



# TUFLOW Classic and HPC 2017-09 Release Notes

## Document Updates and Important Notices

(in reverse chronological order)

**Oct 25, 2017:** Updated for the 2017-09-AC Build, which addresses a critical bug fix if using the HPC solver with Material IL/CL and direct rainfall, and some minor enhancements. A description of the bug fix is in Item 4.7 and other changes to this document for this revision are highlighted in light purple.

**NOTE: ALL USERS MUST UPGRADE TO BUILD 2017-09-AC.**

**Sep 29, 2017:** Updated for the 2017-09-AB Build, which addresses critical bug fixes. A description of the bug fixes and other changes to this document for this revision are highlighted in light green.

Sep 7, 2017: First issued with Build 2017-09-AA, the first build for the 2017-09 release.

Note, this document will be updated from time-to-time with any new builds or new content.

## Licensing and Executable Versions

To run simulations using Build 2017-09-AA requires payment of the 2017/2018 annual software maintenance fee 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](mailto:sales@tuflow.com), or for general support [support@tuflow.com](mailto:support@tuflow.com). Use of the TUFLOW software is bound by the [TUFLOW Products Licence Agreement](#).

For the TUFLOW HPC solver, this solver is available with a standard TUFLOW licence when running on CPU hardware. To run on NVidia GPU devices a TUFLOW GPU Hardware Licence (previously referred to as the TUFLOW GPU Solver Licence) is required for each device. Refer to the [TUFLOW Price List](#) for more details.

After user consultation, 32-bit versions of TUFLOW are not being provided with the 2017 release. For the 2017 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, it is rare that the double precision version is required due to the nature of the solution scheme.

## Table of Contents

<b>Document Updates and Important Notices .....</b>	<b>1</b>
<b>Licensing and Executable Versions .....</b>	<b>1</b>
<b>1 2017-09 Release Overview.....</b>	<b>4</b>
<b>2 1D Domains.....</b>	<b>5</b>
2.1 Culverts .....	5
<b>3 2D Domains.....</b>	<b>6</b>
3.1 Check on input grid extents .....	6
3.2 Read MID (or Read RowCol) Bug Fix .....	6
<b>4 Boundaries and Links.....</b>	<b>6</b>
4.1 Boundary XF Files .....	6
4.2 Global Rainfall Boundary Multiplier .....	6
4.3 Material losses extended to gridded rainfalls .....	7
4.4 SA Cell Selection .....	7
4.5 SX Links.....	7
4.6 SA/RF HPC Critical Bug Fix – Build 2017-09-AB .....	10
4.7 Material IL/CL HPC Critical Bug Fix – Build 2017-09-AC .....	10
<b>5 Customising Output .....</b>	<b>10</b>
5.1 Map Output .....	10
5.2 Time-Series Output.....	13
<b>6 Simulation Control .....</b>	<b>14</b>
6.1 Specify start and end time via run options arguments .....	14
6.2 Simulations start-up time summary .....	14
6.3 End After Maximums now includes a monitoring start time.....	14
6.4 New Auto Terminate option .....	14
<b>7 Minor Enhancements and Bug Fixes from the 2016-03 Release.....</b>	<b>15</b>
7.1 Minor Enhancements from the 2016-03 Release .....	15
7.2 Fixed Software Bugs from the 2016-03 Release.....	16
<b>8 Licensing .....</b>	<b>17</b>
8.1 New software lock licensing options .....	17
8.2 Ability to control the dongle search order .....	17
<b>9 Backward Compatibility to the 2016-03 Release.....</b>	<b>18</b>
9.1 Defaults == PRE 2017 .....	18
9.2 Non-Backward Compatible Changes from the 2016-03 Release.....	18
9.3 New Intel Compiler .....	18
<b>10 TUFLOW HPC 2D Solver (TUFLOW GPU Mark II).....</b>	<b>19</b>

10.1	HPC 2D Solution Scheme - Description .....	19
10.2	TUFLOW HPC versus 2016-03 TUFLOW GPU .....	20
10.3	Specifying Classic or HPC 2D Solver.....	21
10.4	Hardware Specification.....	21
10.5	Timestepping and Control Numbers.....	22
10.6	Repeated Timestep Feature .....	25
10.7	Miscellaneous Items .....	26
10.8	Mass Conservation .....	27
10.9	TUFLOW HPC (as yet) Unsupported Features .....	28
<b>11</b>	<b>TUFLOW HPC Q&amp;A.....</b>	<b>30</b>
11.1	Does the HPC 2D solver produce the same results as the Classic 2D solver? .....	30
11.2	Do I need to change anything to run a Classic model in HPC? .....	30
11.3	Do I need to review or recalibrate if I switch from Classic to HPC? .....	30
11.4	Why does my HPC simulation take longer, or is not much faster, than Classic?.....	31
11.5	I'm familiar with Classic, so do I need to be aware of anything different with HPC?.....	32

## 1 2017-09 Release Overview

The 2017-09 release includes some major new features and range of enhancements. The most significant of these are discussed below.

New HPC solver (major rework of TUFLOW GPU). Changes include:

- Ability to run on CPU or GPU.
- New cell schematisation (to be consistent with Classic and use of cell side elevations and sampling).
- Enhanced 1<sup>st</sup> order and new 2<sup>nd</sup> order spatial schemes.
- Full 1D/2D linking using the same SX and HX methods as TUFLOW Classic, giving access to all 1D functionality.
- Significantly better scaling when running one simulation over multiple GPU devices.
- Inclusion of more TUFLOW Classic functionality (e.g. layered flow constrictions, SA Trigger boundaries, additional outputs).
- Previous “GPU” solver available for legacy models.

Other significant enhancements are much improved stability and stability options for SX connections to large 1D structures for Classic and HPC, along with numerous enhancements and new functionality in TUFLOW for both Classic and HPC solvers as discussed within 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).

## 2 1D Domains

### 2.1 Culverts

#### 2.1.1 Matrix Blockage methodology now supports AEP nomenclature

AEP as well as ARI nomenclature now supported. To set the return period naming convention to ARI or AEP a new .tcf command is available:

```
Blockage Return Period == [ AEP | ARI ]
```

If this command above is not set, the first occurrence of either “`Blockage ARI ==`” or “`Blockage AEP ==`” sets the return period naming convention. To set the return period of the current event, one of the .tcf commands below are used (typically in an event file). Note that only one of AEP or ARI can be used for a simulation.

```
Blockage ARI == <ARI Value>
```

```
Blockage AEP == <AEP Value>
```

The return period values in the first column of the matrix file specified with the .tcf command “`Blockage Matrix File == <file>`” must be in the same convention. For example, if specifying “`Blockage Return Period == AEP`”, then values in the matrix file must also be specified in AEP.

If interpolation between return periods is required, this is linearly interpolated between events using the ARI. For example, if “`Blockage ARI == 200`” for the table below (which is between the ARI 100 and 2,000), for a culvert with blockage “Category B”, the value applied will be linearly interpolated between a blockage of 10% and 20%.

Example Matrix Blockage File (in ARI):

ARI	Category A	Category B	Category C	Category D	Category E
1	0	10	10	0	100
20	0	10	10	20	100
50	0	10	20	50	100
100	0	10	20	50	100
2000	0	20	50	70	100
10000	0	50	70	100	100

Refer to the TUFLOW manual for more details on setting up Matrix Blockage.

#### 2.1.2 Matrix Blockage extended for Energy Loss Method

The Energy Loss Method (ELM) for structure blockage detailed in Stage 1 report of Project 11 of [Australian Rainfall and Runoff](#) is only valid for outlet controlled flow regimes. As detailed in the 2016 TUFLOW Manual, for inlet control regimes no additional loss or reduction in area is applied in the 2016 release of TUFLOW. For the 2017 release, if using the ELM approach and the culvert is flowing in an inlet controlled flow regime, the culvert flow area is adjusted as per the Reduced Area Method (RAM). This reduced area is output to the relevant check and results files.

## 3 2D Domains

### 3.1 Check on input grid extents

By default the input grid extents for any grids read using the “Read Grid <Grid Type> ==” command are compared with the TUFLOW model extent and if the input GIS raster is entirely outside of the model area these are skipped to reduce the simulation start-up time. This can be particularly useful if the DEM data is across many tiles, of which only some tiles fall within the 2D domain. In other words, the .tgc file can reference all tiles (eg. for the whole of the UK), but only those required for the 2D domain are accessed.

To process all grids, regardless of the grid extents, the following .tcf file command can be included.

```
Process All Grids == ON
```

Processing of all input grids is set to “ON” if the previous defaults are used, e.g. “`Defaults == Pre 2017`”.

### 3.2 Read MID (or Read RowCol) Bug Fix

Build 2017-09-AA may experience an ERROR 2091 when it reaches the end of a .mid file being read by a .tgc “Read MID” (or “Read RowCol”) command. This error is due to a known Fortran v17 compiler issue that does not correctly detect the end of a file, and unfortunately occurs at random. Build 2017-09-AB corrects this issue for “Read MID” using an alternative approach. The occurrence of a failed end of file detection was rectified for other commands in 2017-09-AA, however, it seems that not all instances were captured due to the nature of the problem, so please let us know if you have a similar issue! This issue does not occur for TUFLOW builds prior to the 2017-09 release as these builds used a different Fortran compiler version that does not have this issue.

## 4 Boundaries and Links

### 4.1 Boundary XF Files

XF files are now used to speed up the reading of boundary data in both .csv and .ts1 file formats. This works for data referenced from the “BC Database ==” and “Pit Inlet Database ==”. As with other .xf files created by TUFLOW (for example “Read Grid Zpts ==”), the save date of the .xf file is compared to the save date of the original data file (.csv or .ts1). If the original dataset has been modified since the .xf file was created the original dataset is re-read and the .xf file regenerated. XF files can be turned off globally with the .tcf command “`XF Files == OFF`”, or just for the new boundary/pit database files using “`XF Files Boundaries == OFF`”.

### 4.2 Global Rainfall Boundary Multiplier

New .tcf command “`Rainfall Boundary Factor == <value {1.0}>`” sets a global multiplication factor for all rainfall boundaries including:

- Global Rainfall
- Read GIS Rainfall
- Read GIS SA RF

- Gridded rainfalls

As per previous TUFLOW versions map output of the applied rainfall depths and rainfall rate are output using the RFC (cumulative rainfall) and RFR (rainfall rate) “Map Output Data Types ==”.

#### 4.3 Material losses extended to gridded rainfalls

Material IL and CL are now applied to gridded rainfall. A new map output of rainfall losses applied based on the losses in the material file is available see Item 5.1.6 below.

Please also note the critical bug fix described in Item 4.7.

#### 4.4 SA Cell Selection

An SA region GIS object selects cells based on the TUFLOW cell centre (ZC) being within the GIS region object. If an SA region selects no cells (this can occur if the polygon is small compared to the TUFLOW cell size), the polygon centroid is used to select the 2D cell that the GIS region’s centroid falls within. If the selected cell is within the model domain and active this cell is used for the inflow. This change now guarantees that at least one cell is selected by an SA region (previously a small SA region may not select any cells), and allows an SA inflow region to be used for a small inflow (i.e. on a single cell) without cell size dependency of the GIS layer.

#### 4.5 SX Links

##### 4.5.1 SX Flow Distribution Cutoff Depth

In rare situations, flow exchange may occur on “dry” cells at SX 1D/2D locations. This occurs because of a numerical precision issue relating to the optimised compiling of the source code. This has been corrected by introducing a shallow depth cutoff for SX cells. The default value has been set to 1.5 times the 2D domain’s wet/dry depth (ie.  $1.5 \times 0.002\text{ m}$  or 3 mm by default). The approach and the cutoff depth are controlled by the .tcf command:

```
SX Flow Distribution Cutoff Depth == [ {AUTO} | <cutoff_depth> ]
```

The default AUTO option sets the cutoff depth to the 1.5 times the wet/dry depth of the 2D domain. The 2016-03 release defaulted to a cutoff\_depth of 0.005 m (5 mm). For legacy models use “`Defaults == PRE 2017`” or “`SX Flow Distribution Cutoff Depth == 0.005`” for backward compatibility with 2016-03. Prior to 2016-03, this feature was not available.

If the cutoff\_depth is non-zero ( $> 0.0001$ ), the allocation of the 1D flow across SX cells is proportioned by depth, i.e. the deeper SX cells are assigned a greater proportion of the flow.

##### 4.5.2 SX Head Distribution Cutoff Depth

At an SX link, the water level sent through to the 1D node is based on the water levels of the wet SX cells. Prior to the 2017 release, this level was simply the average of the wet cells. This approach can cause issues where there are a slightly wet SX cells that are elevated above the SX cells within the main flow path. This primarily occurs if using direct rainfall, or if there is some side flow cascading down the higher SX cells.

For the 2017 release, the default is to calculate the 1D water level based on proportioning to the SX cell depth in a similar manner to how flow is distributed to SX cells. This means that the water level transferred to the 1D node is biased to that of the deeper cells.

The approach taken is controlled by the new .tcf command:

```
SX Head Distribution Cutoff Depth == [ {AUTO} | <cutoff_depth> ]
```

If the cutoff\_depth value is less than 0.0001 there is no proportioning to depth and the approach taken prior to the 2017-09 release is adopted (therefore set cutoff\_depth to 0.0 for backward compatibility). The default AUTO option sets the cutoff\_depth value to 1.5 times the wet/dry depth, and therefore defaults to 0.003 m (3 mm). If different 2D domains have different wet/dry depths, the cutoff\_depth varies for each domain accordingly. Setting cutoff\_depth to any value greater than 0.0001 enforces the specified cutoff\_depth across all 2D domains.

If cutoff\_depth is set to AUTO or is greater than 0.0001, any wet SX cells with a depth less than cutoff\_depth is excluded, and the 1D water level is calculated as a depth weighted average of the remaining wet SX cells. This may change results very slightly, therefore for legacy models use “`Defaults == PRE 2017`” or “`SX Flow Distribution Cutoff Depth == 0.0`” for backward compatibility.

#### 4.5.3 SX Storage Factor

**Note:** Build 2017-09-AB fixes a bug and provides an enhancement for the SX Storage feature as follows:

- Where more than one SX connection occurs for the same 2D cell and the SX connections are from different GIS layers, the adjustment of the SX storage was incorrect. This has been corrected.
- For large models with many SX connections, the start-up time could be excessively long whilst “Checking storage reduction factors...”. This has been corrected.

The storage of 2D SX cells are now adjusted to include the storage of the connected 1D node (1D pits excluded). 1D nodes connected via a 2D SX connection have a 1D HX link, and therefore as this link is a type of water level (H) boundary the storage of the node is not used or accounted for within the 1D calculations, as there is no need to apply the mass balance equation (the water level is pre-defined). Usually this storage is negligible to the overall model storage and is inconsequential, however, some provision for this storage has found to be beneficial in stabilising SX links, particularly where the 1D structure is large in comparison to the 2D cell size – an increasingly common situation as 2D cell sizes become smaller with faster computing hardware. The additional storage at the SX cells is applied by adjusting the SRF (Storage Reduction Factor) value of the cell (a negative SRF increases storage) – refer to the Storage Reduction (2d\_srf) section in the TUFLOW Manual.

As of the 2017-09 release the default is to distribute the average 1D node storage across connected SX cells. Varying the 2D SX cell surface area over time based on the 1D nodal area was trialled, but was not adopted as mass is created or lost as the SX cell’s surface area is increased or reduced.

Should an SX link continue to be unstable, the 1D storage allocated to the connected SX cell(s) can be further increased using the 2d\_bc “a” attribute. If the “a” attribute is greater than zero, it is used

as a multiplier of the additional storage from the 1D node. For example, specifying an “a” value of 2.0 doubles the additional storage applied. Note that an “a” value of 0.0 assumes a multiplier of 1.0. Should you wish to not apply the additional SX storage to an SX connection, specify an “a” value of -1. Testing has shown small artificial increases in storage usually have immeasurable effect on results, however, sensitivity testing is recommended to confirm this assumption. For example, if specifying an “a” value of 2 stabilises the link, sensitivity test by increasing “a” further to 3 or 4, and if there is no appreciable or unacceptable change in results, increasing the storage by a factor of 2 can be considered to have negligible effect on results.

Two other .tcf commands also affect how the SX cell storage is adjusted as follows:

`“SX Storage Factor == [ {1.0, 20.0} | <sxs_factor>, <sxs_limit> ]”`. The first value, `<sxs_factor>`, is a multiplier that globally adjusts all additional storages applied to SX cells. As discussed above, if using a `<sxs_factor>` value greater than 1.0, sensitivity testing is recommended. An `<sxs_factor>` value of zero effectively applies no additional storage to any SX cells. The second, optional value, sets the limit by which an SX cell can have its storage increased in addition to its own storage. By default, this is a factor of 20, in which case, if a 1D node’s storage increases an SX cell’s additional storage by more than 20, the factor is limited to 20.

The adjusted SRF value at SX cells can be reviewed by checking the SRF value in the 2d\_grd\_check layer. Note that the SRF value is how much to reduce the storage by, therefore additional storage applied to SX cells appears as a negative number.

To revert to the approach prior to the 2017-09 release specify the .tcf command `“SX Storage Approach == Cell Only”` or `“Defaults == PRE 2017”`.

**Note:** A 1D pit SX connection does not adjust the SX cell’s storage as the storage of the pit is allocated to the 1D node automatically generated on the downstream side of the created pit channel.

**Note:** SX cells as of the 2017-09 release are now not adjusted by SRF commands (e.g. `“Read GIS SRF ==”`). Previously, an SRF command would adjust the SX cell’s storage.

#### 4.5.4 SX Boundary Regions/Polygons

An SX region object can now be digitised in a 2d\_bc layer for connecting SX cells as an alternative to points and lines. All cells with a cell centre within the region/polygon are set as SX cells. A CN line is required to connect a 1D node and must snap to any vertex on the perimeter of the SX region object.

Prior to the 2017-09 release, a region object was treated as point object using the region’s centroid as the point location. No backward compatibility is provided for this approach as the recommended approach for SX connections prior to the 2017-09 release was to use points or lines.

If the “Z” flag is specified for a region SX object, as well as lowering the ZC (cell centre) as per points and lines, the cell sides and cell corners are also lowered to match the elevation on the connected 1D node. Any elevation lowered by an SX region object will be reported with message 2118.

#### 4.6 SA/RF HPC Critical Bug Fix – Build 2017-09-AB

**Note:** Build 2017-09-AB fixes a critical bug in Build 2017-09-AA affecting the application of SA and RF source inflows to the HPC 2D Solver. If a 2D cell was allocated more than two SA/RF boundaries, only the last two would be applied if using TUFLOW HPC. This has been fixed in Build 2017-09-AB. This bug does not occur if using the TUFLOW Classic solver.

#### 4.7 Material IL/CL HPC Critical Bug Fix – Build 2017-09-AC

**Note:** Build 2017-09-AC fixes a critical bug in prior 2017-09 builds affecting the application of material IL and CL rates for direct rainfall boundaries if using the new TUFLOW HPC solver. The incorrect rainfall loss rates can be ascertained via the new RFML CML map output (see Section 5.1.6). This bug does not occur if using the TUFLOW Classic solver.

### 5 Customising Output

#### 5.1 Map Output

##### 5.1.1 New Map Output Formats

A cell centred map output format has been added and is specified using the CC option for “Map Output Format ==”. The output uses the [NetCDF](#) file format, and uses the 2D domain’s cell size and orientation. This feature is only compatible with single 2D domain models and does not include any 1D output via WLLs (the standard “NC” “Map Output Format ==” option is analogous to the ASC or FLT output that outputs on a north-south grid of a resolution controlled by the user and includes any 1D output interpolated from the WLL triangulation). The CC option allows for rotated TUFLOW model grids to be output at the 2D cell size resolution without interpolation to a north-south aligned raster. The file format of the output is described on the [TUFLOW Wiki](#).

##### 5.1.2 TMO (12D) extended to include additional output types

The TMO file format now supports all “Map Output Data Types ==”. Previously, output was limited to water levels, depths and velocity (magnitude and direction).

##### 5.1.3 Time of maximum map outputs extended to include all hazards outputs, bed shear stress, stream power and energy

If tracking of maximums is enabled (the default), the time of maximums can now be tracked for any specified hazard outputs, as well as Bed Shear Stress (BSS), Stream Power (SP) and Energy (E). A new .tcf command is available to control this behaviour. If set to OFF (the default), the maximum values are tracked, but not the time of maximum. This .tcf command does not apply to water level or velocity output; time of maximum H and V continues to be tracked (as per previous releases).

```
Maximums Track Time == [ {OFF} | ON ]
```

##### 5.1.4 New tcf command “**Maximums Start Track Time == <time in hours>**”

If tracking of maximums is enabled (the default), a start time for tracking the maximums can now be specified. For example, if a model starts at 0 hours, and the user only wants to track maximums

after 24 hours of model warmup the .tcf command “`Maximums Start Track Time == 24`” can be added to the .tcf file.

If this command is omitted, TUFLOW defaults to:

- The time of first map output if temporal map output for the simulation is specified.
- If no temporal output is specified, the maximums are tracked from the start of the simulation.

#### 5.1.5 Change to output extents for Time Output Cutoff

For Time Output Cutoff Depth or Time Output Cutoff Hazard mapping, the extent of the mapping is trimmed. Previously the mapped extent was based on the maximum inundation extent for the entire simulation, with areas that did not exceed the cutoff threshold mapped with value of 0.0. Only cells that have inundation above the threshold are now mapped.

#### 5.1.6 New map output data type “RFML”

If “RFML” is specified as a “Map Output Data Types ==”, the material based rainfall losses are output. The output contains the total rainfall losses applied due the initial and continuing rainfall losses specified in the “Read Materials File ==” (.tmf or .csv) file. The RFML option can be used to track the rainfall based material losses that have been applied spatially. The RFC and RFR map output data types can be used to output the cumulative rainfall and rainfall rate.

Please also note the critical bug fix described in Item 4.7.

#### 5.1.7 New UK Hazard output data types “ZUK2” and “ZUK3”

The existing ZUK0 and ZUK1 are based on the UK FD2321 technical report. Two additional hazards ZUK2 and ZUK3 are available based on the FD2320 technical report.

ZUK2 is the flood hazard rating based on the equation below.

$$\text{Flood Hazard Rating} = ((v + 0.5) * D) + DF$$

Where:

v = velocity (m/s)

D = depth (m)

DF = debris factor

For this intermediate approach a precautionary approach has been adopted and a debris factor of 0.5 has been used for depths below and equal to 0.25 m and a debris factor of 1.0 has been used for depths above 0.25 m. These are conservative estimates based on an urban environment, as defined in the Flood Risks to People project. Based on this, the hazard rating equation has been applied to various combinations of flood depth and velocity to produce a matrix of hazard ratings. Applying thresholds to these hazard ratings defines the danger to people at various depths and velocities as shown in Table 13.1.

ZUK3 is degree of flood hazard based on the criteria below where 1, 2, 3 and 4 are output for the low, moderate, significant and extreme hazards respectively.

Thresholds for Flood Hazard Rating $H = d \times (v + 0.5) + DF$		Degree of Flood Hazard	Description
FD2321	FD2320		
<0.75	<0.75	Low	<b>Caution</b> - "Flood zone with shallow flowing water or deep standing water"
0.75 - 1.25	0.75 - 1.25	Moderate	<b>Dangerous for some (i.e. children)</b> - "Danger: Flood zone with deep or fast flowing water"
1.25 - 2.5	1.25 - 2.0	Significant	<b>Dangerous for most people</b> - "Danger: flood zone with deep fast flowing water"
>2.5	>2.0	Extreme	<b>Dangerous for all</b> - "Extreme danger: flood zone with deep fast flowing water"

It is possible to specify both FD2320 and FD2321 based hazard outputs, e.g.:

```
Map Output Data Types == h v q d ZUK0 ZUK1 ZUK2 ZUK3
```

#### 5.1.8 Start date for Blue Kenuue output can be specified

This date is added to the Blue Kenuue output files as per the Blue Kenuue file format. This is specified with the .tcf command “`Blue Kenuue Start Date == <Date in isodate format>`”. The date should be specified in ISO 8601 (isodate) format (yyyy-mm-dd). For example, to specify the 25<sup>th</sup> of September 2016 the .tcf command would be:

```
Blue Kenuue Start Date == 2016-09-25
```

#### 5.1.9 Manning's n, BSS and SP map output now available from HPC solver

For the models with depth varying Manning's n values, the Manning's n computed in the HPC solver was not available for map output, and BSS/SP temporal map output was incorrect. For Build 2017-09-AB, if depth varying Manning's n values are utilised in a model and map output data types (n, BSS or SP) are requested, at each map output interval the computed n values are transferred from the HPC solver allowing Manning's n map output to be written, and BSS and SP output to be correct.

#### 5.1.10 Maximum Bed Shear Stress map output now available from HPC solver

Build 2017-09-AB now tracks the maximum bed shear stress (Map Output Data Type == BSS) for HPC simulations.

#### 5.1.11 Bug fix for NetCDF Vector Output - Build 2017-09-AB

Build 2017-09-AB fixes a bug in Build 2017-09-AA for vector outputs in NetCDF (Map Output Data Format == NC). For the NetCDF raster output, vector datasets are written as two components magnitude and direction. The bug caused the vector direction to overwrite the magnitude.

#### 5.1.12 Bug fix for TMO velocity vector output - Build 2017-09-AB

Build 2017-09-AB fixes a bug in Build 2017-09-AA for velocity vector output in the 12D TMO output format (Map Output Data Format == TMO). This caused the velocity direction dataset to be written incorrectly preventing it from loading into 12D.

### 5.1.13 BSS and SP map output in English Units

BSS and SP for English Units is now output as Pound Force per square foot (lbf/ft<sup>2</sup>) for the 2017-09 release. Previously, it was output as Poundals per square foot.

## 5.2 Time-Series Output

### 5.2.1 Support for plot output regions

**Note:** Build 2017-09-AB fixes a bug in Build 2017-09-AA for 2d\_po.mif layers containing a region. If the layer contains a mix of regions and lines and/or points, any 2d\_po lines and points after the first region were ignored. This has been fixed in Build 2017-09-AB.

Regions are now recognised for 2d\_po objects. Valid output types are outlined in the table below.

**Table of Valid PO Region Data Types**

Type Identifier	Description	Units
HAvg	Average water level (wet cells only)	Metres or Feet
HMax	Maximum water level	Metres or feet
Qin	Flow into the region	m <sup>3</sup> /s or ft <sup>3</sup> /s
QOut	Flow out of the region	m <sup>3</sup> /s or ft <sup>3</sup> /s
SS	Sink / Source flows applied within the region (rainfall, infiltration, source area inflow and SX flows)	m <sup>3</sup> /s or ft <sup>3</sup> /s
Vol	Total volume within the region	m <sup>3</sup> or ft <sup>3</sup>

### 5.2.2 HPC Maximums / Minimums extracted from temporal data

For the HPC solver the maximum/minimum values for plot outputs are post-processed at the end of the simulation based on the “Time Series Output Interval ==” (not every timestep). Previously (for the TUFLOW GPU solver) these were output as 0.0 in the CSV and GIS outputs.

Note: Due to splitting up of a 2D model across multiple cores for TUFLOW HPC, the tracking of time series output maximums is not yet performed every timestep. Options to offer this functionality is being investigated.

### 5.2.3 User specified null value for plot output

New .tcf command:

```
Time Series Null Value == <null value> | {Cell Elevation}
```

For water level outputs a user specified null value is available if all cells are dry. For example, “Time Series Null Value == -9999” outputs this value in the .csv files if the cell at the PO point is dry, or all cells along a PO line are dry. For a PO line, the default is to use the lowest cell elevation, consistent with previous TUFLOW versions.

For other PO outputs, e.g. flow, flow area, velocity, a value of 0.0 is output if all cells are dry.

#### 5.2.4 WARNING 2368 for 2d\_po Lines for TUFLOW HPC

Lines and polylines should only be used for 2d\_po Q output types, and not for other output types such as H, for which TUFLOW HPC is not yet set up to handle. A WARNING 2368 is issued if this is the case.

#### 5.2.5 Bug Fix: Not Closing PO\_R.shp Layer if “Write PO Online == ON”

Build 2017-09-AC fixes a bug that doesn't close off the new PO\_R.shp layer if “Write PO Online == ON”.

## 6 Simulation Control

### 6.1 Specify start and end time via run options arguments

The start time and/or end time for a simulation can be specified using the run option -st<time\_in\_hours>, and the end time using -et<end\_time\_in\_hours>. Any start and end time specified via the run option arguments are given the highest priority and override the “Start Time ==” and “End Time ==” settings in the .tcf, event files (.tef) and override files.

### 6.2 Simulations start-up time summary

A new output file is created in the log folder with the name “<simulation\_name>\_start\_stats.txt”. This file contains information on the total time and the time elapsed for each stage of model initialisation. This can be used to identify the stages causing slow simulation start up, it also allows the TUFLOW development team to prioritise development tasks to remove bottlenecks in startup through the use of XF files and other enhancements. If you have a problematic (slow starting) model, please email this file and corresponding .tlf file through to [support@tuflow.com](mailto:support@tuflow.com).

### 6.3 End After Maximums now includes a monitoring start time

The end after maximum feature has been enhanced to not commence monitoring until after a specified time by using the following .tcf command:

`End After Maximum Start Time == <time in hours>`

For example, “`End After Maximum Start Time == 12`” does not commence monitoring till a simulation time of 12 hours, therefore, the simulation cannot terminate prior to this time.

### 6.4 New Auto Terminate option

A new option for stopping simulations after the flood peak. The 2D cells that are monitored are controlled by specifying a value of 0 (exclude) or 1 (include) using the .tcf commands:

`Set Auto Terminate == <AT value>`

`Read GIS Auto Terminate == <GIS file containing polygon / region objects>`

For example, in the below, all cells are first set to be excluded for monitoring followed by the reading of a GIS layer to set cells individually.

```
Set Auto Terminate == 0  
Read GIS Auto Terminate == ..\model\gis\2d_AT_001_R.shp
```

At each “Map Output Interval ==” the monitored cells are compared against two criteria:

- The percentage of the wet cells that have become wet since the last map output interval.
- The velocity-depth product at the current timestep compared to the maximum.

For the percentage of cells that have become wet since the last interval, the maximum allowable value is controlled with the .tcf command:

```
Auto Terminate Wet Cell Tolerance == <maximum_allowable_%_of_newly_wet_cells>
```

If set to 0, then if any monitored cells have become wet since the last map output the simulation continues. If set, for example, to a value of 5, then up to 5% of monitored cells can become wet since the last map output without triggering an auto terminate.

For the velocity-depth tolerance, at each output interval the velocity depth product is compared to the tracked maximum value. If the current dV product is within the specified tolerance using “`Auto Terminate dV Value Tolerance == <tolerance value>`” of the tracked maximum, the cell is within the range. The total number of cells that are allowable within the specified range is controlled with .tcf command:

```
Auto Terminate dV Cell Tolerance == <%_of_cells_within_dV_tol_range>
```

If set to a value of 1, then up to 1% of monitored cells can be within the tolerance value without triggering an auto terminate. The larger the “Auto Terminate dV Value Tolerance ==”, the further the dV product needs to have dropped from the peak value.

The time that the auto terminate feature commences can be controlled using the .tcf command “`Auto Terminate Start Time == <time_in_hours>`”, otherwise the “`Start Time ==`” is used.

Note, this option is only assessed at every “`Map Output Interval ==`”.

## 7 Minor Enhancements and Bug Fixes from the 2016-03 Release

### 7.1 Minor Enhancements from the 2016-03 Release

#### 7.1.1 Z Shape Route Name Check

When a Z Shape Route with the same name as an existing Z Shape Route is found, the route is extended, for the 2017 release additional checks are performed that the route cut off type matches that of the existing route. If the Z Shape Route types are different ERROR 2409 is generated.

#### 7.1.2 Extra empty (template) GIS files are now output

- 1d\_nwkb – 1D network layer with the “pBlockage” attribute set to a character field rather than numeric field for use with matrix culvert blockages.
- 2d\_sa\_tr – 2D source area layer with additional attributes required for the SA Trigger option.

## 7.2 Fixed Software Bugs from the 2016-03 Release

### 7.2.1 “SXL” incorrectly applied to nodes – L option only applicable to pits

If “SXL” type is specified on a 1d\_nwk **node** an ERROR 1214 is now returned. The SXL option is only compatible with 1D **pits**. Previously this would cause the 2D cell to be lowered significantly (causing WARNING 2118 to be issued).

**Note:** Build 2017-09-AB fixes a bug in Build 2017-09-AA that causes ERROR 1214 described above to be incorrectly returned for SXL connections on “W” weir type pits. This has been fixed in Build 2017-09-AB.

### 7.2.2 SA Trigger option now works correctly if “Read GIS Streams ==” specified

SA Trigger option now works correctly if “Read GIS Streams ==” has been specified. Previously the Stream option would overwrite the Trigger and the SA inflow would occur at the start of the simulation.

### 7.2.3 Blue Kenue output now works with DP version

Blue Kenue output now works with double precision (DP) versions of TUFLOW. Previously any Blue Kenue files generated from DP versions of TUFLOW were incompatible with BlueKenue.

### 7.2.4 Structure group check files causing crash

Fixed issue with structure group check files that occurs if one or more 1D channel was reversed. This would cause a crash when writing the check file. This did not affect the results, which correctly handled the reversed channel. Structure group check files (\_Str\_Grp\_All.csv and \_Str\_Grp\_Multi.csv) now list whether the channel is reversed.

### 7.2.5 Grid maximums not correct inside an output zone

Fixed issue with grid maximums inside an output zone if no grids or cell centred outputs were set for the entire model. This would cause a checker boarding in the tracked maximum velocity and hazard outputs.

### 7.2.6 Maximum outputs zero if no temporal requested

Fixed issue with maximum outputs if no temporal output was requested for the simulation, previously the maximum outputs were all 0.0 if no temporal output was requested.

### 7.2.7 Summary Output Interval in .tsf file now correct for M2D models

Fixed bug with the Summary Output Interval in the .tsf file for multiple 2D domain models

## 8 Licensing

### 8.1 New software lock licensing options

Software locks utilise the same WIBU licensing system as the currently issued TUFLOW dongles, and are available in the same local and network configurations as the physical dongles. Please refer to the [TUFLOW Price List](#) for more information on software locks. Software locks are tied to a machine and do not require a hardware lock (dongle).

Both BMT WBM and CH2M can issue software locks.

### 8.2 Ability to control the dongle search order

With the addition of software locks, there are five licence types / vendors:

- BMT WBM physical dongle
- BMT WBM software licence
- Aquaveo (SMS) physical dongle
- CH2M physical dongle
- CH2M software licence

With numerous licencing options available, setting the preferred licence type can speed simulation startup. The licence search order can be set via a licence control file “TUFLOW\_licence\_settings.lcf”, this replaces the TUFLOW\_Dongle\_Settings.dcf. Note that this file can occur in several locations. When looking for a licence setting file TUFLOW searches in the following locations:

- A “TUFLOW\_licence\_settings.lcf” in the same location as the TUFLOW executable. This is given the highest priority.
- C:\BMT\_WBM\TUFLOW\_Licence\_Settings.lcf
- C:\BMT\_WBM\TUFLOW\_Dongle\_Settings.dcf

If no licence settings files are found, TUFLOW defaults to the order listed above. To control the search order in the TUFLOW\_licence\_settings.lcf the command specified is “[WIBU Firm Code Search Order == <list of firm codes>](#)”.

The firm codes used below are unique identifiers used by Wibu.

Vendor / Licence Description	Wibu Firm Code
BMT WBM physical dongle	101139
BMT WBM software licence	6000224
Aquaveo physical licence	101394
CH2M physical licence	101987
CH2M software licence	5000219

If multiple firmcodes are to be used these are entered with a space between the values. For example, in the below, this sets the search order to BMT WBM software lock licences, then BMT WBM hardware lock (dongle) licences.

**WIBU Firm Code Search Order == 6000224 101139**

Any firm codes not listed are not used. For example to search for only an Aquaveo hardware lock licence, the .lcf command would be:

**WIBU Firm Code Search Order == 101394**

## **9 Backward Compatibility to the 2016-03 Release**

### **9.1 Defaults == PRE 2017**

“Defaults == PRE 2017” can be used to set the defaults to those used for the 2016-03 release. The changed settings are as provided below:

- SX Flow Distribution == AUTO (previously 0.005 m)
- SX Head Distribution == AUTO (previously average level of all wet cells used – specify 0.0 to invoke)
- New SX Storage feature takes into account previously unaccounted for 1D nodal storage. To revert to 2016-03 approach use “SX Storage Approach == CELL ONLY”.
- XF files of boundary .csv and .ts1 files automatically created. To not create or use (as per 2016-03 release) specify “XF Files Boundaries == OFF”.

### **9.2 Non-Backward Compatible Changes from the 2016-03 Release**

Changes for which there is presently no backward compatible switches are listed below. Backward compatible commands can be provided if necessary for some of these items.

- Regions in 2d\_bc layers now applied as regions (previously only cell over region centroid selected).
- Material IL and CL now applied to gridded rainfall (previously not applied). Please note the critical bug fix described in Item 4.7.
- SA regions now always select a 2D cell even if there are no cell centres falling within the region (previously a SA region would not select any cell if no cell centres fell within the region).
- If using “Reveal 1D Nodes == ON”, “Time Series Output Interval ==” must be specified.

### **9.3 New Intel Compiler**

The 2017 release is compiled using the latest Intel Fortran and C++ compilers. The 2010 through to the 2016 versions of TUFLOW were compiled with the Intel Fortran Version 11 compiler.

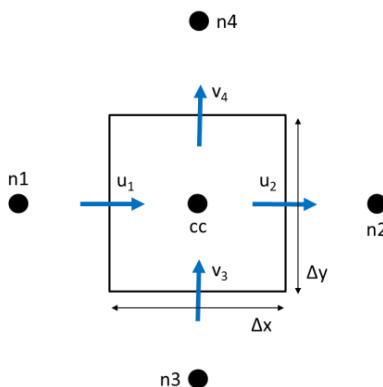
Due to upgrade of the compilers, slightly different results (usually fractions of a mm) compared with the 2016 release will occur over the model if using the same default settings. Whilst this is of no major concern, this effect can cause a cell on the edge of the floodplain to be wet in one version and dry in another, so more pronounced effects may occur around the perimeter of the flooded area.

## 10 TUFLOW HPC 2D Solver (TUFLOW GPU Mark II)

### 10.1 HPC 2D Solution Scheme - Description

The TUFLOW HPC solver resolves the 2D Shallow Water Equations (SWE) on the same uniform Cartesian grid as used by TUFLOW Classic, using a finite volume scheme. Water depth/level is calculated at the cell centres, and velocity components at the cell mid-sides or faces in the same manner as TUFLOW Classic.

The finite volume scheme applies the conservation of mass over the cell for calculating the rate of change of cell depth. The cell centre (for the cell in question) is given the notation cc, while the surrounding neighbours are given the notation n1..n4. The u velocity at the left and right faces are notated  $u_1$  and  $u_2$ , and the v velocities at the bottom and top faces are notated  $v_3$  and  $v_4$ . The cell width and height are  $\Delta x$  and  $\Delta y$  respectively. The time rate of change for the cell averaged depth is shown in Equation 1.



$$\frac{A\partial h}{\partial t} = \Phi_1 - \Phi_2 + \Phi_3 - \Phi_4 + S_Q \quad (1)$$

The volume fluxes across the four cell sides and the net volume from source boundaries determine the rate of volume change and the change in depth. Source boundaries include SA, ST and RF boundaries, soil infiltration, evaporation, and any flow linkages to 1D elements via SX links. By computing the face fluxes for all model faces, and referencing these when computing the depth derivative for each cell, volume conservation is guaranteed to numerical precision.

The calculation of the cell side volume fluxes is available in either 1<sup>st</sup> or 2<sup>nd</sup> order spatially. For the 1<sup>st</sup> order scheme, this uses depth of the upstream cell (often referred to as upwinding), bounded to be greater than or equal to 0, and less than or equal to the surface elevation of the upstream cell less the bed elevation at the cell side mid-point. For the 2<sup>nd</sup> order scheme the depth at the face is computed as the average of the two cell averaged depths, however, this method in its simplest form is not total variation diminishing (TVD) and is known to be unstable. A hybrid method is implemented in which the depth at the cell face transitions from interpolated depth, in the limit of a smoothing varying solution, to the upstream depth (1<sup>st</sup> order upwinding) when the solution shows short scale reversal or upstream controlled supercritical flow.

The solution of the cell side fluxes includes the inertia and sub-grid scale turbulence (eddy or kinematic turbulent viscosity) term. The same options for the eddy viscosity coefficient calculation in Classic (constant, Smagorinsky or a combination of constant and Smagorinsky) are available,

and the default values for both schemes are presently set the same. BMT WBM are doing further research in this area, as there is some dependency on cell size for this term (refer to the IAHR paper referenced below).

The face fluxes may also be factored down by flow constriction factors where sub-grid-scale interfering geometries exist.

The calculation steps are highly independent. The calculation of flux for one cell face may be performed independently of the other faces, and likewise the summation of flux for each cell volume may be performed independently of the other cell volumes. Applying the same algorithm to millions of data elements is ideally suited to modern multi-core CPUs, and particularly suited to GPU hardware acceleration.

The 1<sup>st</sup> order approach can experience numerical diffusion like all 1<sup>st</sup> order schemes, and does not resolve strongly two-dimensional hydraulics (eg. flow expansion downstream of a constriction) as well as a 2<sup>nd</sup> order solution. The 2<sup>nd</sup> order solution demonstrates no discernible numerical diffusion, and resolves complex 2D hydraulics, including hydraulic jumps as demonstrated using the UK EA Benchmark Flume Test 6A in the IAHR paper referenced below.

For further details on the scheme, refer to the IAHR 2017 paper by Collecutt and Syme, "[Experimental Benchmarking of Mesh Size and Time-Step Convergence for a 1<sup>st</sup> and 2<sup>nd</sup> Order SWE Finite Volume Scheme](#)". Note that at the time this paper was written, the scheme utilised cell centre definitions for velocity, which was prone to a zero-energy 'checker-board' mode in the solution. We have since adopted using the cell mid-side points for the definition of u and v velocities which has eliminated the checker-board mode with only very minor changes to the results.

## 10.2 TUFLOW HPC versus 2016-03 TUFLOW GPU

The TUFLOW HPC scheme differs from the original TUFLOW GPU scheme, in that TUFLOW GPU is only a 1<sup>st</sup> order upwind solver that calculates water level/depth and velocities all at the cell centres, and ignores the ground elevations and other parameters at the cell mid-sides.

TUFLOW GPU's 1<sup>st</sup> order only approach can experience numerical diffusion like all 1<sup>st</sup> order schemes, and does not resolve strongly two-dimensional hydraulics (eg. flow expansion downstream of a constriction) as well as a 2<sup>nd</sup> order solution. TUFLOW GPU also does not support the higher resolution topographic sampling at cell mid-sides, and therefore does not support functionality such as thin breaklines to represent narrow obstructions such as concrete levees and fences. The primary difference between TUFLOW GPU and TUFLOW HPC 1<sup>st</sup> order, is that TUFLOW HPC 1<sup>st</sup> order, like 2<sup>nd</sup> order, solves fluxes at the cell sides and utilises cell mid-side elevations and roughness values.

TUFLOW GPU results, can also exhibit significant numerical noise (unsteadiness) and checkerboarding (oscillating water levels from one cell to the next). These issues do not occur for TUFLOW HPC, primarily due to the solving of fluxes at the cell faces, rather than the cell centres.

TUFLOW GPU also does not include the new functionality of 1D linking and numerous other additions to TUFLOW HPC. The TUFLOW GPU option is provided for legacy models, noting that the TUFLOW GPU code has essentially been frozen and unlikely to be further enhanced.

TUFLOW GPU is generally faster than TUFLOW HPC due to its simpler solution, smaller memory footprint, simpler code base and limited features.

### 10.3 Specifying Classic or HPC 2D Solver

#### 10.3.1 Specifying which solver

For legacy reasons, the default 2D solver continues to be TUFLOW Classic's ADI implicit scheme.

The 2D solver is controlled by the .tcf command:

```
Solution Scheme == [ {Classic} | HPC ]
```

Specifying HPC invokes the HPC 2<sup>nd</sup> order solution. To run the 1<sup>st</sup> order solution specify:

```
Solution Scheme == HPC 1st
```

#### 10.3.2 Running the original (2016-03) TUFLOW GPU solver

For backward compatibility reasons, the original TUFLOW GPU solver is still invoked by "GPU Solver == ON". This command is not used for running the HPC solver. The original TUFLOW GPU Solver only runs on NVidia GPU devices and continues to require a TUFLOW GPU Hardware licence.

#### 10.3.3 Using scenarios to control which solver

Scenarios can be used to control which solver as per the example .tcf file syntax below:

```
If Scenario == CLA
    Solution Scheme == Classic
Else If Scenario == HPC
    Solution Scheme == HPC
Else If Scenario == GPU_Original
    GPU Solver == ON
Else
    Stop == "Invalid solution scheme specified"
End If
```

### 10.4 Hardware Specification

TUFLOW HPC can be simulated on CPU or NVidia GPU hardware, whilst TUFLOW Classic continues to be CPU based and the original TUFLOW GPU Solver NVidia GPU based. For HPC the following .tcf command controls the hardware to use.

- "Hardware == GPU" to use an NVidia GPU. A GPU module licence is required for each GPU device being accessed.
- "Hardware == CPU" to run on one or more CPU cores/threads. By default two (2) threads are used. The number of threads can be controlled with the .tcf command "CPU Threads == <number\_of\_threads>". For example, "CPU Threads == 6" runs the HPC 2D solver across 6 CPU threads, noting that the number of threads requested is limited to the maximum number of threads available on the machine, and the available TUFLOW Thread licences. The default number of

Thread licences is twice the number of TUFLOW Engine licences. For example, a Local 4 licence allows up to a maximum of 8 threads in use at any one time across all simulations.

The .tcf syntax above can be expanded to control the hardware used by HPC as follows:

```
If Scenario == CLA
    Solution Scheme == Classic
Else If Scenario == HPC
    Solution Scheme == HPC
If Scenario == CPU
    Hardware == CPU
Else If Scenario == GPU
    Hardware == GPU
Else
    Stop == "Invalid hardware specified"
End If
Else If Scenario == GPU_Original
    GPU Solver == ON
Else
    Stop == "Invalid solution scheme specified"
End If
```

## 10.5 Timestepping and Control Numbers

**Note:** Build 2017-09-AB improves upon the adaptive timestepping where a repeat timestep occurs prior to synchronising with a times series output or map output time. In Build 2017-09-AA, very small following timesteps could result in this situation, which has been rectified in Build 2017-09-AB. Therefore, very slight changes in results may occur between 2017-09-AA and 2017-09-AB due to slightly different timestepping.

**Note:** Build 2017-09-AB fixes an issue where if the .tcf “Time Series Output Interval ==” was not specified, and there is no 2D time series output, but a “1D Output Interval ==” is specified, the HPC timestepping synchronised time series output using an interval equal to the initial “Timestep ==” value instead of the “1D Output Interval ==”. This has been corrected.

The HPC solver, by default, uses adaptive timestepping to progress through the simulation. The timestep is adjusted so that it complies with the mathematical stability criteria of a 2D SWE explicit solution. In contrast, TUFLOW Classic, being an implicit solution, is not numerically bound by these criteria, and can progress through the simulation using a larger, fixed, timestep. Therefore, HPC typically uses a smaller timestep than Classic.

The HPC timestep needs to conform to the limiting timestep as determined by three stability criteria or Control Numbers:

- Courant Number,  $N_u$
- Wave Celerity Number,  $N_c$
- Diffusion Number,  $N_d$

Prior to advancing in time, the limiting timestep for all wet 2D cells is determined, and the minimum of the limiting timesteps for all cells is selected as the next timestep.

The three stability criteria (Control Numbers) are:

#### 10.5.1 Courant Number, $N_u$

The Courant number,  $N_u$ , ensures that water entering one side of a 2D cell does not pass through the other side within one timestep. That is, the velocity of the water multiplied by the timestep does not exceed the cell length. The equation for  $N_u$  is given below and  $N_u$  must be less than or equal to 1.0, where  $u$  is the X-velocity,  $v$  the Y-velocity,  $\Delta t$  is the timestep and  $\Delta x$  and  $\Delta y$  are the cell dimensions.

$$N_u = \max\left(\frac{|u|\Delta t}{\Delta x}, \frac{|v|\Delta t}{\Delta y}\right) \leq 1.0$$

#### 10.5.2 Shallow Wave Celerity Number, $N_c$

The SWE is derived from the fluid flow equations to represent long waves, i.e. the wave length is substantially longer than the water depth. Long waves include floods, the ocean tide and tsunamis, excluding near shore wave breaking which has accelerations in the vertical (third) dimension that violate the long wave SWE assumptions. The term “Shallow” in Shallow Water Equations is somewhat misleading in that the SWE are applicable in deep water provided the wave length is substantially longer and there are no accelerations in the vertical.

The shallow wave celerity number,  $N_c$ , is based on a long wave’s propagation speed, which is the square root of gravity times the water depth. The  $N_c$  criteria ensures that the wave does not propagate from one side of the 2D cell to past the other side in one timestep. The equation for  $N_c$  is provided below, and  $N_c$  must be less than or equal to 1.0.  $g$  is acceleration due to gravity,  $h$  the water depth,  $\Delta t$  the timestep, and  $\Delta x$  and  $\Delta y$  are the cell dimensions.

$$N_c = \max\left(\frac{\sqrt{gh}\Delta t}{\Delta x}, \frac{\sqrt{gh}\Delta t}{\Delta y}\right) \leq 1.0$$

#### 10.5.3 Diffusion Number, $N_d$

Turbulent Eddy Viscosity causes diffusion of momentum, and in the HPC scheme the diffusion is handled explicitly. Consider a cell that has a  $u$ -velocity that is higher than its neighbours by some amount delta. The diffusion number represents the fraction of delta that may diffuse through one face of the cell in one timestep (remembering a cell may have up to four active faces). With the fourth order temporal explicit solver, we have found that this fractional quantity must be kept at or below 0.3.

The diffusion number,  $N_d$ , is:

$$N_d = \max\left(\frac{\nu_T \Delta t}{\Delta x^2}, \frac{\nu_T \Delta t}{\Delta y^2}\right) \leq 0.3$$

where  $\nu_T$  is the eddy viscosity coefficient (the constant eddy viscosity coefficient plus that calculated using the Smagorinsky formula),  $\Delta t$  the timestep, and  $\Delta x$  and  $\Delta y$  are the cell dimensions.

#### 10.5.4 Initial Timestep

The standard “Timestep ==” command is only used by the HPC solver for the first timestep if adaptive timestepping mode. For consistency with typical Classic timesteps, this timestep is divided by 10 to provide a timestep commensurate with the explicit solution scheme used by HPC. Therefore, the general advice is to specify the same timestep as would typically be used when running the Classic solver. This is usually one fifth to half of the 2D cell size in metres.

As of the 2017-09-AC Build the .tcf command below can be used to set the HPC initial timestep directly (i.e. the timestep is not divided by 10).

```
Timestep Initial == <initial_dt>
```

#### 10.5.5 Minimum Timestep

The minimum permissible target timestep allowed by the HPC solver is set using the .tcf command below. By default this is set to the minimum of 0.1 s or the cell size divided by 200 m/s. In most cases, where there is no erroneous data or poor model setup, the target timestep will always be well above the default minimum timestep. As discussed in the HPC Q&A (Section 11), a very small target timestep is a strong indicator of a model construction or input data issue, and the minimum permissible timestep is used to alert the modeller of this possibility.

```
Timestep Minimum == <min_permissible_dt>
```

As of the 2017-09-AC Build if the minimum permissible timestep is greater than the initial timestep the simulation will abort with an ERROR 2440 in the .tlf file and an ERROR 3010 in the hpc.tlf file.

Note that if setting the initial timestep using the standard “Timestep ==” command, the initial timestep is the timestep divided by 10 – see Item 10.5.4 above. Prior to Build 2017-09-AC, HPC would enter an infinite loop if this occurred.

#### 10.5.6 Changing the Control Number Limits

The default control number limits can be reduced (or increased – be careful!) to effectively underclock or overclock the simulation. Using the .tcf command below factors all three control numbers. For example, a value of 0.8 reduces the limits from 1.0, 1.0, 0.3 to 0.8, 0.8, 0.24, and will increase the run time by 20%. Reducing the control number limits may be useful if the simulation is exhibiting erratic behaviour or numerical “noise”, although testing has found this is rare in real-world models, and if occurring is more likely to be a sign of poor data or poor model schematisation.

```
Control Number Factor == <CN_factor>
```

#### 10.5.7 1D Timestep

The 1D timestep for a HPC 1D/2D linked model is now the maximum or limiting timestep the 1D solver can use. The 1D solver has been reconfigured to act as an adaptive/varying timestep solution, and can step at different multiples of steps to the HPC 2D solver. Both 1D and 2D solutions are always synchronising at the 2D target timestep, or a multiple of the 2D target timestep

if the 1D timestep is sufficiently greater for the 2D to perform more than one step. If the 1D limiting timestep is less than half the 2D target timestep, the 1D proceeds in two or more steps eventually synchronising with the 2D timestep. Where there is not a one to one synchronisation of the 1D and 2D timesteps, a usually negligible mass error may occur and can be checked by reviewing the CME% values shown on the Console Window, the .tlf file or the \_MB.csv file in the same manner as Classic.

To force the 1D and 2D timestepping to be synchronised one-to-one use the .tcf command:

```
HPC 1D Synchronisation == EVERY 2D Timestep
```

#### 10.5.8 Fixed Timestep

The HPC solver can be run using a fixed timestep by setting the Control Number Factor to 0 (zero) as per the .tcf command below.

```
Control Number Factor == 0
```

In this case the timestep used by HPC is fixed and the timestep is set using the .tcf command below. Note that the timestep is not divided by 10 in this instance.

```
Timestep == <fixed_dt>
```

When running using a fixed timestep, there are no checks on exceedance of control numbers or application of the repeated timestep feature described in Section 10.6. If the control numbers are exceeded, the simulation has a high risk of going unstable, which is detected by the occurrence of NaNs (Not a Number) in the calculations. However, unstable results can occur prior to NaNs being detected, therefore if no NaNs occur this is not an indication the simulation was stable.

Prior to Build 2017-09-AC, if a fixed timestep was used, HPC would enter an infinite loop if a high control number occurred. As of 2017-09-AC HPC the simulation continues until unstable or finished.

#### 10.6 Repeated Timestep Feature

If using the adaptive timestepping (the default), the timestep is calculated using the hydraulic conditions from the end of the previous timestep. Therefore, it is possible for one or more of the control number criteria to be violated at one or more locations within the model. For example, a sudden change in rainfall from one timestep to the next (which occurs with stepped rainfall boundaries) would potentially cause a violation. The HPC solver, by default, treats a 20% exceedance of a control number as a violation and will implement the repeat timestep feature.

The repeat timestep feature involves retaining the complete hydraulic solution from the previous (good) timestep. Should a control number anywhere within the model be exceeded by more than 20%, the solution reverts to the retained timestep, the timestep is reduced and then repeated. Should a timestep need to be repeated more than ten times consecutively, the solution stops. This occurrence is rare and an indication of poor model schematisation or poor data.

Each timestep is also tested for the occurrence of NaNs. A NaN is “Not a Number” and occurs due to undefined mathematical calculations such as a divide by zero or square root of a negative

number. The occurrence of a NaN is also indicative of a sudden instability. Should a NaN occur, the repeat timestep feature is implemented.

Repeated timesteps are displayed to the Console Window and the number of them for a time interval are provided in the nRS\_NaNs and nRS\_HCNs columns in the \_HPC.csv file output in the results folder. They are also reported to the \_messages layer.

Repeated timesteps are an indication the 2D HPC solution is numerically “on-the-edge”. Models that have a high number of repeated timesteps should be sensitivity tested by reducing the control number limits using “Control Number Factor ==” .tcf command. For example, repeat the simulation using “Control Number Factor == 0.8” and compare the results. If there are acceptably immeasurable changes in the results, then running at the default control number limits can be considered satisfactory.

## 10.7 Miscellaneous Items

### 10.7.1 Ctrl-C Termination of Simulation

For 1D linked simulations the HPC 2D and 1D Classic solvers are communicating every timestep, and the simulation can be terminated early and cleanly by using the standard Ctrl-C option. This also includes 2D only simulations that trigger the creation of hidden 1D nodes via the 2D QT boundary. Otherwise, for all other 2D only simulations the HPC 2D solver only communicates with the TUFLOW.exe at map output intervals or, if implemented, the next time-varying rainfall grid input, therefore, if Ctrl-C is applied the simulation will terminate cleanly, **but not until the next map output or rainfall grid input interval is reached**.

### 10.7.2 Information in .tsf File

The output to the .tsf file has been expanded to include the following:

```
Solution Scheme == HPC  
Hardware == GPU  
GPU Device IDs == 0
```

Time varying values are provided as follows. These are provided every “Mass Balance Output Interval ==”. Note, for 2D only simulations with no hidden 1D nodes, if “Mass Balance Output Interval ==” is not equally divisible into the “Map Output Interval ==”, the mass balance output provided by the HPC solver may not be synchronised with the the “Mass Balance Output Interval ==” steps.

```
HPC Number of HCN Repeated Timesteps == 0 ! HCN is High Control Numbers  
HPC Number of NaN Repeated Timesteps == 0 ! NaN - Not a Number indicates  
an instability occurred  
HPC Number of NaN WARNING 2550 == 0
```

## 10.8 Mass Conservation

### 10.8.1 HPC 2D Solver Mass Balance Tracking and Other Information (\_HPC.csv file)

The \_HPC.csv file is written to the 2D “Output Folder ==” location. The file contains mass balance tracking of volumes of water in and out, mass error calculations and other information. Note that the mass tracking is carried out using single precision real numbers (~7 significant figures), therefore, numerical imprecision in the mass balance tracking can arise due to use of single precision that is unrelated to any mass error in the calculations. Essentially, the HPC scheme, being a finite volume scheme, is volume conservative having no mass error. This differs from TUFLOW Classic, which relies on convergence of the matrix solution, with poor convergence evident by a high mass error percentage.

The columns in the \_HPC.csv file are provided in the table below.

Column	Description
Time (h)	Simulation time in hours
iStep	Number of computational steps
nRS_NaNs	Number of repeated timesteps due to Not a Number (NaN) occurring. A NaN occurs when the solution becomes unstable causing a divide by zero or other undefined calculation forcing a lowering and repeat of the timestep.
nRS_HCNs	Number of repeated timesteps due to High Control Numbers (HCN). A HCN means that one of the three stability criteria was exceeded by more than 20% forcing a lowering and repeat of the timestep.
dtTarget	The target timestep or timestep calculated from the stability criteria. A lower timestep may have been used if approaching a times series or map output time, as HPC will reduce the timestep so that the solution timestepping coincides with the output time.
tLastMax	Time of the last recorded maximum water level.
H Vol In	Volume in since the previous time via 2D H boundaries (HQ, HT).
H Vol Out	Volume out since the previous time via 2D H boundaries (HQ, HT).
S/RF Vol In	Volume in since the previous time from 2D source boundaries (RF, SA, ST, Infiltration).
S/RF Vol Out	Volume out since the previous time from 2D source boundaries (RF, SA, ST, Infiltration).
HX Vol In	Volume in since the previous time via 2D HX boundaries.
HX Vol Out	Volume out since the previous time via 2D HX boundaries.
SX Vol In	Volume in since the previous time via 2D SX boundaries.
SX Vol Out	Volume out since the previous time via 2D SX boundaries.
V In-Out	Volume In less Volume Out of 2D domain.
dVol	Change in total volume of water within 2D domain.
V Err	“dVol” less “V In-Out”.
Total V	Total volume of water in 2D domain.
Q ME%	Mass error expressed as a percentage of “V Err”/(V <sub>in</sub> + V <sub>out</sub> ).

### 10.8.2 HPC Whole of Model Mass Balance Tracking (\_MB.csv file)

This file is similar to the Classic \_MB.csv file and covers the whole model, i.e. including any 1D domains.

### 10.8.3 Build 2017-09-AC – Volume Output to .tlf and .tsf Files

As of Build 2017-09-AC summary volume statistics are now output for the whole model (i.e. 1D included) at the end of the .tlf file for HPC in a similar manner to Classic. For the .tsf file, the volume outputs are now for the whole model (1D and 2D), whereas previously only the 2D component was provided for HPC runs. Additional summary volume information has also been added to the .tsf file for both Classic and HPC.

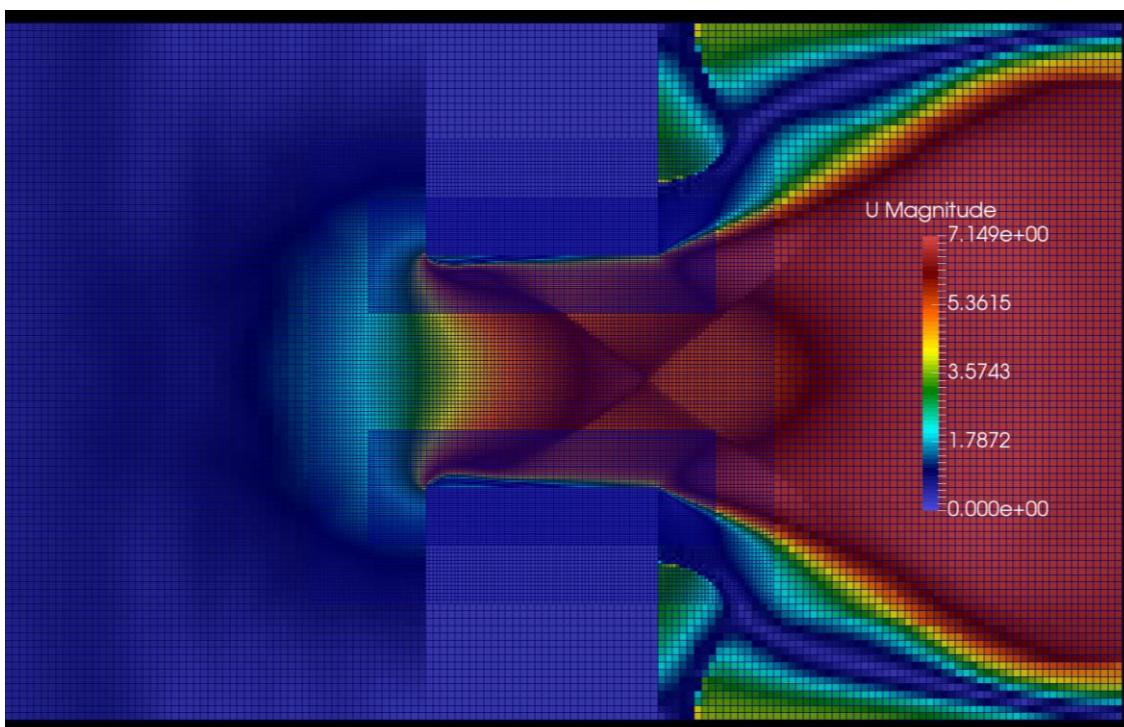
## 10.9 TUFLOW HPC (as yet) Unsupported Features

TUFLOW Classic functionality not yet built into the TUFLOW HPC solver includes:

### 10.9.1 Multiple 2D Domains

We are investigating the incorporation of Multiple 2D Domains using HPC for the 2018 release.

Of note for the 2018 release is incorporation of nested gridding on a quad-tree type arrangement. The computational feasibility of this has been prototyped with excellent results as demonstrated in the image below, which shows undistorted flows through three levels of nesting in a complex flow expansion case.



### 10.9.2 Traditional flow constrictions

2d\_fc layers are not yet supported, and results in an ERROR 2320 during start-up. Note that layered flow constrictions are supported.

### 10.9.3 Weir factors (WrF)

WrF factors for adjusting/calibrating 2D weir flow are not yet available for HPC.

### 10.9.4 Only Manning's n

Only Manning's n bed resistance values can be used (Chézy and Fric option are not supported).

### 10.9.5 Read GIS Z Shape Route

Evacuation route monitoring ("Read GIS Z Shape Route ==") is not yet built into HPC.

### 10.9.6 Read GIS Objects

"Read GIS Objects ==".tgc command is not yet supported.

### 10.9.7 Reporting Locations, LPs and Structure Groups

Reporting locations, "Read GIS LP ==" and structure groups are not yet fully built into HPC, but will be provided in an update for the 2017-09 release. 2D PO (plot output) are supported.

### 10.9.8 Map Output Data Types not supported

"Map Output Data Types ==" not supported are listed below. Except for those not required or applicable, these outputs are planned for future updates/releases.

- UD (User Defined output)
- MB1 and MB2 (HPC is mass conserving to numerical precision and does not require this output)
- RC (Route Categories)
- n (Manning's n values for when using depth varying n values)
- R (flow Regime)
- AP and WI (Atmospheric Pressure and Wind)
- Tracked maximum hazards except for Z0, ZUK0, ZUK1, ZUK2 and ZUK3, which are presently tracked within the HPC solver every timestep.
- No minimums, except for the new minimum timestep (dt) output, are tracked using HPC.

### 10.9.9 TUFLOW AD

The TUFLOW AD module presently only works with the Classic solver. A HPC AD solver has been developed and is scheduled for testing and release during 2017/2018.

### 10.9.10 Material IL/CL Rainfall Losses

Note that TUFLOW HPC accounts for IL/CL direct rainfall (RF) losses via materials. The original TUFLOW GPU solver ignores the IL/CL values for materials.

Please note the critical bug fix described in Item 4.7.

## 11 TUFLOW HPC Q&A

### 11.1 Does the HPC 2D solver produce the same results as the Classic 2D solver?

No. TUFLOW Classic uses a 2<sup>nd</sup> order ADI (Alternating Direction Implicit) finite difference solution of the 2D SWE, while the HPC solver uses a 2<sup>nd</sup> order explicit finite volume TVD (Total Variation Diminishing) solution (a 1<sup>st</sup> order HPC solution is also available). As there is no exact solution of the equations (hence all the different solvers!), the two schemes produce different results.

However, in 2<sup>nd</sup> order mode the two schemes are generally consistent with testing thus far indicating Classic and HPC 2<sup>nd</sup> order produce peak level differences usually within a few percentage points of the depth in the primary conveyance flow paths. Greater differences can occur in areas adjoining the main flow paths and around the edge of the inundation extent where floodwaters are still rising or are sensitive to a minor rise in main flow path levels, or where upstream controlled weir flow across thick or wide embankments occurs due to the different numerical approaches.

For deep, fast flowing waterways, 1<sup>st</sup> order HPC tends to produce higher water levels and steeper gradients compared with the Classic and HPC 2<sup>nd</sup> order solutions. These differences can exceed 10% of the primary flow path depth. Typically, lower Manning's n values are required for HPC 1<sup>st</sup> order (or the original TUFLOW GPU), to achieve a similar result to TUFLOW Classic or HPC 2<sup>nd</sup> order.

Significant differences may occur at 2D HQ boundaries. Classic treats the 2D HQ boundary as one HQ boundary across the whole HQ line, setting a water level based on the total flow across the line. Due to model splitting to parallelise the 2D domain across CPU or GPU cores, HPC applies the HQ boundary slope to each individual cell along the boundary. As with all HQ boundaries, the effect of the boundary should be well away from the area of interest, and sensitivity testing carried out to demonstrate this.

### 11.2 Do I need to change anything to run a Classic model in HPC?

For single 2D domain model, no, other than inserting the .tcf commands “Solution Scheme == HPC” and, if running on a GPU device, “Hardware == GPU”.

HPC does not yet support multiple 2D domain models.

Note that some more specialised or rarely used features are still not incorporated into the HPC solver – refer to Item 10.8.3.

### 11.3 Do I need to review or recalibrate if I switch from Classic to HPC?

Yes, if transitioning from Classic to HPC (or any other solver), it is best practice to compare the results, and if there are unacceptable differences, or the model calibration has deteriorated, to fine-tune the model performance through adjustment of key parameters.

Typically, between TUFLOW Classic and HPC 2<sup>nd</sup> order this would only require a slight adjustment to Manning's n values, any additional form losses at bends/obstructions or eddy viscosity values. Regardless, industry standard Manning's n and other key parameters should only be used/needed.

Use of non-standard values is a strong indicator there are other issues such as inflows, poor boundary representation or missing/erroneous topography.

A greater adjustment of parameters would be expected if transitioning between HPC 1<sup>st</sup> order (or the original TUFLOW GPU) and Classic or HPC 2<sup>nd</sup> order.

## 11.4 Why does my HPC simulation take longer, or is not much faster, than Classic?

If running on the same CPU hardware, a well-constructed Classic model on a good timestep is nearly always faster than HPC running on a single CPU thread. Running a single HPC simulation across multiple CPU threads may produce a faster simulation than Classic. Primary reasons why the HPC may produce a slower run, especially are discussed below.

### 11.4.1 Over utilisation of CPU threads/cores

Trying to run multiple HPC simulations across the same CPU threads. If, for example, you have 4 CPU threads on your computer and you run two simulations that both request 4 threads, then effectively you are overloading the CPU hardware by requesting 8 threads in total. This will slow down the simulations by more than a factor of 2. The most efficient approach in this case is to run both simulations using 2 threads each, noting that if you are performing other CPU intensive tasks, this also needs to be considered.

You can control the number of threads requested by either using the -nt<number\_threads> run time option, e.g. -nt2, or use the .tcf “CPU Threads ==”. The -nt run time option overrides “CPU Threads ==”.

By default, the number of CPU threads taken is two (2).

**Note:** If Windows hyperthreading is active there typically will be two threads for each physical core. For computationally intensive processes such as TUFLOW, it is recommended that hyperthreading is deactivated so there is one thread for each core, especially on dedicated modelling machines where an improvement performance is likely.

**Note:** If Windows hyperthreading is active, there is little benefit in setting “CPU Threads ==” to greater than the number of physical cores, plus this also uses additional TUFLOW Thread licences for no or little gain in performance. Therefore, set “CPU Threads ==” to no greater than the number of physical cores (which can be viewed in Task Manager).

**Note:** To request the maximum number of threads on a machine use “CPU Threads == MAX” if hyperthreading is deactivated and “CPU Threads == MAX/2” if hyperthreading is active. The latter requests half the number of threads, which for most Windows machines is the same as the number of physical cores.

### 11.4.2 Low end GPU

If running a simulation using a low end or old GPU device (these are usually the graphics card that comes standard with computer), simulations can be only marginally faster, or even slower, than running Classic or HPC on CPU hardware. If running on a GPU device, high end NVidia graphics are strongly recommended. The performance of different NVidia cards varies by orders of magnitude – for benchmark tests using the original TUFLOW GPU solver see [this Wiki page](#).

#### 11.4.3 The HPC adaptive timestepping is selecting very small timesteps

Common reasons why HPC adopts a very small timestep are provided below. To review and isolate the location of the minimum timestep the timesteps are output to:

- Console window and .hpc.tlf file
- .hpc.dt.csv file (this file contains every timestep)
- “Minimum dt” map output (excellent for identifying the location of the minimum timestep adopted – add “dt” to “Map Output Data Types ==”)

Common reasons for selecting very small timesteps are:

- The model has one or more or erroneous deep cells. The Celerity Control Number described further above reduces the timestep proportionally to the square root of the depth, so any unintended deep cells can cause a reduction in the timestep.
- Poorly configured or schematised 2D boundary or 1D/2D link causing uncontrolled or inaccurate flow patterns. The high velocities may cause the Courant Number to control the timestep or the high velocity differentials can cause the Diffusion Number to force the timestep downwards. In these situations Classic would often become unstable alerting the modeller to an issue. However, HPC will remain stable relying on the modeller to perform more thorough reviews of flow patterns at boundaries and 1D/2D links.
- If using the SRF (Storage Reduction Factor), this proportionally reduces the  $\Delta x$  and  $\Delta y$  length values in the control number formulae. This may further reduce the minimum timestep if a cell with an SRF value greater than 0.0) is the controlling cell. For example, applying a SRF of 0.8 to reduce the storage of a cell by 80% or a factor of 5, also reduces the controlling timestep for that cell by a factor of 5.

#### 11.5 I'm familiar with Classic, so do I need to be aware of anything different with HPC?

Yes! TUFLOW Classic tells you where your model has deficient or erroneous data, or where the model is poorly set up by going unstable, or producing a high mass error (a sign of poor numerical convergence of the matrix solution). The best approach when developing a Classic model is to keep the timestep high (typically a half to a quarter of the cell size in metres), and if the simulation becomes unstable to investigate why. In most cases, there will be erroneous data or poor set up such as a badly orientated boundary, connecting a large 1D culvert to a single SX cell, etc.

HPC, however, remains stable and does not alert the modeller to these issues. Therefore, the following tips are highly recommended, otherwise there will be a strong likelihood that any deficient aspects to the modelling won't be found till much further down the track, potentially causing costly reworking. So, it's very much modeller beware!

- Use of excessively small timesteps is a strong indicator – see discussion further above.
- If the timestepping is erratic (i.e. not changing smoothly), or there is a high occurrence of repeated timesteps, these are indicators of an issue in the model data or set up. See Item 11.4.3 for how to review the timestep values.

- Be more thorough in reviewing the model results. Although this is best practice for any modelling, it is paramount for unconditionally stable solvers like HPC that thorough checks of the model's flow patterns, performance at boundaries and links is carried out.
- The CME%, which is an excellent indicator that the Classic 2D solver is numerically converging, is not generally of use for HPC, which is volume conserving and effectively 0% subject to numerical precision. Non-zero whole of model CME% for HPC 1D/2D linked models is usually an indication of either the 1D and 2D adaptive timesteps being significantly different, or a poorly configured 1D/2D link – see Item 10.5.7.