY)	26
5 2D Solvers (Classic/HPC)	27
5.1 Overview	27
5.2 Mesh Size Insensitive Turbulence Model (HPC Only)	27
5.3 Flow depth used for bed friction calculation (HPC Only, incl. Quadtree excl. SGS)	31
5.4 Non-Newtonian Flow (HPC Only)	32
5.5 HPC Advection-Dispersion (AD) Solver (New to HPC)	34
6 Boundaries and Links	35
6.1 Improved HPC Water Level (H and QT) Boundaries (HPC Only)	35
6.2 Bug Fixes and Minor Enhancements.....	35
7 Outputs and Check Files	36
7.1 Time-Series Output in NetCDF format	36
7.2 Extra plot outputs.....	36
7.3 NetCDF Map Output	37
7.4 Quadtree Map Output	37
7.5 SGS Map Output	37
8 Simulation Control	40
8.1 Default Simulation Log Folder Changed.....	40
8.2 Package Model (-pm) Enhancements	40
8.3 Version (-version) Option.....	40
8.4 Optimising Multi-GPU Performance (HPC Only)	40
8.5 NVLink – Multi-GPU Performance (HPC Only)	41
9 Minor Enhancements and Bug Fixes	42
9.1 Bug fix for FC Shape where negative depth caused NaN.....	42
9.2 Bug fix for SA streams	42



9.3	Bug fix for PLOT_R.....	42
9.4	Enhancement to the QGS file	42
10	Licensing and Installing	43
10.1	Security Certificate.....	43
11	Backward Compatibility to the 2018-03 Release	44

1 2020-01 Release Overview

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.

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.

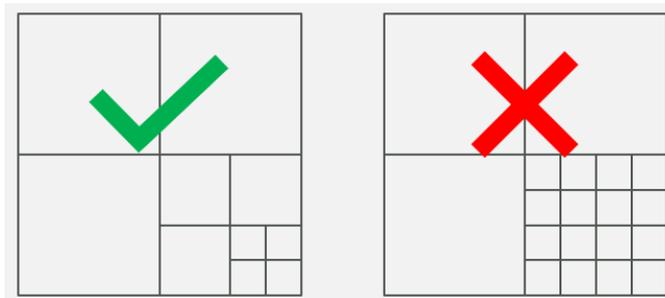
3 2D Domain Construction

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.

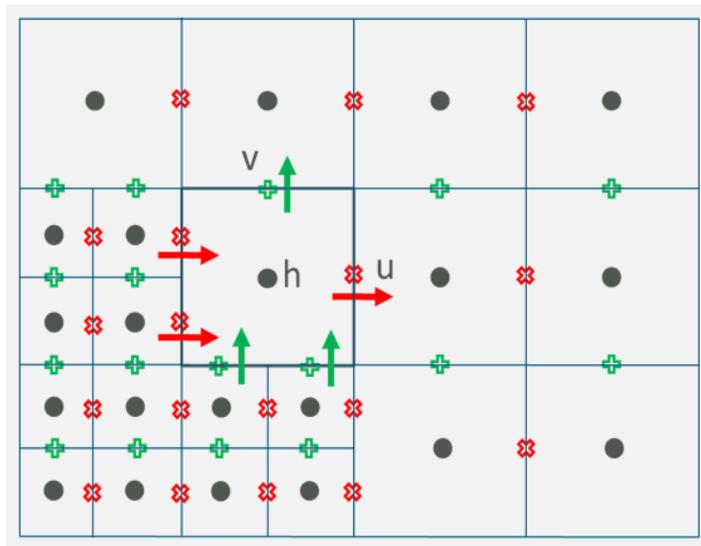
Each 2D cell face can have a maximum of 2 adjoining cell faces, meaning that multiple levels of refinement will need to transition through the intermediate resolutions. For example, the image on the left is a valid Quadtree mesh whereas the image on the right shows an invalid mesh. It is possible for 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 2nd order in space and 4th 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 scheduled for development during 2020 and will be released once completed and tested.

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

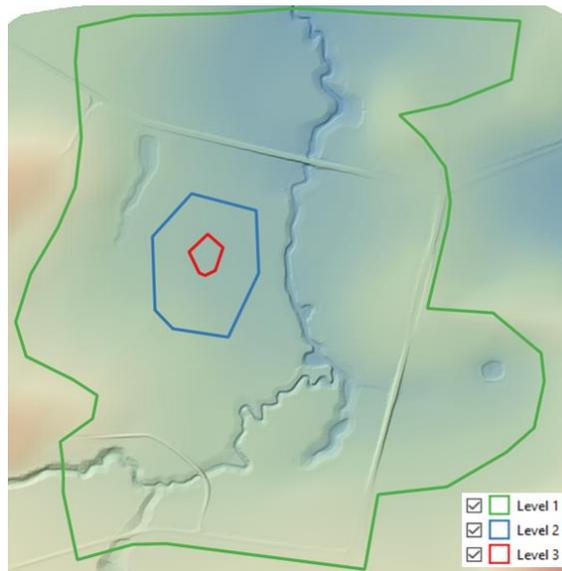
A 2d_qnl (Quadtree Nesting Level) GIS layer is used to define the location and levels of mesh refinement. An empty (template) 2d_qnl GIS layer can be created in the usual manner by using “Write GIS Empty Files ==”. Alternatively, the layer only requires a single attribute of Integer type, nominally called “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

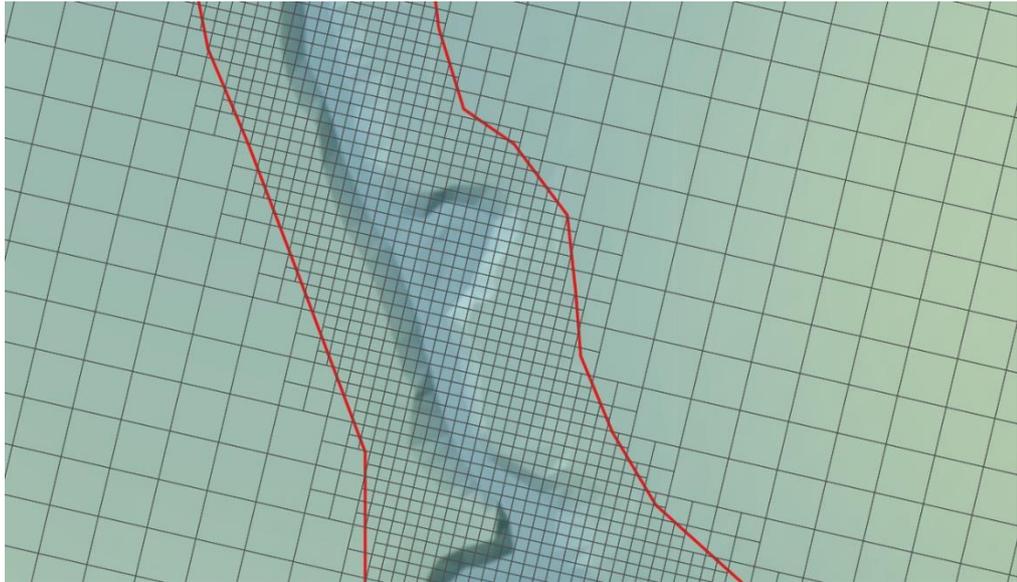
indicates the cell size within the polygon would be at Level 2 (i.e. half the base cell size). 3 would be cells at $\frac{1}{4}$ of the base cell size, 4 for $\frac{1}{8}^{\text{th}}$ and so on up to a maximum of 9 ($\frac{1}{256}^{\text{th}}$). For numerical precision reasons, the maximum nesting level 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 one Level 1 polygon defined in the 2d_qnl layer, but for all other levels there is no limit on the number of polygons.

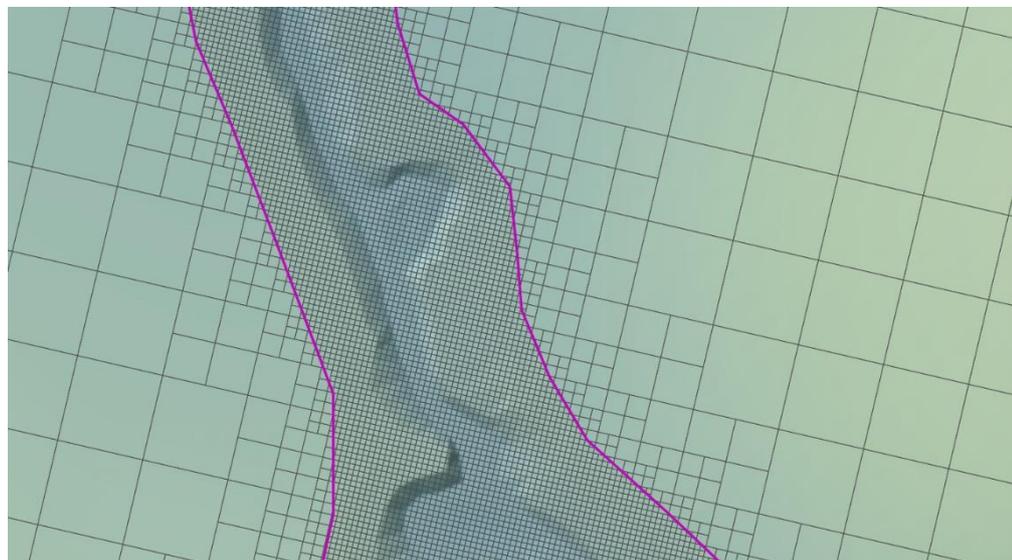
When refining mesh areas, if a refinement polygon sits within a polygon of the next higher level, e.g. a Level 3 polygon is defined within a Level 2, as per the image below, no automatic meshing is required.



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

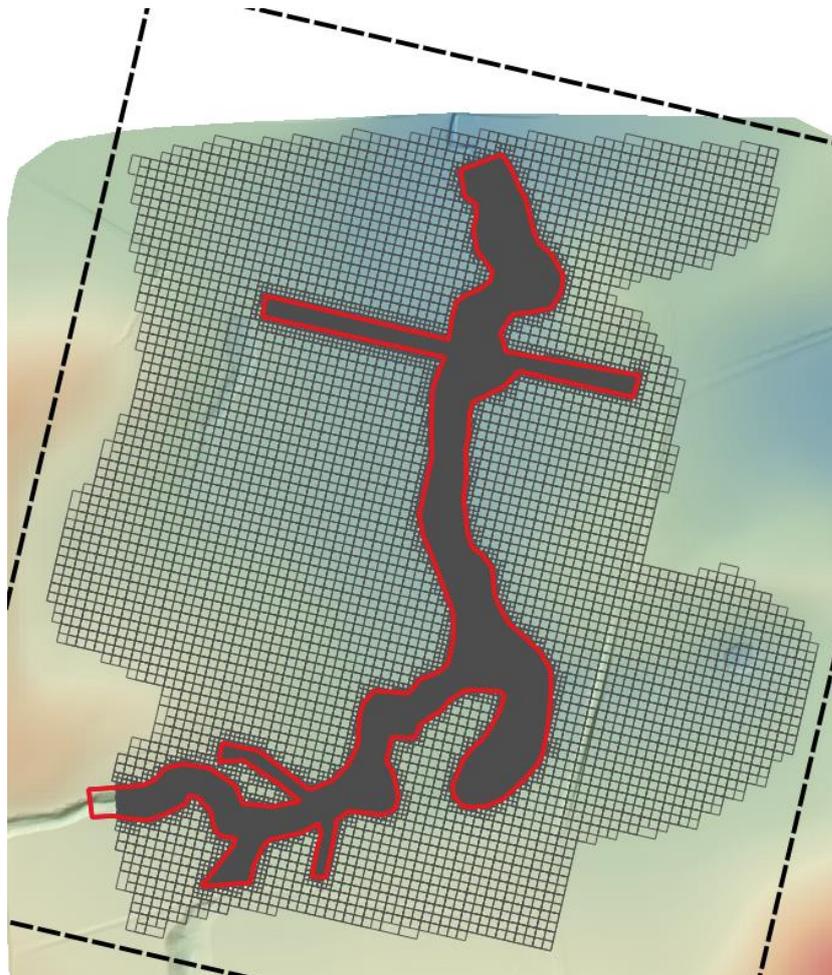
Note: No Level 1 polygon is required if the model origin and extent are defined in the .tgc file. In this situation the rectangle representing the .tgc computational domain is used as the Level 1 polygon. For example, if the .qcf file includes the following commands and the only 2d_qnl polygon is Level 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.

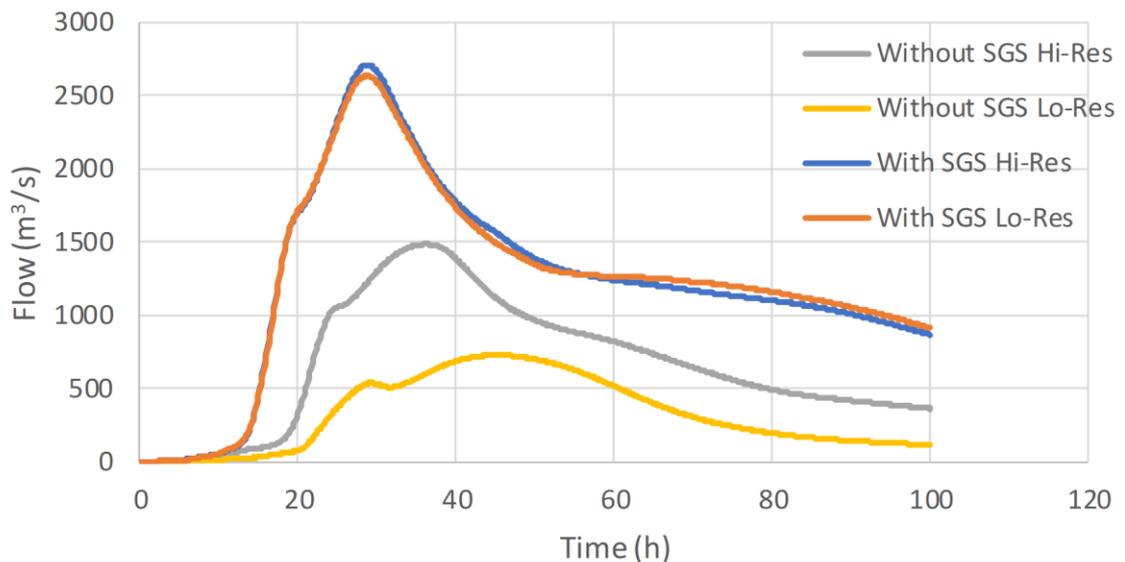
Small Quadtree models with refinement can be run with the demo or free version of TUFLOW by using "Demo Model == ON" in the .tcf file. The same limitations as for a single grid apply (i.e. the total number of cells in the Quadtree mesh must be less than 30,000 regardless of refinement level and the simulation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

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

3.2.1 Introduction

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:

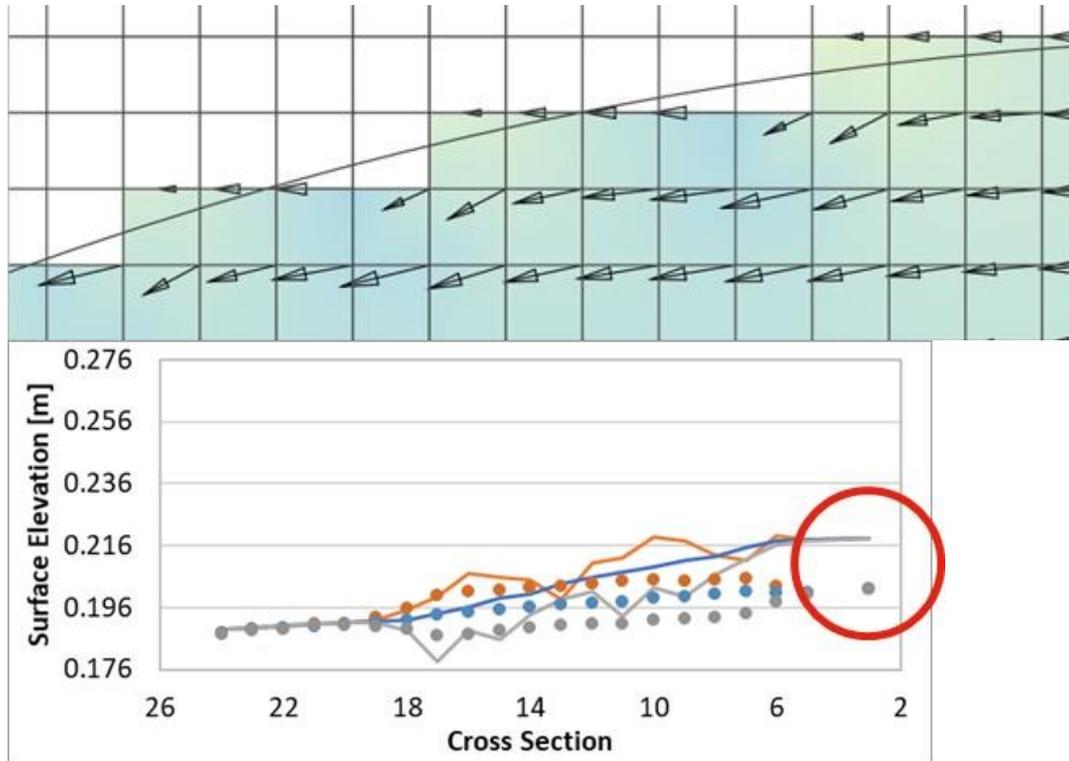
- Catchment scale models flow much more effectively with water not being “trapped” by a coarse cell resolution, and, importantly, excellent cell size convergence (ie. demonstration that by reducing the cell size(s) the model results do not demonstrably change) at much coarser cell sizes. The chart below 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.



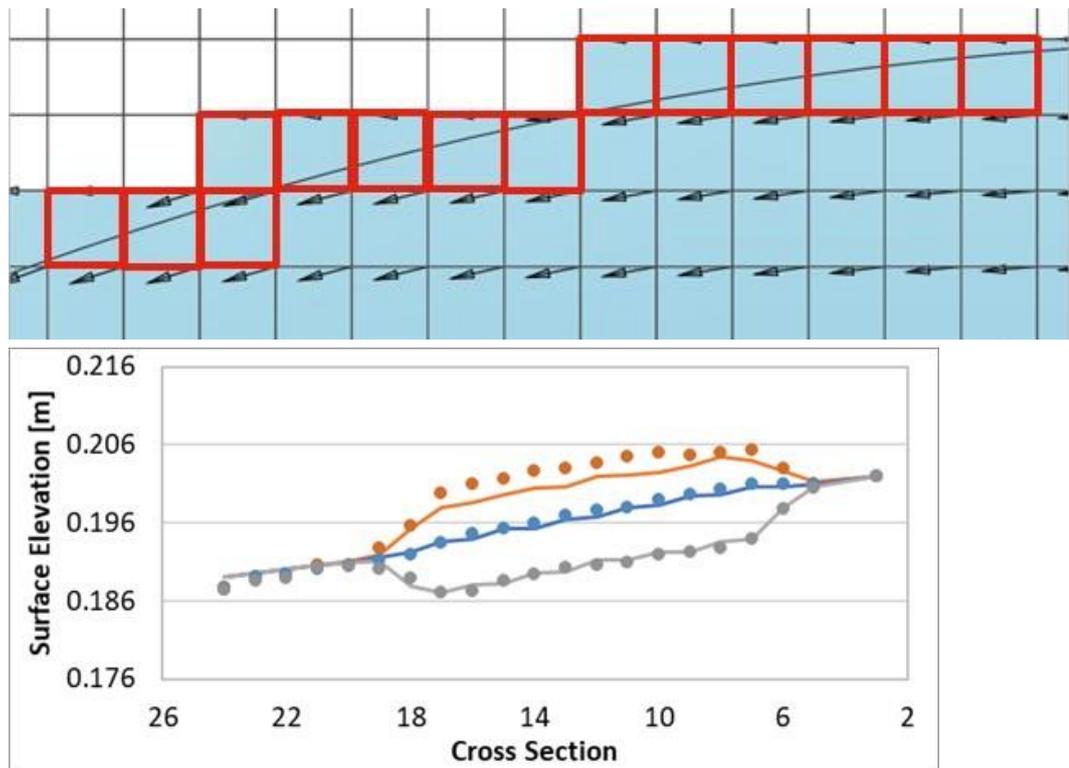
- Disturbed flow fields that can be apparent along a “saw-tooth” regular mesh wet-dry boundary completely disappear, with no spurious additional head losses generated and the results consistent with a well-designed flexible mesh. This has major benefits in that open channels can now be accurately modelled 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)

For both TUFLOW Classic and TUFLOW HPC 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).

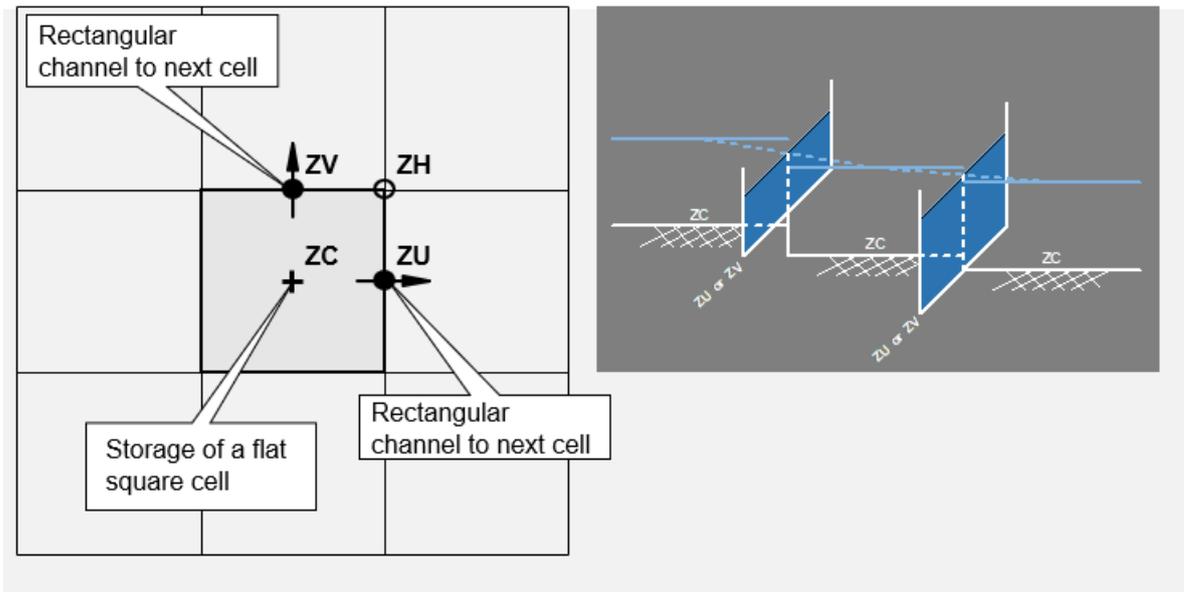


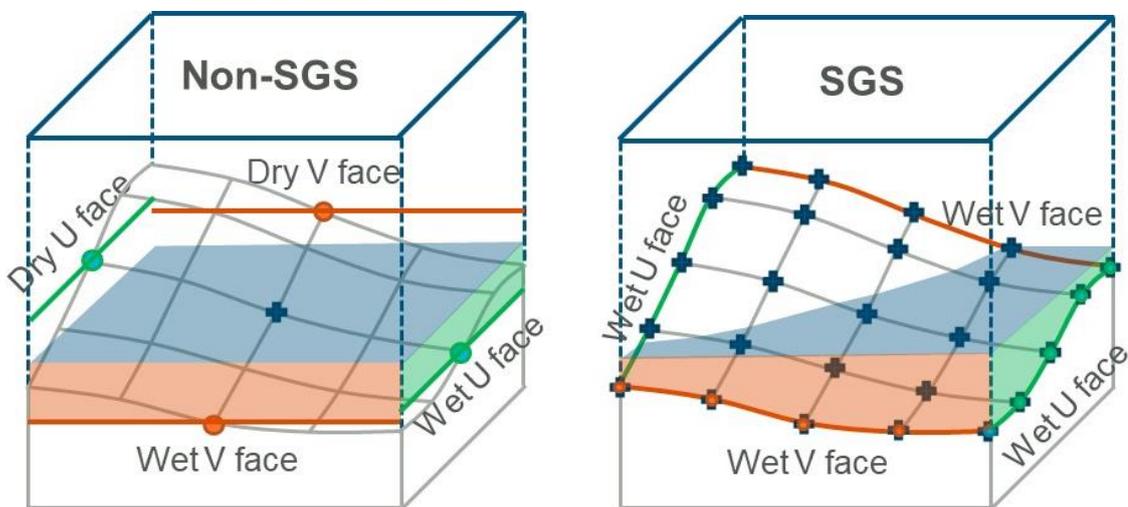
Diagram of Standard TUFLOW Cell Architecture

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.

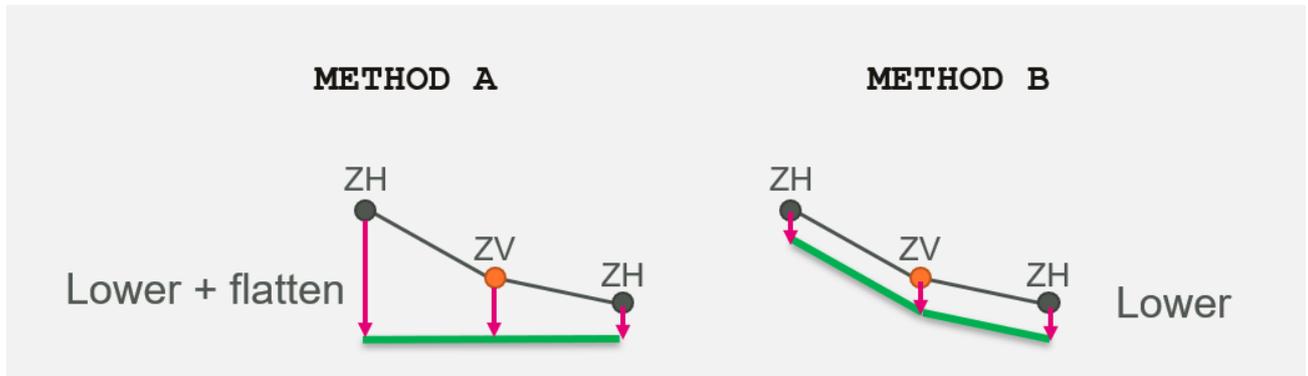


Diagram of SGS SX Z Flag Approach Options

`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.

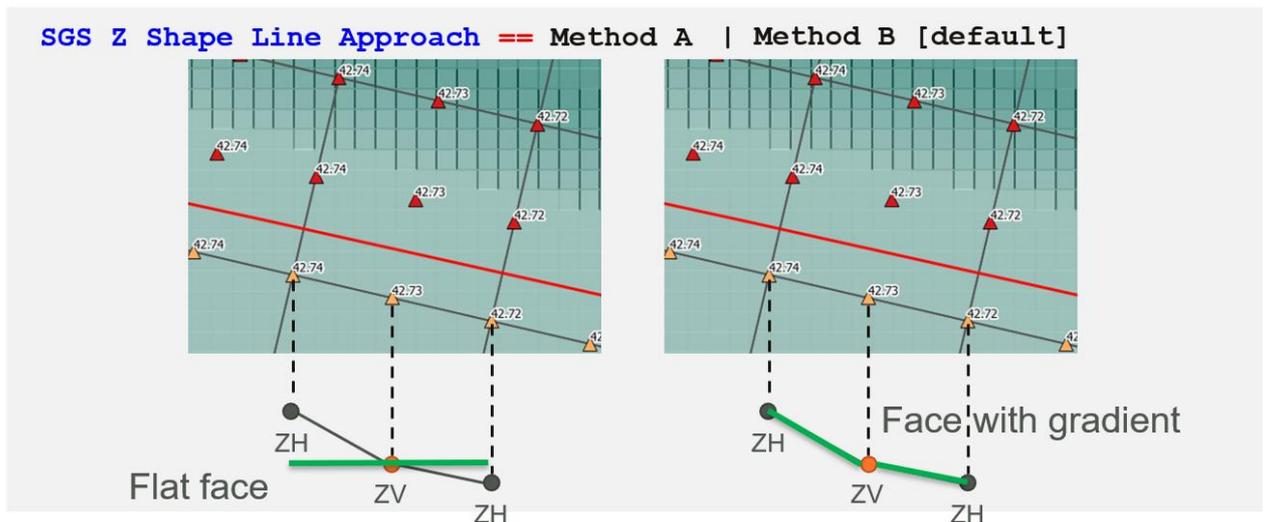


Diagram of SGS Z Shape Line Approach Options

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

See discussion in Section 7.5.1 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
```

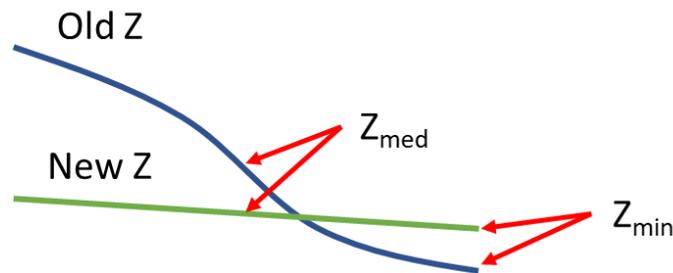
Ignores the MAX/MIN options and always applies the new elevations.

```
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

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 ==).

- 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: 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 1D Nodes with SXZ flag 1D Pits with SXL flag	Read GIS Z HX Line Read GIS Z Shape Route 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

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 “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.
- The “Zmax” attribute is the maximum elevation, ie. the elevation at which the cell area or cell face flow width is fully wet.
- “Zavg” is the average (mean) elevation of the sampled values.
- “Zmed” is the median elevation of the sampled values.



Example SGS Check File Output

3.2.7 SGS Map Outputs

When running a model with SGS enabled, outputs are on a TUFLOW cell resolution and not on the SGS resolution. When flagging a cell as wet or dry by default TUFLOW uses the Map Cutoff Depth which compares the depth of water using the minimum cell elevation.

When running an SGS model in addition to using a Map Cutoff Depth, a Map Cutoff SGS == command can be used. If both a Map Cutoff Depth and a Map Cutoff SGS command are used the higher of the two depths is used. For example:

```
Map Cutoff Depth == 0.02
Map Cutoff SGS == Perc | 5
```

For more discussion and options refer to Section 7.5.1.

3.2.8 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.

- 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 infiltration models 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

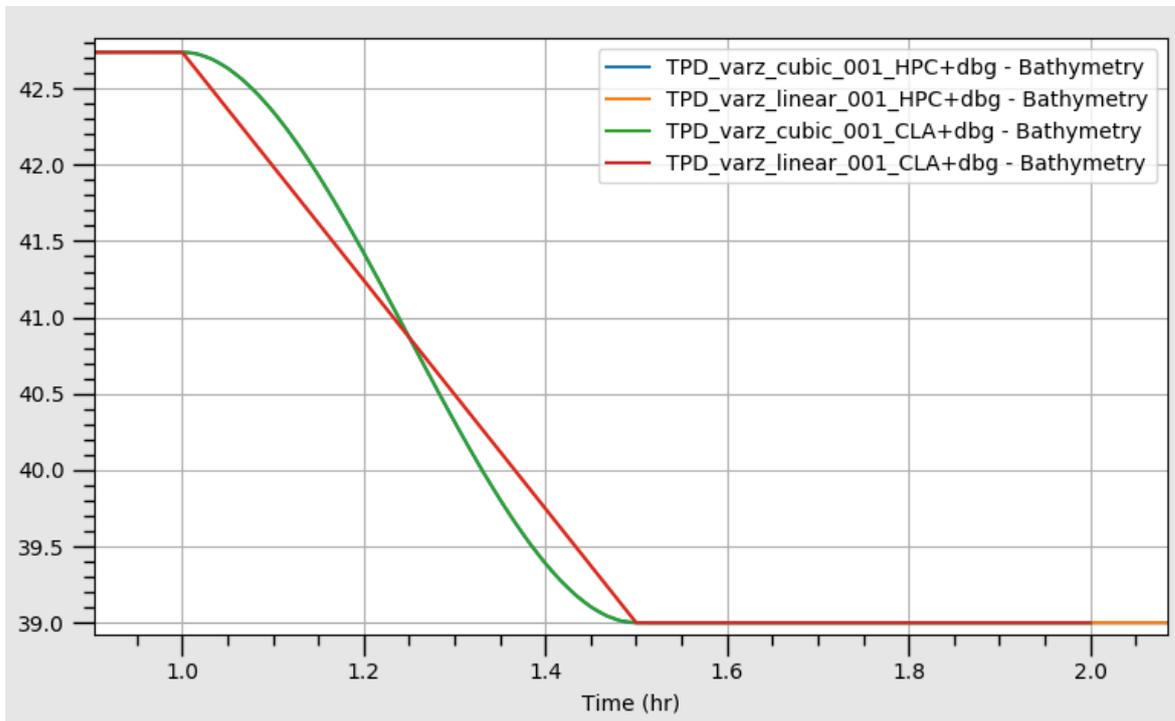
to conform with that required for direct rainfall, which assumes the rainfall is applied over the entire cell irrespective of whether the cell is partially wet or not. Likewise, soil capacity is based on the total cell area (i.e. infiltration will cease once the cumulative infiltration equals soil capacity), and the cumulative wet time for the Horton model will also increment for cells that are partially wet.

3.3 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.

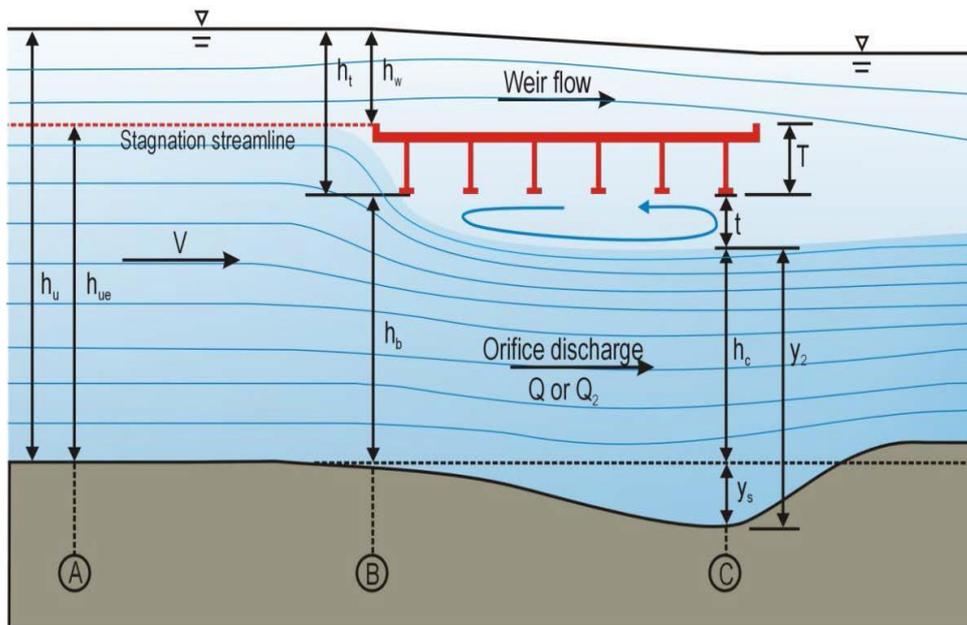


3.4 2D Bridge Decks

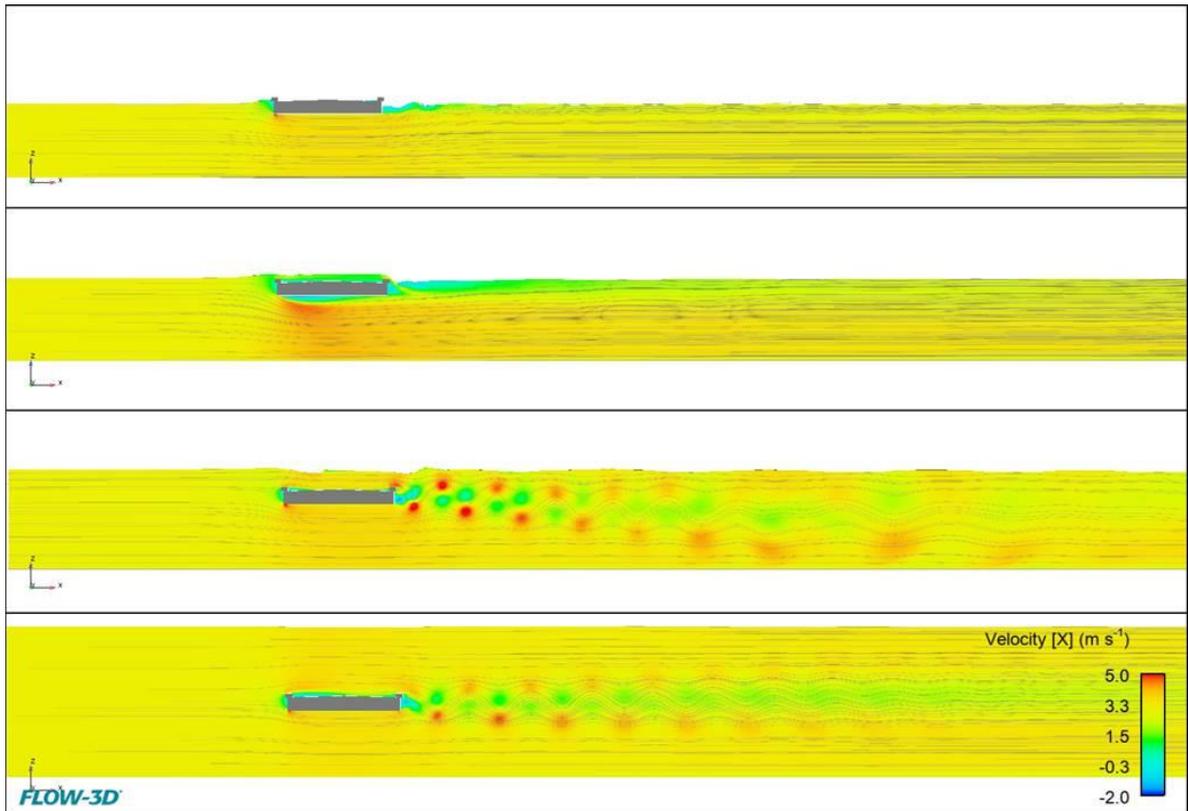
Modelling bridges with layered flow constrictions is not an exact science

Historically three-layered approach used with either accumulated or pro-rata losses

A joint research exercise with the Queensland Department of Transport and Main Roads (DTMR) 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, and subject to more benchmarking, the new approach will be formally released.

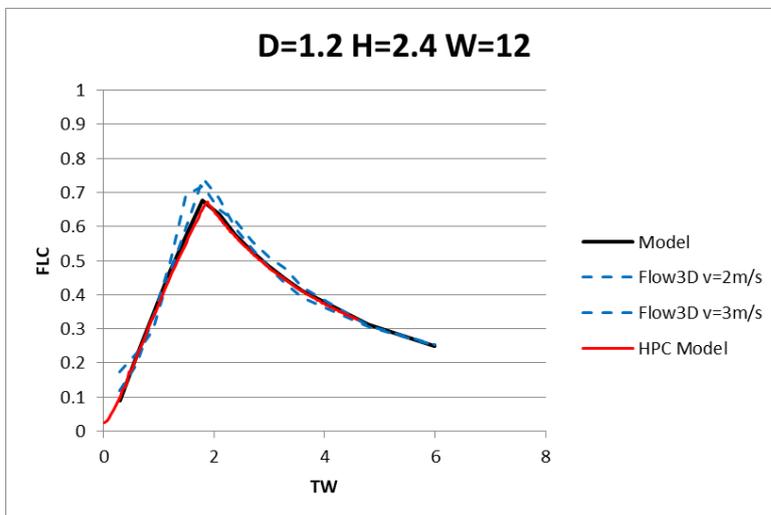


DTMR have carried out numerous Flow3D CFD simulations across a range of deck dimensions and deck to depth ratios for a solid deck configuration to determine head losses for flow surcharging against or over a bridge deck, including pressure flow conditions. Of interest is that the maximum energy loss occurs for an upstream level that is above the deck invert. The image below shows the results from one of the deck configurations for various flow stages.



Promisingly, a consistent relationship on the energy loss versus the deck depth, downstream water level and other parameters has been observed. This relationship has been built into Layer 2 of a 2D layered flow constriction in the TUFLOW HPC 2D solver with good reproduction of the energy loss due to the bridge deck as estimated by the Flow3D CFD modelling (see chart below).

We are aiming to finalise this functionality and provide improved guidelines on modelling bridges in 2D during 2020. If you or your organisation has any suitable data (flume or real-world) that can be used for benchmarking, please contact support@tuflow.com. The data would need to provide reliable estimates of the flow rate and water levels upstream and downstream of the bridge.



4 1D Solver (ESTRY)

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

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 10th 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. 2005¹. 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

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

¹ 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: 1st 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.

1st 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 1st order spatial solution and for early calibration of the Brisbane River modelling required lower Manning’s n values to calibrate compared with 2nd order schemes).

Numerical dispersion also helps stabilise a model, but for the wrong reasons. A 2nd 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 1st order scheme can exhibit turbulence like effects and good stability but is not physics based and will not be as accurate as 2nd order schemes. And 2nd 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 2nd 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.

5.3 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 τ_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.

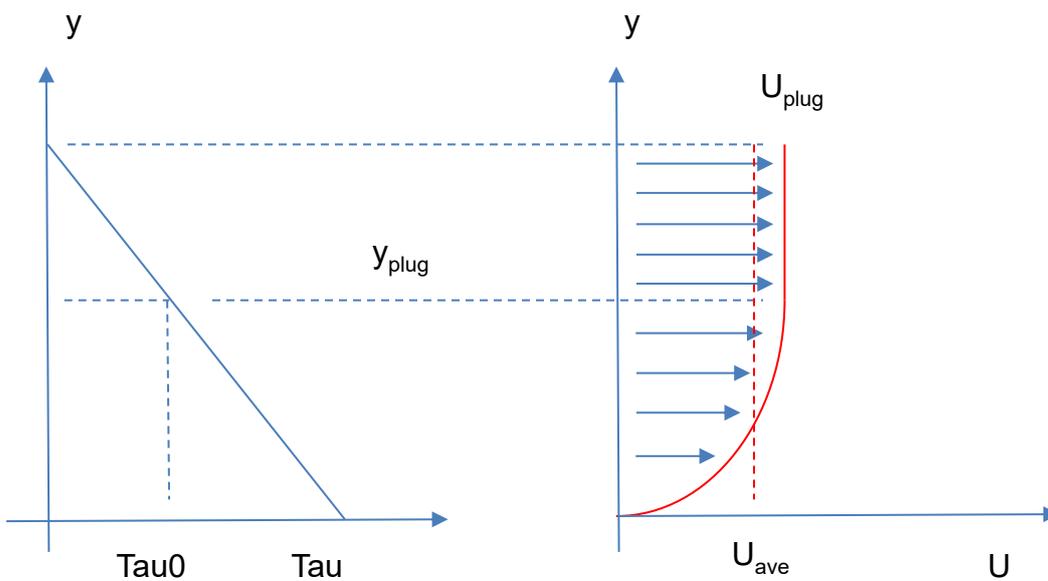


Figure 1 Non-Newtonian Plug Flow

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 μ_{2D} can produce unbounded results when $n < 1$ and $|\dot{\gamma}|$ tends to zero. Therefore, upper and lower viscosity limits, μ_{low} and μ_{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, μ_{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, μ_{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 [Pa sⁿ], mu limits are in [Pa s], and tau0 is in 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.

5.5 HPC Advection-Dispersion (AD) Solver (New to HPC)

Note: This new feature applies to TUFLOW HPC only, 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 is computed using the full anisotropic diffusion matrix (Falconer et. al. as references in TUFLOW AD manual) rotated by flow direction.

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

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

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

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

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 –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 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.

6.2 Bug Fixes and Minor Enhancements

6.2.1 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.

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.

7.2 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 m³/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.

7.4 Quadtree Map Output

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 XMDF output if SMS TRIANGLES is specified.

7.5 SGS Map Output

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.

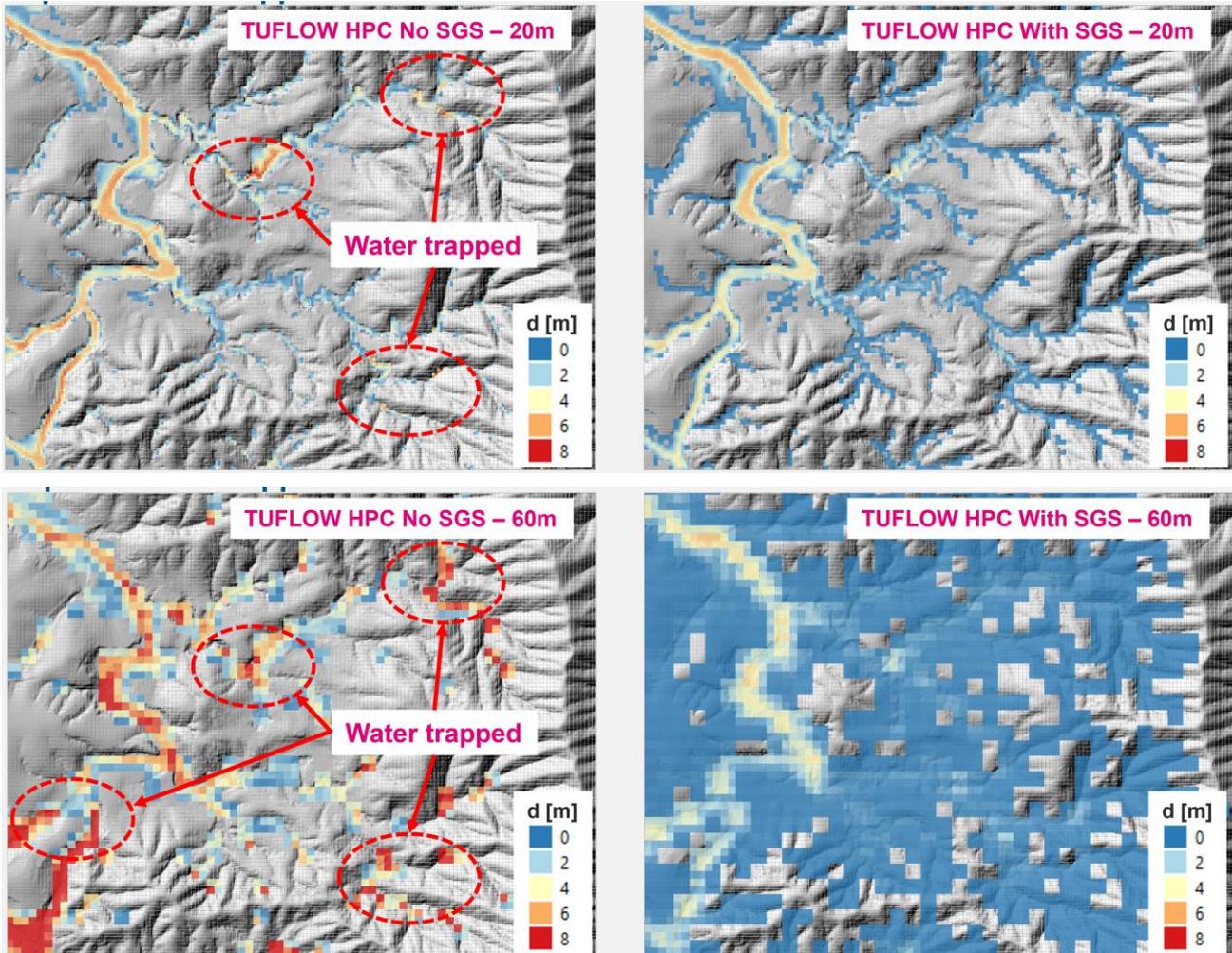


Figure 2 Effect of Using SGS on a Catchment Wide Model

7.5.1 Map Cutoff SGS == Commands

If Sub-Grid Sampling (SGS) is applied the commands below can be used to control how the `Map Cutoff Depth ==` can be used in conjunction with a partially wet cell to control when cells are shown as “dry” in the map output.

`Map Cutoff SGS == Med | <optional depth above> ! Use the median elevation as datum, an optional depth above can be specified`

`Map Cutoff SGS == Min | <optional depth above> ! Use minimum elevation, this is the same as Map Cutoff Depth, and a warning is reported if you specify both Map Cutoff Depth and Map Cutoff SGS == Min`

```
Map Cutoff SGS == Avg | <optional depth above> ! Use mean average elevation as datum, an optional depth above can be specified
```

```
Map Cutoff SGS == Perc | <Mandatory percentile> ! Define the elevation percentile to be used as a cutoff. For example Map Cutoff SGS == 25 with use the 25th percentile elevation in the cell as a map cutoff
```

For example, the following two commands:

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

```
Map Cutoff SGS == Perc | 25
```

will use the maximum of

- 0.05 above the lowest SGS sampled elevation, or
- the 25th percentile of the SGS elevations sampled within the cell

to set the water elevation at which the cell is considered “dry” for map outputs.

7.5.2 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 | MINIMUM
```

where

- EXACT 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.
- AVERAGE 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).
- 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).

```
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 depends on the [SGS Depth Interpolation Approach](#) setting as follows:

```
0.0 if SGS Depth Interpolation Approach == EXACT
```

```
1.0 if SGS Depth Interpolation Approach == AVERAGE or MINIMUM
```

.

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”.

8.2 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.

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 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.

9 Minor Enhancements and Bug Fixes

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 stream 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.

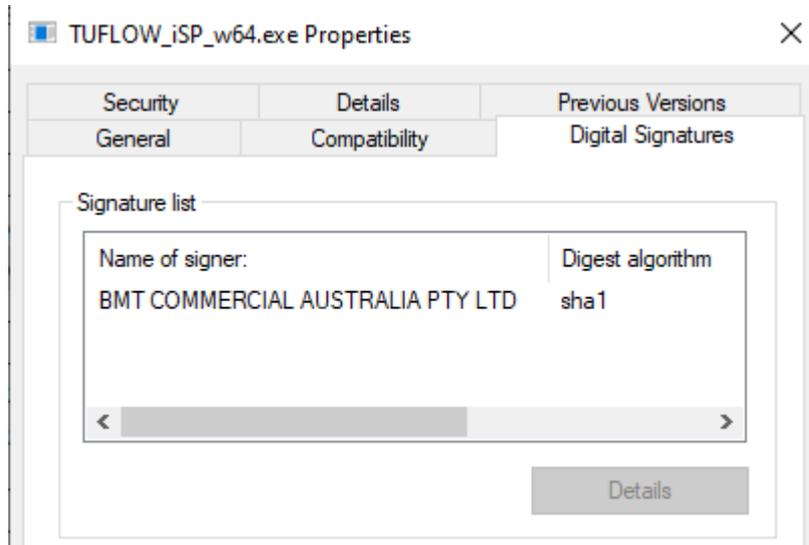
9.4 Enhancement to the QGS file

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.

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.



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.