

## **TUFLOW FV Release Notes** **Builds 2020.01 – 2020.03**

[www.tuflow.com](http://www.tuflow.com)

[TUFLOW FV Wiki](#)

[TUFLOW Tutorial Model](#)

[support@tuflow.com](mailto:support@tuflow.com)

## Document Updates and Important Notices

**May 14th, 2020:** TUFLOW FV 2020.01 is a major release that includes significant upgrades to the Sediment Transport and Particle Tracking Module offerings plus a range of new functionality, enhancements and minor bug fixes. For new modelling projects it is recommended to upgrade to Build 2020.01 to make use of this new functionality and improved workflows. For existing projects, while we have provided backward compatibility it is recommended to assess potential changes by completing result comparisons between previous and new builds.

**August 31, 2020:** The 2020.02 TUFLOW FV Build is a major update that includes TUFLOW FV's new GPU Solver capabilities (Refer Section 4). Also addressed are several bug fixes and enhancements. Features and fixes related to the 2020.02 Release are provided in light green.

**July 23, 2021:** The 2020.03 TUFLOW FV Build includes a range of bug fixes and enhancements. Features and fixes related to 2020.03 are shaded in light blue. For TUFLOW FV GPU users a number of important fixes and enhancements are included and it is recommended to upgrade. If modelling on CPU hardware with sediment transport, passive tracers or water quality variables please take note of Section 5.5 regarding Fortran Compiler upgrades that may affect results when upgrading to 2020.03.

Note, this document will be updated periodically with 2020 software updates describing the new builds and content. An updated TUFLOW FV User Manual for 2020 is currently in final stages of documentation and will soon be available and announced via the TUFLOW FV Forum and LinkedIn TUFLOW User Group. In the interim please use these Release Notes in combination with the existing [TUFLOW FV Manual](#).

**Both Windows and Linux operating system builds are available on the [TUFLOW Download Page](#). Linux users please contact [support@tufLOW.com](mailto:support@tufLOW.com) for guidance on Linux CUDA enabled (GPU compatible) installation.**

## Sediment Transport Module Control File

If using pre-2020 builds of TUFLOW FV with sediment transport, the Sediment Control File will require update to the new Sediment Transport Module format. Without this update you will likely encounter a series of 'Unrecognised flag encountered on line' errors. Please refer to the [Sediment Transport and Particle Tracking User Manual](#) and [Tutorial Module 05 download package](#) for simple examples to help guide you. If still unsure, please contact [support@tufLOW.com](mailto:support@tufLOW.com). A sediment transport tutorial example/model will be uploaded to the wiki in the coming months.

## Licensing and Executable Versions

To run simulations using the Windows or Linux 2020.01 build requires payment of the 2019/2020 annual software maintenance fee and for the TUFLOW licence to have been updated (i.e. via RaC/RaU files).

To run simulations using the Windows or Linux 2020.02 build requires payment of the 2020/2021 annual software maintenance fee and for the TUFLOW licence to have been updated (i.e. via RaC/RaU files).

To run simulations using the Windows or Linux 2020.03 build requires payment of the 2020/2021 annual software maintenance fee and for the TUFLOW licence to have been updated (i.e. via RaC/RaU files).



For tutorial and demo models, 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 suite is bound by the [TUFLOW Products Licence Agreement](#).

## Table of Contents

<b>Document Updates and Important Notices.....</b>	<b>2</b>
<b>Sediment Transport Module Control File.....</b>	<b>2</b>
<b>Licensing and Executable Versions.....</b>	<b>2</b>
<b>1 2020 Release Overview.....</b>	<b>6</b>
1.1 Sediment Transport Module Overview .....	6
1.2 Particle Tracking Module Overview .....	6
1.3 New Features and Enhancements .....	7
1.4 Upgrading to the 2020 Release.....	7
1.5 2020 Release Updates .....	8
<b>2 Sediment Transport Module.....</b>	<b>9</b>
2.1 Available Sediment Transport Models.....	11
2.2 User Manual.....	12
2.3 STM Bug Fixes and Enhancements .....	12
<b>3 Particle Tracking Module.....</b>	<b>14</b>
3.1 Available Particle Models .....	14
3.2 User Manual.....	16
3.3 PTM Bug Fixes and Enhancements .....	16
<b>4 GPU Module .....</b>	<b>18</b>
4.1 Introduction .....	18
4.2 CUDA Enabled Linux Installers .....	18
4.3 Will the Results be the Same as TUFLOW FV on CPU? .....	18
4.4 Compatible Graphics Cards .....	18
4.5 Running TUFLOW FV on GPU .....	22
4.6 Troubleshooting.....	22
4.7 Command Line Updates for Hardware Selection .....	23
4.8 GPU Engine Bug Fixes and Enhancements .....	23
<b>5 Hydrodynamic Engine .....</b>	<b>24</b>
5.1 Model Geometry .....	24
5.2 Boundary Conditions .....	26
5.3 Model Outputs .....	26
5.4 Hydrodynamic Engine Bug Fixes and Enhancements .....	28
5.5 Fortran Compiler Optimisation .....	37
<b>6 Advection Dispersion Module .....</b>	<b>38</b>
6.1 Atmospheric Heat Exchange.....	38



6.2	Advection Dispersion Module Bug Fixes and Enhancements .....	38
<b>7</b>	<b>Water Quality Module .....</b>	<b>40</b>
7.1	Application Programming Interface (API) .....	40
7.2	AED Version 2.0.0 .....	40
7.3	Output headers.....	43
<b>8</b>	<b>General Updates .....</b>	<b>44</b>
8.1	Enhancement to the QGS file .....	44
8.2	Shapefile Projection Changes.....	44
<b>9</b>	<b>Visualisation and Model Development Tools .....</b>	<b>45</b>
9.1	MATLAB Toolbox .....	45
9.2	Python Toolbox.....	45
9.3	TUFLOW Viewer QGIS Plugin .....	46
<b>10</b>	<b>Mesh Generation Tools .....</b>	<b>48</b>
10.1	Aquaveo SMS .....	48
10.2	Rising Water Software GIS Mesher.....	48
<b>11</b>	<b>User Manuals, Tutorials and Supporting Materials.....</b>	<b>49</b>
11.1	User Manuals .....	49
11.2	Tutorial Models.....	49
11.3	TUFLOW FV Wiki .....	52
11.4	TUFLOW FV Forum.....	52
11.5	TUFLOW LinkedIn User Group .....	52
<b>12</b>	<b>Licensing and Installing.....</b>	<b>53</b>
12.1	Linux Installer Packages .....	53
12.2	Security Certificate .....	53
12.3	Extra Licence Control File location .....	53

## 1 2020 Release Overview

The major announcement for the 2020 Release (Build 2020.01) is the unveiling of our new state-of-the-art Sediment Transport Module and Particle Tracking Modules.

Build 2020.02 is the first software update of the 2020 TUFLOW FV Release. A major announcement for this update is the release of our new TUFLOW FV GPU Solver (refer Section 4).

Build 2020.03 is the second software update and includes a range of enhancements and fixes.

### 1.1 Sediment Transport Module Overview

TUFLOW FV's sediment transport module is a multiple fraction, multi-bed-layer sediment transport and coupled hydrodynamic-morphological model. It has been used extensively in creeks, rivers, estuaries, coastal and ocean environments allowing simulation of :

- Sediment transport due to currents and/or wave driven processes
- Morphological evolution and feedback to hydrodynamics
- Multiple sediment fractions, including mixed cohesive and non-cohesive sediments (sands silts, clays and muds in a single model)
- Sediment exchange between the water column and seabed (deposition and erosion)
- Advection and dispersion of suspended sediment (2D or 3D)
- Flocculation and hindered settling
- Bed layer consolidation
- Bed load transport, bed slumping and armouring
- Fluid density (for 3D baroclinic simulations) .

The implementation of the sediment transport module allows the user a high level of control over sediment characteristics. Within a single model run, sediment fraction groups can be assigned as cohesive or non-cohesive. There is also the added flexibility of selecting a range of common sediment transport models/equations independently for each fraction. More information on the Sediment Transport Module is provided within Section 2.

### 1.2 Particle Tracking Module Overview

The particle tracking model is a Lagrangian model for the 2D/3D transport equations of discrete particles. This scheme allows particles to be tracked independently to the model mesh, allowing the tracking of individual particles over time. One or more named particle groups can be specified to control how particles respond to hydrodynamic and other environmental conditions such as wind, waves, currents or other user defined stimuli.

Particles can be introduced as fixed or moving source terms and can define characteristics such as particle age, settling, buoyancy and motility. Particles can be deposited on the bed and resuspended.

The particle tracking module supports a range of options for simulating particle motility, (e.g. swimming) in response to environmental conditions. The motile velocity component is added to other particle velocity components driven by hydrodynamics, wind drag, etc. Multiple sequential motility behaviours can be simulated based on specified start times (relative to age since particle seeding).

A crawling motility can be specified and will be applied when particles are currently located in dry cells.

Core functionality of the particle tracking module includes:

- Lagrangian particle tracking with particles divided into named groups
- Fixed and moving source terms
- Particle advection plus diffusion (random walk component)
- Vertical advection/diffusion scheme (validated to reproduce analytical concentration profiles)
- Multiple particle groups
- Particle deposition and resuspension (exchange with bed)
- Particle deposition/resuspension driven by current plus wave-driven bed shear
- Multiple bed layers allowing for consolidation and/or bioturbation
- Eulerian (particle concentration) outputs based on TUFLOW-FV mesh
- Interfacing with water quality modelling

Further information on the particle tracking module is provided in Section 3.

### 1.3 New Features and Enhancements

The 2020 Release also includes a range of user experience updates and minor enhancements including:

- Heat module updates
- Model output improvements
- Python tools update
- QGIS plugin updates including the native read of TUFLOW FV NetCDF and Particle NetCDF
- Changes to module licensing
- Updates to the TUFLOW FV User Manual
- New Sediment Transport and Particle Tracking User Manual
- New GPU Module.
- New horizontal momentum and scalar mixing model options.
- Inclusion of a new water quality Application Programming Interface (API)
- Bathymetric database structure option
- Evapoconcentration and tracer age within the advection dispersion module
- New GPU run options

### 1.4 Upgrading to the 2020 Release

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 simulation maxima, or comparing your preferred model statistic sets and reviewing any differences are easy

ways to do this. If you have any queries on the comparison outcomes, or require clarification of any of the new features in these release notes, please email [support@tuflow.com](mailto:support@tuflow.com).

## **1.5 2020 Release Updates**

Software updates to 2020 will be provided throughout 2020 and 2021 as further features become available and will be included in this documentation.

2020.02, released August 2020 is the first update to the 2020 release and includes numerous bug fixes and new features. Changes specific to the 2020.03 release are shaded in green.

2020.03, released July 2021 is the second update to the 2020 release and includes numerous bug fixes and new features. Changes specific to the 2020.03 release are shaded in blue.



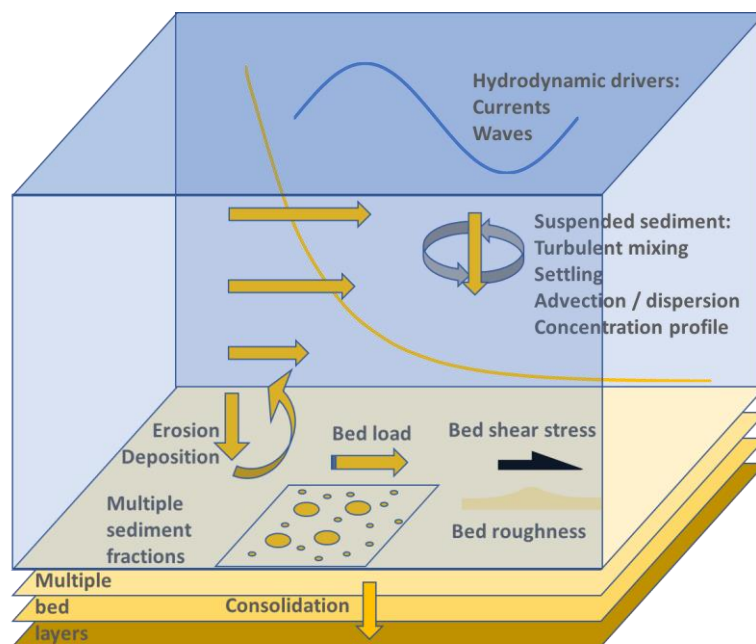
## 2 Sediment Transport Module

The TUFLOW Sediment Transport Module (STM) is a flexible and sophisticated bed load and suspended load sediment transport model that enables the 2D or 3D simulation of:

- Sediment transport in creeks, rivers, reservoirs, estuaries, coastal and ocean environments.
- Sediment transport due to currents and/or wave driven processes.
- Morphological evolution with hydrodynamic feedback.
- Sediment exchange between the water column and the bed (deposition and erosion).
- Advection and dispersion of suspended sediment.
- Bed load transport, bed slumping, bed consolidation and sediment sorting/armouring processes.

One-or-more sediment fractions can be simulated and transported as bed or suspended load. This fraction-based implementation allows a high level of control over sediment characteristics. For example, within a single model run, sediment fraction groups can be assigned as cohesive or non-cohesive and there is flexibility to select from a range of common sediment transport models/equations independently for each fraction. Figure 1 provides a conceptual model of the discrete sediment transport processes modelled by the STM and each component is mentioned in *Italics* in the following sections.

The STM is a Eulerian solver and fundamentally tracks sediment mass across discrete control volumes defined by the hydrodynamic model mesh or grid. If a Lagrangian scheme is preferred the [Particle Tracking Module](#) provides sediment transport behaviour within a Lagrangian frame of reference.



**Figure 1 TUFLOW Sediment Transport Module Conceptual Model**

The Hydrodynamic (HD) Engine controls the overall simulation and communicates the *Hydrodynamic Drivers* such as *Currents and Waves* to the STM. The HD Engine, Advection Dispersion (AD) Module

and STM work together to calculate the horizontal and vertical *Turbulent Mixing, Settling Advection Dispersion* and *Concentration Profiles* of suspended sediment within the water column. *Multiple Suspended Sediment Fractions* are individually tracked as scalar constituents by the AD Module.

Provided with hydrodynamic and suspended sediment conditions, the STM is responsible for resolving the wave-current boundary layer including potential bed forms (*Bed Roughness*) and resultant *Bed Shear Stress*. The STM calculates *Bed Load Transport* potential and sediment *Erosion-Deposition* exchanges at the bed and tracks the resultant change in sediment mass in the bed.

The STM conceptualises the bed as consisting of *Multiple Vertically Stratified Bed Layers*. Each layer comprises a mixture of the defined *Multiple Sediment Fractions*. The mass ( $\text{kg/m}^2$ ) of each sediment fraction within an individual cell's bed layer is tracked by the STM. The total quantity of sediment in each bed layer ( $\text{kg/m}^2$ ) and the associated dry density ( $\text{kg/m}^3$ ) are used to calculate the layer thickness (m).

The uppermost layer that contains a finite sediment mass is the 'active layer' and it is subject to sediment erosion and deposition. If the active layer erodes entirely, sediment is then progressively eroded from the underlying discretised layers. Sediment always deposits to the active layer and can be transferred to lower (usually higher density) layers via *Consolidation*. Simulation of sediment sorting and armouring of the active layer is also possible.

Each sediment transport process is calculated on a timestep specified via the STM Update  $\Delta t$  which is set independently to the HD timestep. The STM Update  $\Delta t$  will typically be much larger than the HD timestep, for example, the HD timestep may be sub-second to seconds and the STM Update  $\Delta t$  may occur every five to fifteen minutes. The purpose of this approach is to speed up simulations which are run over the larger timescales at which morphological changes typically occur. The adopted Update  $\Delta t$  is specific to the sediment transport problem being investigated, and the user judgement is required in its selection.

The net water column exchange, bed load and bed slumping are integrated over an STM Update  $\Delta t$  using an explicit Eulerian scheme. At computational cells experiencing net erosion, the mass update proceeds from the active layer down through underlying layers until either the calculated mass increment has been accounted for or the sediment source is exhausted. In computational cells experiencing net deposition, the resulting mass increase will be accumulated in the top bed layer.

Sediment consolidation fluxes are subsequently updated by looping through all layers and transferring mass to each next lower layer as required. Following the sediment mass updates, the layer thicknesses are re-calculated in each bed cell based on the specified layer dry density values. The total change in bed layer thickness is calculated for each computational cell and is passed to the HD model. Where morphological coupling is specified, the updated bed level is applied to the HD model.

## 2.1 Available Sediment Transport Models

The STM is built on a library of different model/equation options as provided in Table 1 and Table 2. If you have a specific model or equation set not currently available, that you would like to include in the STM, please contact [support@tufLOW.com](mailto:support@tufLOW.com).

The available sediment transport models can be separated into three core categories as follows:

- 1 Global models that are assigned to all sediment fractions. Concentration profile, bed roughness, bed shear stress and bed slumping models are required to be globally specified.
- 2 Sediment fraction models that are assigned to individual sediment fractions independently of other sediment fractions being computed. Settling, erosion, deposition, bed load, critical stress and consolidation models can all be specified on a fraction by fraction basis.
- 3 External models that are assigned globally but can be applied to one-or-more sediment fractions. Currently the van Rijn TRANSPOR model is the only external model supported by the STM.

### 2.1.1 Global Models

The STM models presented in Table 1 are specified globally and apply to all sediment fractions. The concentration profile models are required to be applied to the entire model domain. In contrast, whilst still applying to all sediment fractions, bed roughness, bed shear and bed slumping model parameters can be varied spatially via material blocks.

**Table 1 TUFLOW STM globally applied profile and bed models**

Concentration profile	Bed roughness	Bed shear stress	Bed slumping
<ul style="list-style-type: none"> <li>Default (3D higher-order reconstruction)</li> <li>First-order (higher-order reconstruction only at bed interface)</li> <li>Legacy (no reconstruction)</li> </ul>	<ul style="list-style-type: none"> <li>Specified Nikuradse Roughness (ks)</li> <li>Proportional to median sediment diameter (d50)</li> <li>van Rijn (2004)</li> </ul>	<ul style="list-style-type: none"> <li>Default</li> <li>van Rijn (2004)</li> <li>Bijker (1984)</li> </ul>	<ul style="list-style-type: none"> <li>None</li> <li>Simple (specified slope)</li> </ul>

### 2.1.2 Sediment Fraction Models

The STM provides the flexibility to define fraction independent models/equations. Table 2 provides an overview of the available sediment fraction models for settling, erosion, deposition, bed load, critical shear stress and consolidation. Notably, some models are more applicable to cohesive (CS) sediment, while others will be more applicable to non-cohesive (NCS) fractions or a combination of the two. Where applicable the labels NCS or CS are provided next to each sediment fraction model within Table 2. Models applicable to both NCS and CS are not labelled.

**Table 2 TUFLOW STM Sediment Fraction Models**

Settling	Erosion	Deposition	Bed load <sup>1</sup>	Critical Stress	Consolidation
<ul style="list-style-type: none"> <li>• None</li> <li>• Constant ws</li> <li>• Flocculation (CS)</li> <li>• Flocculation + hindered settling (CS)</li> <li>• van Rijn (1984) (NCS)</li> <li>• van Rijn External (2004) (NCS)</li> </ul>	<ul style="list-style-type: none"> <li>• None</li> <li>• Mehta (CS)</li> <li>• van Rijn (1984) (NCS)</li> <li>• van Rijn External (2004) (NCS)</li> <li>• Soulsby-van Rijn (NCS)</li> <li>• Bijker (NCS)</li> </ul>	<ul style="list-style-type: none"> <li>• None</li> <li>• Unhindered</li> <li>• Krone (CS)</li> </ul>	<ul style="list-style-type: none"> <li>• None</li> <li>• Meyer-Peter-Müller</li> <li>• MPM-Shimizu</li> <li>• van Rijn External (2004)</li> <li>• Soulsby-van Rijn</li> <li>• Bijker</li> <li>• Wilcock-Crowe</li> </ul>	<ul style="list-style-type: none"> <li>• None</li> <li>• Constant</li> <li>• Soulsby</li> <li>• Soulsby-Egiazaroff</li> </ul>	<ul style="list-style-type: none"> <li>• None</li> <li>• Constant</li> </ul>

<sup>1</sup>The majority of bed load is typically non-cohesive sediment. Cohesive sediments tend to be transported by erosion/resuspension and then advected as suspended sediment prior to being deposited.

### 2.1.3 External Models

The STM currently includes the van Rijn TRANSPOR “external” sediment transport model, which in itself provides a comprehensive boundary layer and sediment transport model.

Once the external model is specified, one-or-more individual sediment fractions can call the van Rijn external model to specify settling, erosion or bed load. Additionally, van Rijn can also be used to globally assign bed roughness and bed shear models for all fractions. The compatible TRANSPOR models are listed as ‘van Rijn External (2004)’ in Table 1 and Table 2.

## 2.2 User Manual

The STM User Manual is available via the [TUFLOW FV Documentation](https://tufLOW.com/Download/TUFLOW_FV/ReleaseNotes/FV-Release-Notes_2020.pdf) page on the TUFLOW Website.

## 2.3 STM Bug Fixes and Enhancements

### 2.3.1 Bed mass layer output

An indexing bug has been fixed that resulted in sediment fraction masses being incorrectly assigned in multi-dimensional *bed\_mass\_layer\_#* map outputs.

### 2.3.2 Bed load transport at structures

To allow for bed load transport at hydraulic structures you can now enter the command:

`Bed load transport == 1` within a structure block. This will only work for nodestring structures. i.e. linked nodestrings are not supported. The default value is 0 or off. For example:

```
Structure == nodestring, 1
    Energy loss function == coefficient
    Form loss coefficient == 0.6
    Bedload transport == 1
End Structure
```

### 2.3.3 Exceeding the maximum number of sediment fractions

If constructing models with more than the maximum allowed sediment fractions (currently limited to 10), the following error message will now be thrown:

*ERROR: Number of sediment fractions (Nsed) is greater than maximum allowed (10).*

*ERROR-STACK:fvctrl\_construct: Number of sediment fractions (Nsed) is greater than maximum allowed (10).*

If more than 10 sediment fractions are required this number can be increased in future releases of TUFLOW FV (contact [support@tufLOW.com](mailto:support@tufLOW.com) if you require 10+ sediment fractions).

### 2.3.4 Changes to the DEFAULT material

Previous builds of TUFLOW FV used the integer number 0 to assign the default material. The new preferred method is to use the keyword 'DEFAULT' as follows:

```
Material == DEFAULT
...
Default commands...
...
End Material
```

This functionality has been included to avoid situations where material 0 had been used explicitly in a 2dm setup and where 0 was used a default in the sediment control file. For backward compatibility both 0 and DEFAULT are supported. This fix is also relevant to the Particle Tracking Module (refer Section 3.3.4).

### 2.3.5 Sediment Layer Outputs

2020.03 fixes issues associated with sediment layer result output requests if a sediment layer doesn't exist. Previous builds would result in a memory overflow runtime error. The new behaviour will still produce an output variable, however if not setup in the fvsed control file, the output will be assigned as a NaN (not a number).

### 3 Particle Tracking Module

The TUFLOW Particle Tracking Module (PTM) enables the 2D or 3D simulation of discrete Lagrangian particles as they are transported by a flow field and/or other forcing terms (e.g. wind drift). Particle behaviours such as settling, buoyancy, decay, sedimentation and resuspension can all be simulated.

The PTM is invoked through the HD Engine, which controls the overall simulation, supplies the forcing fields to the PTM and handles certain PTM outputs. The fundamental assumption that TUFLOW FV uses for its HD Engine, Advection Dispersion (AD), Sediment Transport (ST) and Water Quality (WQ) Modules is the use of a Eulerian or mesh-based frame of reference (an Eulerian scheme). This system discretises the model domain into individual cells that are used to solve the non-linear Shallow Water Equations (SWE). At each timestep the solution of these equations provides a single discrete value of a given result at each cell. In contrast, the PTM uses a Lagrangian Scheme that tracks individual particles through time, their trajectory not subject to being computed on a numerical mesh.

This tracking of discrete particles can be used to output particle age and fate, which are often useful metrics for environmental applications that are not easily modelled using the Eulerian scheme. Animal migration, search and rescue, ship navigation and salvage, pest species transport and environmental contaminants are a few of many useful examples of particle tracking.

#### 3.1 Available Particle Models

The PTM is built on a library of differing model/equation options as provided in Table 3 and Table 4. If you have a specific model or equation set not currently available that you would like to include in the PTM, please contact [support@tufLOW.com](mailto:support@tufLOW.com).

**Table 3 Lagrangian particle behaviour options.**

Particle forcing options	Description	Available options
Horizontal advection	Equal to the reconstructed flow velocity at the particle location. User can select whether vertical profile reconstruction of the horizontal velocity occurs.	<ul style="list-style-type: none"> <li>No vertical velocity profile reconstruction.</li> <li>Logarithmic profile reconstruction.</li> </ul>
Vertical advection	As above for 3D hydrodynamic models only.	N/A
Settling / Buoyancy	Vertical advection speed due to either negative (downwards) or positive (upwards) buoyancy.	<ul style="list-style-type: none"> <li>None</li> <li>A range of settling options</li> </ul>
Horizontal dispersion	Gaussian random walk component that can be applied as a function.	<ul style="list-style-type: none"> <li>None</li> <li>Constant diffusivity</li> </ul>
Vertical dispersion	Random walk component including higher-order terms to improve accuracy in the case of strong vertical concentration gradients.	<ul style="list-style-type: none"> <li>None</li> <li>Constant diffusivity</li> <li>Parabolic diffusivity</li> </ul>



		<ul style="list-style-type: none"> <li>• HD model diffusivity</li> </ul>
Particle motility	Particle advection velocity in response to user inputs (fixed direction/magnitude, speed/target, crawl) and/or local environmental conditions interrogated from hydrodynamic model (speed/currents, speed/waves).	<ul style="list-style-type: none"> <li>• None</li> <li>• Various motility models (refer Table 4)</li> </ul>
Wind drift	Particle drift as a function of wind velocity (where supplied by hydrodynamic model). Will only apply to surface particles.	<ul style="list-style-type: none"> <li>• None</li> <li>• Proportional to wind velocity plus direction offset</li> </ul>
Wave drift	Particle drift as a function of wave field (where supplied by hydrodynamic model).	<ul style="list-style-type: none"> <li>• None</li> <li>• Stokes drift (depth-averaged)</li> <li>• Lagrangian stokes drift (depth-varying)</li> </ul>
Erosion (Resuspension)	Pickup of particles located on the bed, into the water column. Pickup flux ( $\text{kg}/\text{m}^2/\text{s}$ ) is calculated using a specified particle erosion model (refer Sediment Transport Module documentation). A stochastic process is used to select individual particles for pickup.	<ul style="list-style-type: none"> <li>• None</li> <li>• Various options analogous to STM</li> </ul>
Deposition	Settling to the bed where deposition threshold criteria (if any) are satisfied. The fraction of particles allowed to settle is calculated using specified particle deposition model (refer STM documentation). A stochastic process is used to select and prevent individual particles from depositing.	<ul style="list-style-type: none"> <li>• None</li> <li>• Various options analogous to STM</li> </ul>

**Table 4 Description of particle motility options**

PTM Motility Options	Description
Velocity (horizontal)	Particle swims with specified velocity (u, v specification)
Speed, Waves	Particle swims at specified speed in direction oriented relative to waves
Speed, Currents	Particle swims at specified speed in direction oriented relative to currents
Speed, Target	Particle swims towards a target (X, Y in model coordinate system) at a specified speed
Crawl, Velocity	Particle crawls at specified velocity when located in dry cell (u, v specification)

## 3.2 User Manual

The PTM User Manual is available via the [TUFLOW FV Documentation](#) page on the TUFLOW Website.

## 3.3 PTM Bug Fixes and Enhancements

### 3.3.1 Particle restart file

An issue whereby particle restart files were not being read correctly via the `particle restart file ==` command has been fixed.

### 3.3.2 Particle seeding end time

A fix has been added to address models where particle seeding block `end time ==` commands were resulting in unexpected model completions as shown below.

```

Thread StackSize=4194304
PA
Entering timestep loop
t = 09/05/2019 23:00:00. dt = 0.000 / 0.000 s. elapsed time = 0.0 s.
Writing PTM_NETCDF output. t = 09/05/2019 23:00:00.
Writing PTM_NETCDF output. t = 09/05/2019 23:15:13.
Trying to open file: log\EMB_Amon_1_4_000_257303.253533_ptm.rst ... OK. File unit: 102
Closing file unit: 102
t = 09/05/2019 23:30:05. dt = 13.382 / 0.582 s. elapsed time = 21.2 s.
Writing PTM_NETCDF output. t = 09/05/2019 23:30:05.
Trying to open file: log\EMB_Amon_1_4_000_257303.501518_ptm.rst ... OK. File unit: 102
Closing file unit: 102
Writing PTM_NETCDF output. t = 09/05/2019 23:45:12.
Trying to open file: log\EMB_Amon_1_4_000_257303.753425_ptm.rst ... OK. File unit: 102
Closing file unit: 102
t = 10/05/2019 00:00:10. dt = 12.229 / 0.582 s. elapsed time = 44.9 s.
Writing PTM_NETCDF output. t = 10/05/2019 00:00:10.
Trying to open file: log\EMB_Amon_1_4_000_257304.002782_ptm.rst ... OK. File unit: 102
Closing file unit: 102
Writing PTM_NETCDF output. t = 10/05/2019 00:15:11.
Trying to open file: log\EMB_Amon_1_4_000_257304.252967_ptm.rst ... OK. File unit: 102
Closing file unit: 102

D:\Support\TSF20049\runs>PAUSE
Press any key to continue . . .

```

### 3.3.3 Particle restart file

When using particle restart files, water depths assigned to particles had been incorrectly assigned, leading to particles' status being set to 2 (i.e. on land). This has been fixed.

### 3.3.4 Changes to the DEFAULT material

Previous builds of TUFLOW FV used the integer number 0 to assign the default material. The new preferred method is to use the keyword 'DEFAULT' as follows:

```
Material == DEFAULT
```

```
...
```

```
Default material commands...
```

...

#### End Material

This functionality has been included to avoid situations where material 0 had been used explicitly in a 2dm setup and where 0 was used a default in the sediment control file. For backward compatibility both 0 and DEFAULT are supported. This fix is also relevant to the Sediment Transport Module (refer Section 2.3.4)

## 4 GPU Module

### 4.1 Introduction

TUFLOW FV's GPU Module is a powerful new solver built into the existing TUFLOW FV software. As its name implies it utilises the substantial parallel computing ability of modern Graphics Processor Units, or GPUs.

The front end is still TUFLOW FV, and only one additional line in the model input is needed to select the GPU solver. Similarly, the output data is still written by TUFLOW FV and the same output formats and data types are available. TUFLOW FV GPU has been extensively tested and benchmarked including the 2012 UK Environment Agency 2D Benchmark Tests, where along with TUFLOW and TUFLOW FV, were top performers in terms of consistent results and run times. The scheme has also been benchmarked against theoretical solutions and through comparisons with TUFLOW and TUFLOW FV for both 2D and 3D (TUFLOW FV only) simulations.

### 4.2 CUDA Enabled Linux Installers

To run TUFLOW FV GPU on Linux requires a CUDA enabled build of TUFLOW FV that we can readily provide along with further instruction on driver installation and setup. **Linux users please contact [support@tuflow.com](mailto:support@tuflow.com) for guidance on Linux CUDA enabled (GPU compatible) installation.**

### 4.3 Will the Results be the Same as TUFLOW FV on CPU?

Not quite. TUFLOW FV CPU and TUFLOW FV GPU simulations will produce slightly different results due to subtle differences in the hardware implementation of complex math operations such as `sqrt()` and `log()`. We do however expect results between the CPU and GPU versions to be very similar based on the extensive testing conducted to date. The areas more likely to exhibit differences are in regions of wetting and drying, however these differences tend to be highly localised.

### 4.4 Compatible Graphics Cards

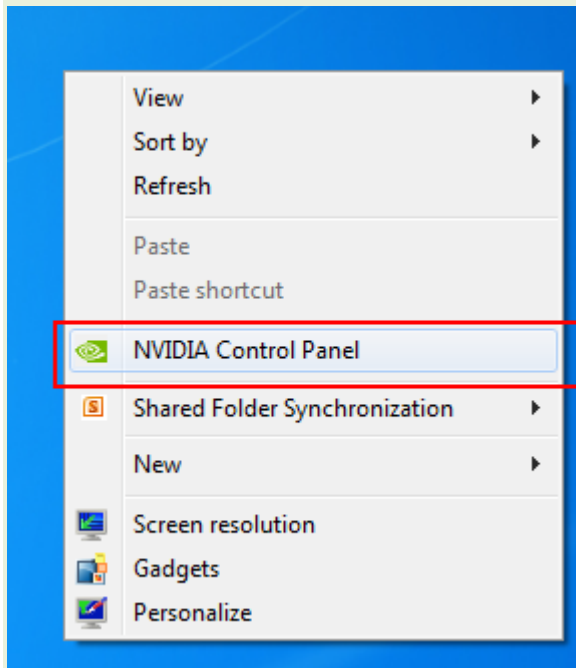
TUFLOW GPU requires an NVIDIA CUDA enabled GPU of Compute Capability 3.0 or higher. A list of CUDA enabled GPUs can be found on the following website:

<http://developer.nvidia.com/cuda-gpus>

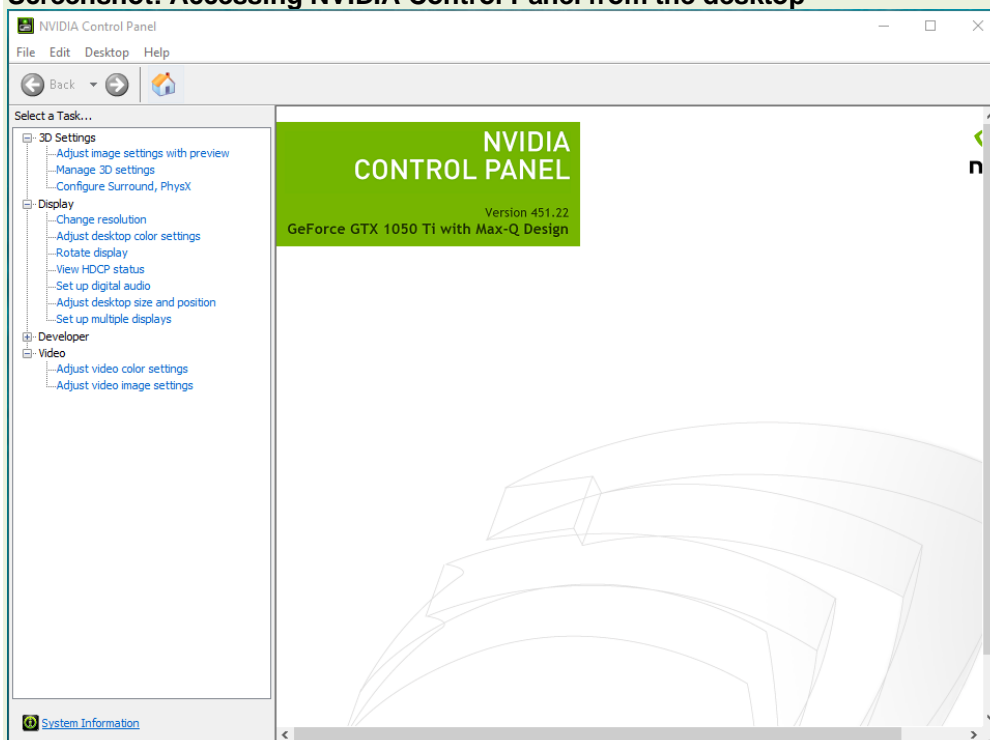
To check if your computer has an NVIDIA GPU and if it is CUDA enabled:

1. Right click on the Windows desktop
2. If you see "NVIDIA Control Panel" or "NVIDIA Display" in the pop-up dialogue, the computer has an NVIDIA GPU
3. Click on "NVIDIA Control Panel" or "NVIDIA Display" in the pop-up dialogue
4. The GPU model should be displayed in the graphics card information
5. Check to see if the graphics card is listed on the following website:  
<http://developer.nvidia.com/cuda-gpus>

The following screen images show the steps outlined above; this may vary slightly between NVIDIA card models.



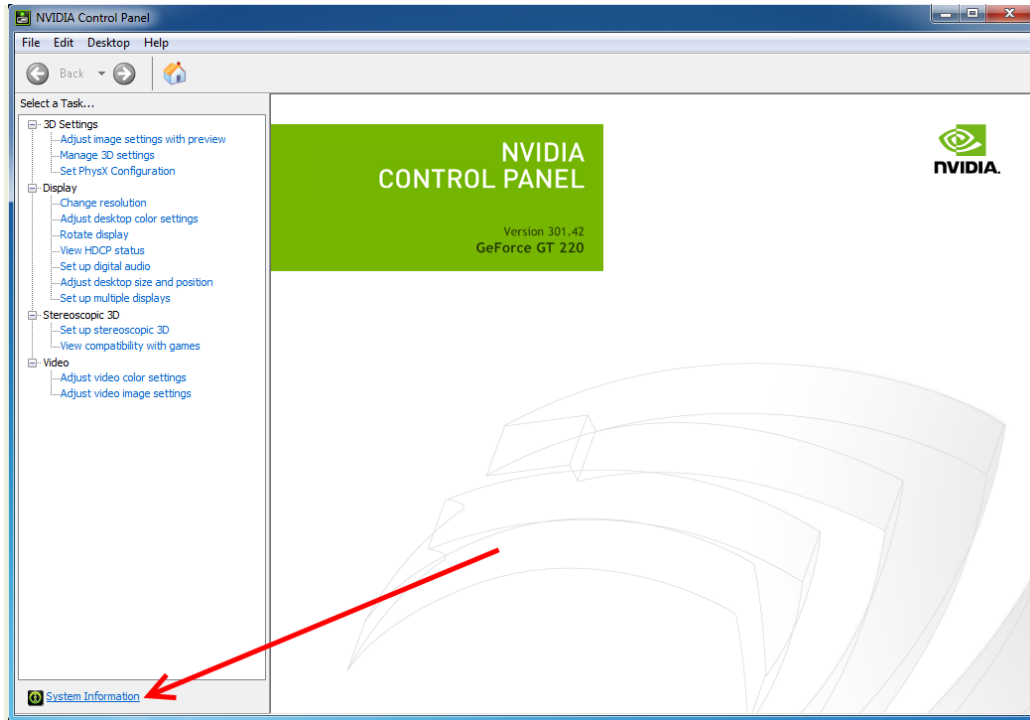
**Screenshot: Accessing NVIDIA Control Panel from the desktop**



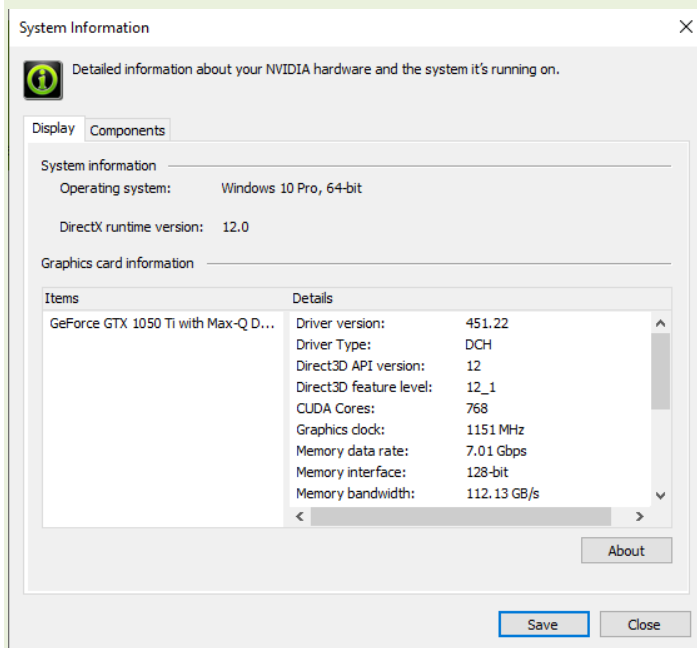
**Screenshot: NVIDIA GPU Model**

More information on the card can be found in the “System Information” section, which is accessed from the NVIDIA Control Panel. The system information contains more details on the following:

- The number of CUDA cores
- Frequency of the graphics, processors and memory
- Available memory including dedicated graphics and shared memory.



**Screenshot: Accessing System Information from NVIDIA Control Panel**



**Screenshot: NVIDIA System Information**

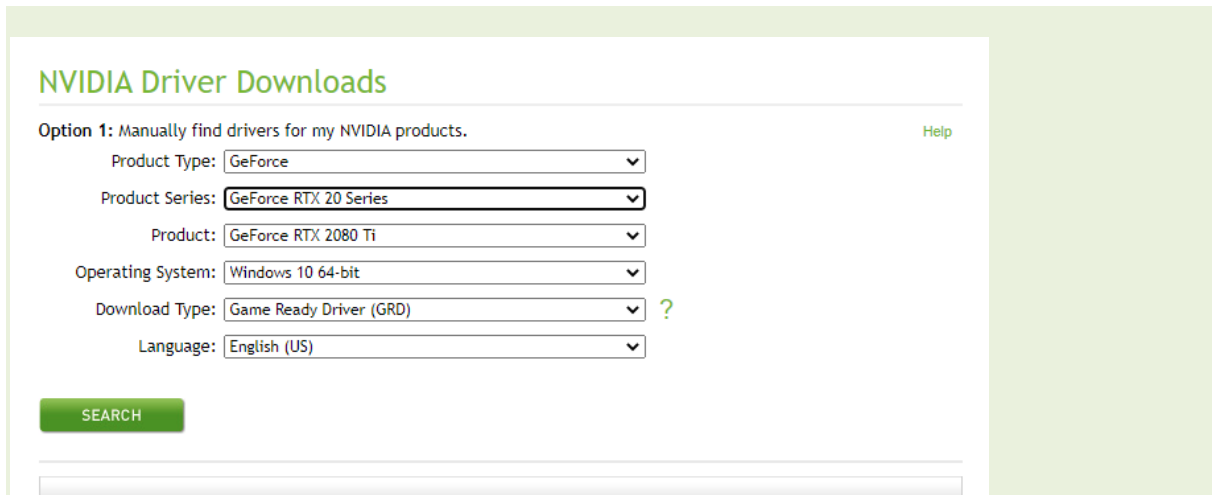


#### 4.4.1 Compute Capability

On the [NVIDIA website](#) each CUDA enabled graphics card has a “Compute Capability” listed.  
**TUFLOW FV requires cards with Compute Capability of 3.0 or later.**

#### 4.4.2 Updating NVIDIA Drivers

It is likely that the NVIDIA drivers will need to be updated to the latest version as the drivers shipped with the computers are usually outdated. To update, download the set of drivers for your specific card from the NVIDIA download page: <https://www.nvidia.com/Download/index.aspx>



#### Screenshot: NVIDIA Driver Selection

If new drivers are available, please download and install these by following the prompts.

**NOTE:** Even if not prompted by the system, a restart is recommended to ensure the new drivers are correctly detected prior to running any simulations.

## 4.5 Running TUFLOW FV on GPU

When running the GPU solver in TUFLOW FV, the pre (reading of data) and post (writing outputs) is managed by the standard TUFLOW FV engine. This allows the user to utilise the extensive range of existing functionality in TUFLOW FV.

To simulate the model via the GPU solver the command `Hardware == GPU` is required in the .fvc file. The TUFLOW FV GPU simulation is then started in the same manner as a standard TUFLOW FV simulation. This can be done via batch file, right click functionality in Windows or from a text editor.

A list of available control file commands is given in the section below.

### 4.5.1 TUFLOW FV Control File Commands

The following .fvc commands are specific to the TUFLOW FV GPU solver:

```
Hardware == GPU
```

This is required to initialise the GPU solver in TUFLOW FV. If this command is not present the standard CPU solution is used.

```
GPU Device IDs == DeviceID
```

This command is optional and may be used to select which GPU device is used (if multiple CUDA enabled GPU cards are available in the computer or on the GPU itself). Build 2020.02 only allows simulation on one GPU device, however multi-gpu support is in development and will be added to future releases. If you only have one GPU device, or you wish to use the primary device, this command is not needed. If there is more than one GPU device available, and you wish to select a particular device, *DeviceID* is entered using an integer value, noting that CUDA uses 0, 1, 2,...N-1 indexing. For example, to select the third device on a computer, use `GPU Device IDs == 2`.

The *DeviceID* can also be set via the command line (or terminal on Linux) when calling TUFLOW FV using the `-puN` argument where N is the CUDA GPU device index. For example, the below will use *DeviceID* 1.:

```
TUFLOWFV.exe -pu1 my_run.fvc
```

If setting *DeviceID* as a command line argument it will override any `GPU Device IDs ==` commands in the fvc.

## 4.6 Troubleshooting

If you receive the following error when trying to run the TUFLOW GPU model:

```
TUFLOW GPU: Interrogating CUDA enabled GPUs ...
```

```
TUFLOW GPU: Error: Non-CUDA Success Code returned
```

Please try the following:

1. Check the compatibility of your card and the latest drivers are installed (see instructions above).
2. Test with a user account that has administrator privileges as these may be required for running computations on the GPU.
3. If multiple monitors are running from the video card, try running with only a single monitor.

If the above steps fail to get the simulation to run, please email the NVIDIA system information (see instructions above) and TUFLOW FV log file (.log) to [support@tufLOW.com](mailto:support@tufLOW.com).

## 4.7 Command Line Updates for Hardware Selection

In Build 2020.03 it is now possible to override the .fvc `Hardware ==` command via either a -cpu or -gpu command line argument. switch. For example:

The following will run a model on GPU, on Device ID 0:

```
TUFLOWFV.exe -gpu my_run.fvc
```

Or GPU, on Device ID 2:

```
TUFLOWFV.exe -gpu -pu2 my_run.fvc
```

Or CPU:

```
TUFLOWFV.exe -cpu my_run.fvc
```

Note: When running on GPU it is no longer required to set OMP\_NUM\_THREADS=1 in your batch or shell script. If running on GPU, TUFLOW FV will now automatically set OMP\_NUM\_THREADS=1.

## 4.8 GPU Engine Bug Fixes and Enhancements

### 4.8.1 Moving Boundary Fix

2020.03 includes a fix for moving boundary conditions (for example a dredging barge) on GPU (CPU did not suffer from this issue). Previously if the boundary coordinates left the model domain a memory access violation would occur.

### 4.8.2 Improved Error Reporting

Cell error reporting has been improved when running on GPU hardware. 2020.03 will now report individual cell IDs if a cell value is out of bounds, NaN or other unstable model behaviour occurs, analogous to CPU error reporting.

### 4.8.3 2D Scalar Variable Fix

If running GPU in 2D with additional scalar variables, for example sediment, tracers or water quality variables, model runtime issues could occur at start up when the data arrays were passed over to the GPU solver. This has been resolved in the 2020.03 build.

### 4.8.4 Stokes Drift

In 2020.03 the calculation of Stokes drift is now supported on GPU and CPU hardware. Previously only CPU was supported.

#### 4.8.5 2D Transport File

A fix has been added to correctly copy 2D transport file data to and from the GPU. Previously only 3D models were supported.

## 5 Hydrodynamic Engine

This section describes updates to features implemented to commands and features applied within the TUFLOW FV Control File (.fvc).

### 5.1 Model Geometry

#### 5.1.1 GIS Integration

[Tutorial Modules 1 and 3](#) have been upgraded to include the optional assignment of nodestrings, cell materials and cell elevations independent of mesh generation using GIS layers during model initialisation. GIS integration in TUFLOW FV also provides you with access to a range of check files that make it easy to review your model. This set of GIS input and output files better aligns the workflow of TUFLOW FV with its sister products TUFLOW Classic and TUFLOW HPC and is a step towards full TUFLOW products integration.

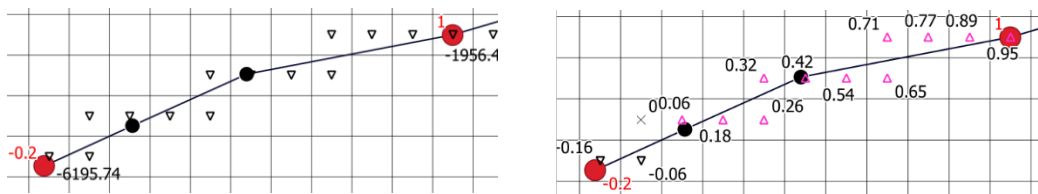
#### 5.1.2 Read GIS Z Line

A bug fix has been made in the Read GIS Z Line routines. In some circumstances where two or more vertices were located on a line between elevation points NULL elevation values were being assigned.

For example, the red points in the figures below are elevation points within the layer 2d\_zln\_Test\_001\_P.shp. The black dots represent vertices on the line 2d\_zln\_Test\_001\_L.shp. The black or pink triangles are from the \_zln\_zpt\_check model output check file layer.

`Read GIS Z Line == ..\model\gis\2d_zln_Test_001_L.shp | ..\model\gis\2d_zln_Test_001_P.shp`

The behaviour in the left and right panels below is the pre-2020 and 2020.01 release respectively. The previous behaviour was leading to incorrect assignment of ZB values.



#### 5.1.3 Read GRID Zpts

- Updates have been made to the Read GRID Zpts command to now assign elevations to both cell centres and cell corner vertices. Previously cell centroids were updated only and could lead to undesirable behaviour if used in combination with autoweirs.
- There have been several grid interpolation speed improvements implemented in the 2020.01 release.

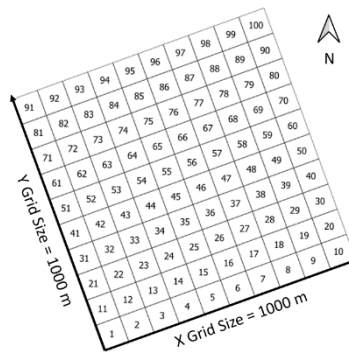
- BIL (Band Interleaved by Line) DEM files are now supported. Associated with this update is support for grids where the origin of the DEM is specified using XLLCENTER, XLLCORNER or ULXMAP conventions.

#### 5.1.4 Regular Mesh

There is now the option to specify a regular mesh in TUFLOW FV as an alternative to an external .2dm file. For example:

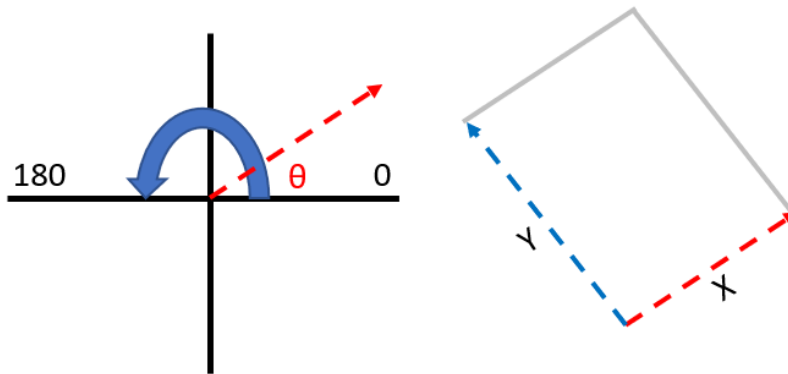
```
! Regular grid definition
grid origin == 548225, 7015813      ! x and y of mesh origin
grid rotation == 20.                ! (degrees)
cell size == 100., 100.             ! Cell size x, cell size y
grid size == 1000., 1000.           ! Grid size x, Grid size y
```

The resulting mesh with 2D element number labelled:



Grid rotation is specified by  $\theta$  (in degrees) positive from East as indicated in the left panel below. Negative values for  $\theta$  will rotate the grid clockwise from East (i.e.  $-20^\circ$  is equivalent to  $340^\circ$ ). The cell size can be independent in the x and y directions and is specified in m/ft or degrees when using cartesian or spherical coordinates respectively. The grid size is specified in m/ft or degrees and is the total length of the mesh in the X and Y direction as indicated in the right panel of the figure below. Using the cell and grid sizes specified TUFLOW FV internally calculates the number of elements required. For example:

Number of elements [X, Y] = grid size [X, Y] / cell size [X, Y]. If the cell size does not divide evenly into the grid size and extra element along the given dimension is added.



### 5.1.5 Nodestring Numbering

A fix has been included to remove a nodestring numbering bug that was affecting cell face selection if Nodestring ID 1, or 2 were not specified in the 2dm or as an external nodestring. This affected nodestring reads from both 2dm files and external nodestrings and could lead to erroneous nodestring location assignment for example at open boundaries, flux nodestrings and structures. If nodestring IDs were started at 1 in existing models this problem should be apparent.

## 5.2 Boundary Conditions

### 5.2.1 User Specified Grid Definition Block

Boundary grid definition blocks can be specified manually with the grid beginning at (X0, Y0) and extending in MX and MY in increments of DX and DY, with the grid rotated by ALP. The grid rotation ALP follows the rotational convention as described in Section 5.1.4. An example is provided as follows with values in curly braces i.e. {1} denoting default values. Previous behaviour required the grid definition block to be derived from an existing gridded dataset.

```
! BC Grid Definition Block
Grid definition == <X0>, <Y0>, <ALP>, <MX>, <MY>, <DX>, <DY>
  Grid definition label == <LABEL>
  Vertical Coordinate Type == < {elevation} | Depth | Sigma | Height>
  Cell Gridmap == < {1} | 0 >
  Boundary Gridmap == < 1 | {0} >
  Suppress Coverage Warnings == < 1 | {0} >
End Grid
```

### 5.2.2 Gridded MSLP Boundaries

The MSLP grid interpolation routines have been updated to fix a bug that was occurring if a gridded dataset did not cover entire mesh.

## 5.3 Model Outputs

### 5.3.1 CFL Stats

There is now a switch to optionally turn off the internal and external CFL output. If set to 0 TUFLOW FV will not produce a `_ext_cfl_dt.csv` and `_int_cfl_dt.csv` output file in the same directory as the log



file (.log). This may be desirable once the model is setup for design simulations to save hard drive space.

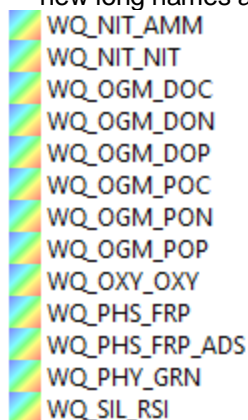
Output CFL Stats == < {1} | 0 >

### 5.3.2 DATV and XMDF Outputs

- The simulation reference time is now written to both XMDF and DATV outputs. Previously this was only written to XMDF outputs.
- An indexing bug that was causing models to crash when using minimum and/or maximum statistics output with XMDF results has been fixed.
- You will now receive a model error and message if you try to output incompatible multi-dimensional output to XMDF or DATV. For example:  
ERROR: Output parameter 'SETRATE' is not supported by the XMDF format.  
ERROR-STACK:fvdomain\_construct:fvinitout:fvout\_construct:Output parameter 'SETRATE' is not supported by the XMDF format.t.

### 5.3.3 NetCDF Outputs

- An indexing bug that was causing models to crash when using minimum and/or maximum statistics output with NetCDF has been fixed.
- NetCDF variable output names for water quality outputs have been changed so that the long name is set to the variable name (refer below image). This was completed to allow for easier management of water quality variables with post-processing software. The below example is the new long names as presented in the TUFLOW Viewer Plugin for QGIS (refer Section 9.3).



WQ\_NIT\_AMM  
WQ\_NIT\_NIT  
WQ\_OGM\_DOC  
WQ\_OGM\_DON  
WQ\_OGM\_DOP  
WQ\_OGM\_POC  
WQ\_OGM\_PON  
WQ\_OGM\_POP  
WQ\_OXY\_OXY  
WQ\_PHS\_FRP  
WQ\_PHS\_FRP\_ADS  
WQ\_PHY\_GRN  
WQ\_SIL\_RSI

### 5.3.4 Point Output

The TUFLOW FV 2D 'points' option provides model parameter time-series at point locations, each defined by an x, y coordinate specified within an [output points file](#). The output points file is a comma separated variable (.CSV) file with headers X, Y and ID (optional) and contains the coordinates for the point, or list of points, of interest. If the ID column is not included, then points are automatically assigned an integer ID starting from 1. An alternative to the ID column is to optionally provide a LABEL column containing a text string which is used to identify the location in the output points file.

For example, below are five example point files where time series outputs will be saved. The first four are acceptable formats. The far right will fail as it tries to use string with the 'ID' field and so causes TUFLOW FV to report an error. Note: The first two files on the left are equivalent.

✓			✓			✓			✓			✗					
X	Y		X	Y	ID		X	Y	ID		X	Y	LABEL		X	Y	ID
159.0758	-31.3638		159.0758	-31.3638	1		159.0758	-31.3638	5		159.0758	-31.3638	Point_1		159.0758	-31.3638	Point_1
159.0845	-31.3727		159.0845	-31.3727	2		159.0845	-31.3727	4		159.0845	-31.3727	Point_2		159.0845	-31.3727	Point_2
159.0906	-31.3814		159.0906	-31.3814	3		159.0906	-31.3814	1		159.0906	-31.3814	Point_3		159.0906	-31.3814	Point_3
159.1001	-31.3948		159.1001	-31.3948	4		159.1001	-31.3948	6		159.1001	-31.3948	Point_4		159.1001	-31.3948	Point_4
159.1154	-31.4032	or	159.1154	-31.4032	5	or	159.1154	-31.4032	2	or	159.1154	-31.4032	Point_5	not	159.1154	-31.4032	Point_5
159.1266	-31.4105		159.1266	-31.4105	6		159.1266	-31.4105	3		159.1266	-31.4105	Point_6		159.1266	-31.4105	Point_6
159.1202	-31.4165		159.1202	-31.4165	7		159.1202	-31.4165	7		159.1202	-31.4165	Point_7		159.1202	-31.4165	Point_7
159.1178	-31.4236		159.1178	-31.4236	8		159.1178	-31.4236	8		159.1178	-31.4236	Point_8		159.1178	-31.4236	Point_8
159.1277	-31.4402		159.1277	-31.4402	9		159.1277	-31.4402	9		159.1277	-31.4402	Point_9		159.1277	-31.4402	Point_9

- A fix has been added to remove an error that was causing point output csv results being set to NaN when using the *wq\_all* output parameter.

## 5.4 Hydrodynamic Engine Bug Fixes and Enhancements

### 5.4.1 Wu Horizontal Momentum Mixing Model

The TUFLOW FV 2020.02 build includes a new eddy viscosity (turbulence) model which is a slight modification of that described by Wu et. al. 2005<sup>1</sup>. **In its current implementation it is recommended that this approach be used on 2D models only and not 3D models.** Like the Smagorinsky horizontal momentum and scalar diffusivity models, 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 dependent on cell size. This has been shown to significantly improve the cell-size convergence of model results compared to the Smagorinsky model.

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

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

Friction velocity is derived from the 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 length scale,  $L_m$ , is set as the lower of either water depth or distance to the dry boundary.

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

With the Wu eddy viscosity formulation, the user definable eddy viscosity coefficients map to  $C_{3D}$ . It is recommended that user specified minimum and maximum eddy viscosity limits ( $m^2/s$ ) are included to provide bounds to the formulation. These coefficients and limits may also be applied within a material block using the horizontal eddy viscosity, horizontal eddy viscosity limits, horizontal scalar diffusivity and horizontal scalar diffusivity limits commands.

```
! Wu Horizontal Mixing Model Example
Momentum Mixing Model == Wu
Global horizontal eddy viscosity == 7.0                ! C3D
Global horizontal eddy viscosity limits == 0.05, 35    ! <Lower, Upper> m^2/s
```

Testing to date has found  $C_{3D} = 7$  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. As always, calibration remains an essential step, however, based on the testing and benchmarking so far, values significantly different to  $C_{3D} = 7$  (provided conventional bed roughness values are used for bed friction and any hydraulic structures are appropriately represented) are likely to indicate other errors (e.g. boundary values or schematisation, poor input data, etc). As always, sensitivity testing of changes in parameters on the model results should also be performed.

#### 5.4.2 New Turbulent Viscosity and Diffusivity Map Output

Turbulent eddy viscosity and diffusivity ( $m^2/s$ ) can now be output using the map output parameters *turb\_visc* and *turb\_diff* respectively.

#### 5.4.3 Structure flux output

A fix has been added to correct *structflux* and *structcheck* outputs that were being affected when using control structures. The timing of structure flows was being incorrectly reported to the output file with only the last entry of the output file receiving correct data.

#### 5.4.4 Profile Output

The TUFLOW FV 3D 'profile' output option produces a NetCDF file of model parameter time-series as vertical profiles.

```
output == Profiles
  output points file == ./Profile_Locations.csv
  output parameters == h,temp,tss
  output interval == 60
  suffix == Outflows
end output
```

The profile locations are specified at x, y coordinates within an [output points file](#). (e.g. the 'Profile\_Locations.csv' file specified above). This file is a comma separated variable (.CSV) file with headers X, Y, ID (ID optional), Vert\_min and Vert\_max and contains the X, Y coordinates for the profile, or list of profiles, of interest. Vert\_min and Vert\_max are the absolute z elevations across which profile data are to be reported. Setting these limits beyond the minimum and maximum cell

elevations will report the entire profile. Setting these limits to a subset of a model's vertical range will only report outputs in the vertical model layers contained within that subset. If the ID column is not included, then points are automatically assigned an integer ID starting from 1 and reported as "PT\_1", "PT\_2" etc. An alternative to the ID column is to optionally provide a LABEL column containing a text string which is used to identify the location in the output points file. This label is used to name the relevant field in the output netcdf file. Below is an example points file that reports profiles at two locations across the entire vertical range of a model.

	A	B	C	D	E
1	X	Y	LABEL	Vert_Min	Vert_Max
2	650572.5	7766732.613	Outflow_Location1	-10	10
3	650573.5	7766733.613	Outflow_Location2	-10	10
4					
5					

Each constituent is written to a <depth> by <timestep> matrix within the relevant field of the profiles NetCDF file.

#### 5.4.5 NetCDF Output Precision

For cartesian models (i.e. `Spherical == 0`, the default) the output precision of NetCDF cell centre and vertices has been increased to double precision. This was updated to allow for more accurate visualisation of high-resolution models. If using spherical models, single precision outputs are still used. Please note these changes affect NetCDF output only and do not affect the input precision of geometry used in the numerical engine which was always read in double precision.

#### 5.4.6 Linked Zone Structures

2020.03 added a fix for linked zone structures when discharging onto dry cells. Previously if the cell was dry no flow would occur.

#### 5.4.7 Non-Contiguous Nodestring Vertex Fix

A fix has been added to ensure the correct identification of nodestrings if mesh nodes are not specified in a contiguous manner in the 2dm. For example, the below snippet from a 2dm file shows non-contiguous nodes ID ordering ND: 1, 2, 3, 8, 9, 12, ...N. This could cause issues when assigning nodestring location. In 2020.03 both contiguous and non-contiguous node numbering are supported.

```
ND 1 1.52291640e+002 -2.48883579e+001 0.00000000e+000
ND 2 1.52291109e+002 -2.48880933e+001 0.00000000e+000
ND 3 1.52292252e+002 -2.48870242e+001 0.00000000e+000
ND 4 1.52290047e+002 -2.48875625e+001 0.00000000e+000
ND 8 1.52290578e+002 -2.48878287e+001 0.00000000e+000
ND 9 1.52291706e+002 -2.48867865e+001 0.00000000e+000
ND 12 1.52292816e+002 -2.48852659e+001 0.00000000e+000
```

#### 5.4.8 Improved Memory Allocation for GIS Layer Inputs

For Linux builds, memory stack allocation for GIS objects had been occasionally been leading to SEGFAULT issues. This issue could be mitigated by using the shell command `ulimit -s unlimited` command prior to calling TUFLOW FV. The memory allocation for these routines has been improved and should no longer required the `ulimit -s unlimited` command.

#### 5.4.9 Log file directory fix on Linux

Support has been added for '\' (backslash) characters in the log directory (`log_dir ==`) file path on Linux OS. On Linux builds pre-2020.03 having a backslash in the file path would result in incorrect creation of the log directory.

#### 5.4.10 Imperial Units Header Fix

When running with Imperial units, `_FLUX.csv`, `_STRUCT_FLUX.csv`, `_STRUCT_CHECK.csv`, and `_MASS.csv` will now output the correct units in the file header. Previously the header was incorrectly reporting metric units. Please note this incorrect reporting was confined to the header only and did not affect the timeseries data within the csv.

#### 5.4.11 New option to Enable or Disable Automatic Directory Creation

Automatic directory creation can now be overridden using the fvc command `Make Dir == 0`. The default behaviour, `Make Dir == 1` will create directories if they don't exist. Please note this command should be placed above the `log_dir ==` command in the fvc if this command is to apply to the log directory path.

#### 5.4.12 Cell Centred Output for PAR

A new map output parameter: `PAR_CC` can be specified to provide cell centre reporting of photosynthetically active radiation (PAR). PAR can also still be specified, however this provides results at horizontal cell faces, which then causes more data points per column to be written than other cell centred constituents such as SAL and TEMP. For example:

```
Output == netcdf

Output parameters == H, V, SAL, TEMP, PAR_CC

Output interval == 900.

End output
```

#### 5.4.13 External Inactive Materials

The use of inactive materials can currently only be applied to materials specified directly via the input mesh file (2dm). In 2020.03 an error will now be thrown if trying to use `inactive == 1` within a material block where the material ID has been set via an external material layer such as GIS or CSV polygon specified.

For example, if material 4 has been set via a GIS polygon and then implemented in the following material block:

```
material == 4 ! Upstream of Barrages
  inactive == 1
end material
```

TUFLOW FV will now throw the error:

*Error: Inactive material block specification, ID 4 is not supported with external GIS material layers.*

#### 5.4.14 Hydraulic Structures - Bathymetry Database

Changes in bathymetry/topography can be modified using the new *bathy\_database* and *bathy\_control* options for *zone* hydraulic structure types. The new zone structure options are applied to any cell centre within a zone polygon and can be useful to emulate scour or deposition events, without needing to complete a full sediment transport and morphological analysis. It also provides improved flexibility and control for modelling events such as levee overtopping or dam breach.

The bathymetry database is invoked using the structure block command `bed adjust == bathy_database`. The command `bathy_database == ..\model\structure\bathy_database.csv` is used to specify a table lookup of two or more digital elevation models (DEMs) that TUFLOW FV will use to interpolate bathymetry.

The following three examples are provided to showcase the new *bathy\_database* functionality.

**Example 1:** Bathymetry selection is controlled by a timeseries file:

```
structure == zone, 1

polygon file == ..\model\structure\zone_poly.csv
bed adjust == bathy_database
bathy_database == ..\model\structure\bathy_database.csv
update dt == 1. ! hrs

control == timeseries
  control parameter == bathy_control
  control header == my_values
  control update dt == 1. ! hrs
  control file == ..\model\structure\bathy_database_ts.csv
end control

end structure
```

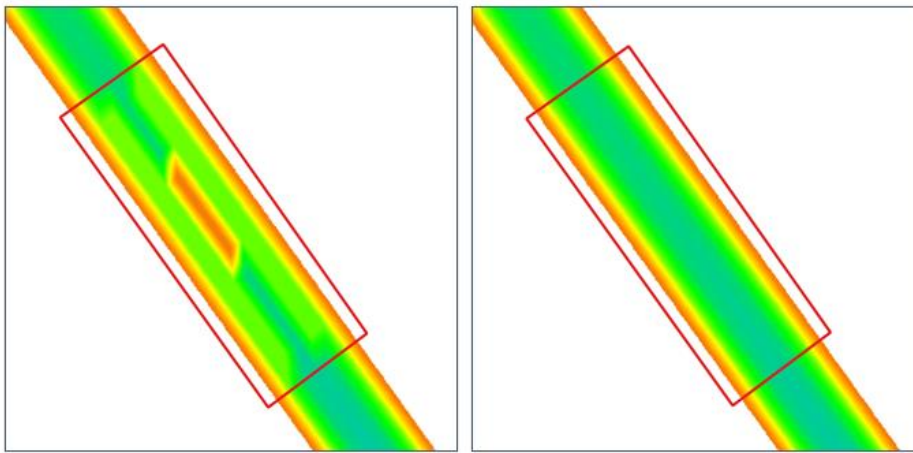
The comma delimited file `..\model\structure\bathy_database.csv` contains two columns as follows:

```
BATHY_CONTROL, BATHY_FILE
0.1, '..\dem\constricted.asc'
200., '..\dem\open.asc'
```

This provides a lookup/interpolation table to sample and assign bed levels. Values in the column BATHY\_CONTROL can be any decimal number and the column BATHY\_FILE contains dem paths.



The image below shows the two bathymetries, *constricted.asc* (left) and *open.asc* (right). The red polygon shows the zone region as demarked by the polygon file `..\model\structure\zone_poly.csv`.



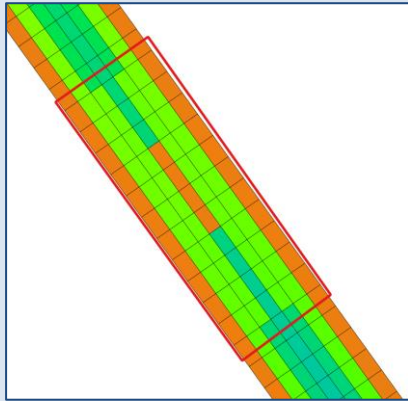
Example 1 contains a timeseries control option as provided via the `control == timeseries` block.

To adjust bed levels the bathy database is sampled using the control file: `..\model\structure\bathy_database_ts.csv` which contains a timeseries of lookup values:

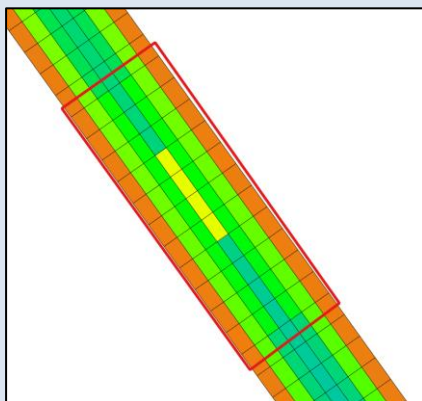
```
TIME, my_values
01/01/2020,0.
02/01/2020,100.
03/01/2020,200.
04/01/2020,200.
05/01/2020,100.
06/01/2020,0.
```

Where values of `my_values` are outside of the range of those specified in bathymetry database `BATHY_CONTROL` column, the closest dem will be selected. Data from `my_values` between ranges are linearly interpolated. For example at time `02/01/2020`, a `my_values` of `100.` will interpolate mean elevation values between *open.asc* and *constricted.asc* as shown via the following screenshots:

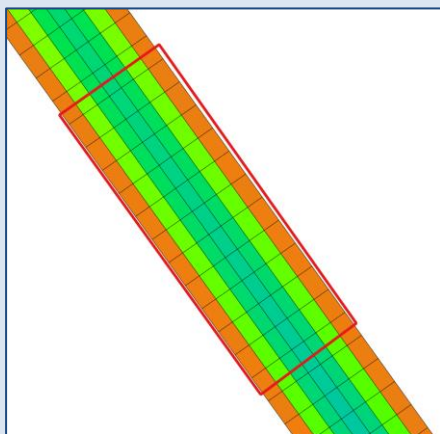
Model bathymetry at model time 01/01/2020 (equivalent to constricted.asc)



Model bathymetry at model time 02/01/2020 (halfway between constricted.asc and open.asc)



Model bathymetry at model time 03/01/2020 (equivalent to open.asc)



**Example 2:** This case is analogous to Example 1, however the structure is dynamically controlled by a sample parameter (flow) through a selected nodestring (in this case nodestring 1). The sampled flow provides the interpolation input to BATHY\_CONTROL. Equally, a sample point and water level, velocity or other sample parameter could be utilised.

```
structure == zone, 1

polygon file == ..\model\structure\zone_poly.csv

bed adjust == bathy_database

bathy database == ..\model\structure\bathy_database.csv

update dt == 1. ! hrs

control == sample_rule
    sample nodestring == 1
    sample parameter == flow
    sample dt == 1. ! hrs
    control parameter == bathy_control
    control update dt == 24. ! hrs
end control

end structure
```

**Example 3:** This case is analogous to Example 2, however it makes use of the new hydraulic structure control response function (refer Section 5.4.15). Here the response parameters allow for a relatively rapid 'erosion' as sample parameter values are increasing and slower 'deposition' period.

```
structure == zone, 1

polygon file == ..\model\structure\zone_poly.csv

bed adjust == bathy_database

bathy database == ..\model\structure\bathy_database.csv

update dt == 1. ! hrs

control == sample_rule
    sample nodestring == 1
```

```

sample parameter == flow

sample dt == 1. ! hrs

control parameter == bathy_control

start control state == 0.

response parameters == 0.25, 24. ! hrs (+ve trend, -ve trend)

control update dt == 24. ! hrs

end control

end structure

```

#### 5.4.15 Hydraulic Structures - Control Response Function

For *Timeseries* or *Sample Rule* control types, a *control parameter* can now be optionally scaled via the control response function:

$$Y_{new} = Y_{old} + dt/Tr * Y_{eq} / (1 + dt/Tr)$$

where:

$Y_{old}$  is the control parameter value from the previous control update  $dt$

$Y_{eq}$  is the current unmodified control parameter value

$dt$  = current model time – the model time at the last control update  $dt$

$Tr$  = response parameter time in hours specified using the control block command:

```
response parameters == Tr_pos, Tr_neg ! hrs (+ve trend, -ve trend)
```

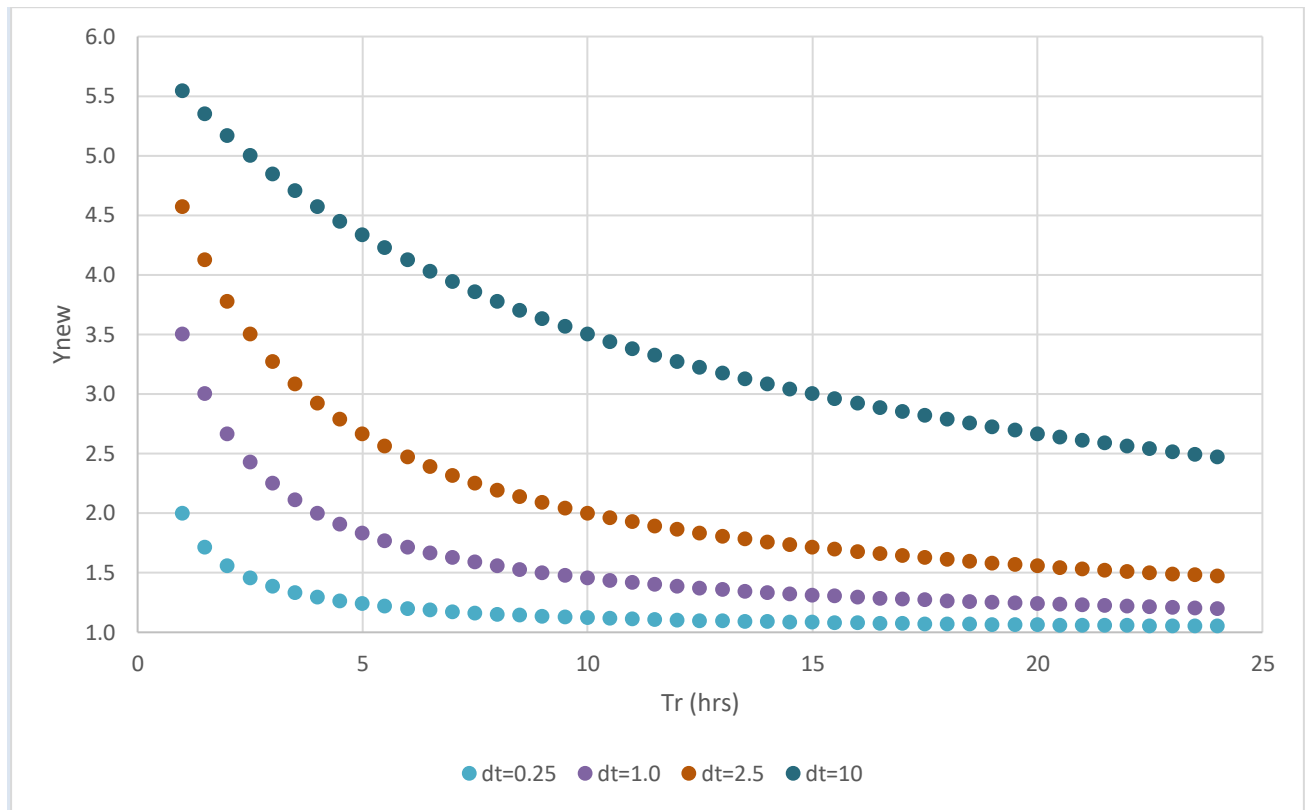
If  $Y_{eq}$  is greater than  $Y_{old}$ , then  $Tr = Tr_{pos}$

If  $Y_{eq}$  is less than  $Y_{old}$  then  $Tr = Tr_{neg}$

The control response function is enabled by having the `response parameters ==` command anywhere inside the control block. The default behaviour is off.

An example of using this feature in combination with the *bathy\_control* control parameter is provided in Example 3 of Section 5.4.14. When used in this context the response parameters are used to increase the rate of 'erosion' when flows are increasing ( $Tr_{pos} = 0.25$  hrs), and to slow down deposition rates where flow rates are decreasing ( $Tr_{neg} = 24.$  hrs).

The below illustrates the response function variation of  $Y_{new}$  as a function of  $Tr$ , assuming  $Y_{old} = 1.0$ ,  $Y_{eq} = 1.0$  for four differing values of  $dt$ ,  $dt$  0.25, 1.0, 2.5 and 10.0.



## 5.5 Fortran Compiler Optimisation

The TUFLOW FV code that runs on the CPU is compiled with Intel's fortran compiler, while the GPU code (written in C) is compiled with NVidia's CUDA compiler. Intel's fortran compiler offers many options for maximising execution speed, some of which involve low level optimisations for floating point operations that may vary in the last few bits of machine precision. After extensive testing it has become apparent that when the model tracks a number of scalar fields in addition to depth and velocity (for example a number of sediment or water quality variables), the CPU execution may change very subtly in a way which causes the evolution of other variables (such as salinity and temperature) to then vary slightly. The magnitude of the change is typically small and model specific, but may be exacerbated in models that require long term simulations with many tens of thousands of timesteps or models with long residence times, and model regions not flushed or affected by external boundary conditions.

For the technically minded, the development team has identified that the issue is with the use of the `/fp:fast` compiler setting, and that changing to almost any setting other than this fixes the issue. We have now chosen to select `/fp:consistent` for the new release.

With this change, a model with additional variables now yields identical hydrodynamic results to the same model without the additional variables (providing there was no coupling between those variables and the hydrodynamics in the first instance). The new compiler setting appears to have negligible impact on computational speed (i.e. less than 2%), and also appear to have significantly improved agreement between the CPU and GPU computation.

## 6 Advection Dispersion Module

The TUFLOW FV Advection Dispersion Module allows the simulation of salinity, temperature (and atmospheric heat exchange), suspended sediments (in combination with the Sediment Transport Module) and water quality constituents (in combination with the Water Quality Module).

### 6.1 Atmospheric Heat Exchange

There are changes to the treatment of sensible and latent heat fluxes in Build 2020.01 that may affect results in regions of wetting and drying when running with the heat module. If not using the heat module then results will be unaffected.

Previously, in some models we were seeing excessive cooling in shallow cells, during periods when the water depth was below the wetting depth. This cooling was occurring because the latent i.e. evaporative flux (heat and mass) terms were being applied while the sensible and long wave radiation terms were switched off. This could produce a negative bias to temperature response in shallow cells (latent heat fluxes are generally negative while the other heat terms may be negative or positive).

TUFLOW FV's updated implementation applies evaporative mass flux when the water depth is below the wetting depth, however all heat flux terms are switched off, except for a volumetric adjustment (due to the evaporative mass flux). The new approach will stabilise the temperature response in shallow cells. Note that this should be kept in mind when setting the wet depth for a model, particularly where the temperature dynamics within shallow cells is important.

Most models will likely be unaffected and/or changes will be limited to shallow areas where wetting and drying is occurring. Models that are most at risk of seeing larger changes will typically have one or more of the following characteristics:

- Large wetting depths. i.e.  $> 0.1\text{m}$ .
- Hot climates or during times of high evaporative summer weather.

In either of the above cases, internal testing has indicated that changes should be limited if the wetting depth is well selected.

### 6.2 Advection Dispersion Module Bug Fixes and Enhancements

#### 6.2.1 Evapo-Concentration

Previously all passive tracers would evapoconcentrate, that is mass would be conserved but concentration would increase in model cells experiencing evaporative fluxes. However, in certain circumstances a tracer constituent may be "volatile" and concentration will be maintained during evaporation. A flag has been added to optionally switch off the default evapo-concentration behaviour using the evapo-concentrate flag for tracer constituents (tracer volatility control):

```
tracer == 1
    evapo-concentrate == 0
end tracer
```

## 6.2.2 Evaporation Under Hyper-Saline Conditions

It is now possible to apply a vapour pressure (evaporation) correction for hyper-saline conditions via the equation:

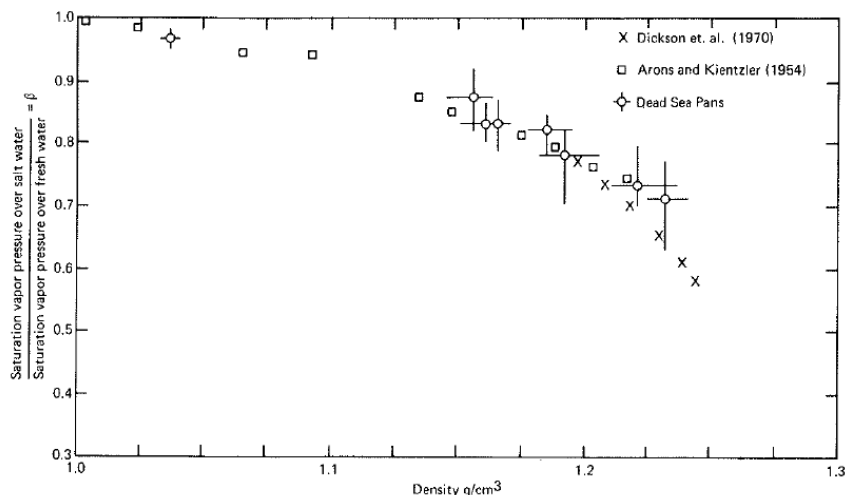
$$\beta = \max[(1 - aS - bS^2), c]$$

where:

$\beta$  is the ratio of saturation vapour pressure over saline water to saturation vapour pressure over freshwater (at the same temperature). This can be specified using the command:

Vapour pressure salinity params == 4.61e-4, 6.51e-7, 0.7 ! a, b, c defaults

The default values (a, b, c) represent a fit to the data published in Salhotra (1985).



Data used to define default vapour pressure salinity parameters. From: Salhotra, A.M., Adams, E.E., Harleman, D.R.F., (1985). *Effect of salinity and ionic composition on evaporation: Analysis of Dead Sea evaporation pans*. *Water Resources Research*, 21(9): 1336-1344.

## 6.2.3 Water Age Tracer

In certain studies it is desirable to track the age of water retained within the model domain. To facilitate these assessments a special tracer “type” has been added. Enabling the water age flag will switch on a tracer source term which effectively accumulates 1 unit of “age” per day of model simulation. The water age tracer “concentrations” will mix with any prescribed water age “concentrations” applied at inflow boundary conditions. Typically a “water age” of zero would be applied at inflow boundaries and initial conditions. Note: The water age flag switches off the default evapo-concentration behaviour for that particular tracer.

```
tracer == 1
  water age == 1
end tracer
```



## 7 Water Quality Module

### 7.1 Application Programming Interface (API)

TUFLOW FV 2020.03 includes an externally facing water quality API and can be used to communicate with the external water quality model (AED 2.0.0) and other compatible water quality models. This communication is different to that historically used between TUFLOW FV and previous versions of AED/AED2, and it recommended existing water quality modellers upgrade to AED 2.0.0 (refer Section 7.2 with this release of TUFLOW FV). To be clear:

- TUFLOW FV 2020.03 will not communicate with previous `tuflofv_external_wq.dll` binaries that have been built without using the new API. That is, users will not be able to use TUFLOW FV 2020.03 with versions of `tuflofv_external_wq.dll` that worked with TUFLOW FV 2020.02 and earlier
- TUFLOW FV 2020.03 will communicate with new compilations of the `tuflofv_external_wq.dll` binaries that have been built using the new API. This applies to both older water quality models (that use AED2) and newer water quality models (that use AED, see following subsection)

The same API will be used to support the upcoming release of the TUFLOW FV Water Quality module, so is a necessary precursor to the launch of our water quality model later in 2021.

This API implementation is a significant and deliberate advancement that formalises and codifies the way in which TUFLOW FV interacts with external water quality models. As such, users are now able to link their own water quality models to TUFLOW FV if they wish, using the framework set out in the TUFLOW FV API, and by replacing the placeholder `tuflofv_external_wq.dll` (distributed with TUFLOW FV) with their own compiled binary. Offering this is consistent with TUFLOW's intent to deliver a flexible and user customisable modelling tool to the water modelling industry.

The API is open source and distributed by TUFLOW. Users wishing to use it to link TUFLOW FV to their own water quality models should contact [support@tuflow.com](mailto:support@tuflow.com) for assistance.

### 7.2 AED Version 2.0.0

It is now possible to optionally couple with the advanced aquatic ecosystem model "AED", release: 2.0.0 (<http://aed.see.uwa.edu.au/research/models/AED>). This water quality model (referred to as "AED") supersedes the water quality model TUFLOW FV has linked to in previous releases (with this superseded model being referred to as "AED2"). Existing water quality models built with TUFLOW FV – AED2 are able to be converted to run with AED.

Older AED2 models will still run with TUFLOW FV 2020.03, if a simulation uses the AED2 version of `tuflofv_external_wq.dll`, where this dll has been built with the new API. Older AED2 versions of this dll will not link with TUFLOW FV 2020.03. If users wish to upgrade from AED2 to AED 2.0.0 and use the AED 2.0.0 `tuflofv_external_wq.dll` compiled with the API, then a conversion process between AED2 and AED driver files is required.

To affect this conversion, changes are required to the water quality name list file (\*.nml), as follows:

1. Rename “aed2.nml” to “aed.nml”
2. Replace all instances of the text string “aed2\_” with “aed\_” within the name list file. This can be undertaken with the Find and Replace function available in most text editors. Changes include:
  - a. “aed2\_bio” should be replaced with “aed\_bio”
  - b. “aed2\_models” should be replaced with “aed\_models”
  - c. All “aed2\_\*\*\*” model names listed within the models block should be replaced with “aed\_\*\*\*”. An example is provided below

<pre style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;">&amp;aed2_models ...models = ...!--&gt; NUTRIENT/CHEMISTRY modules &lt;--! ... 'aed2_sedflux',</pre>	becomes	<pre style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;">&amp;aed_models ...models = ...!--&gt; NUTRIENT/CHEMISTRY modules &lt;--! ... 'aed_sedflux',</pre>
---	---------	---

- d. All “aed2\_\*\*\*” model names that are headers for each name list block should be replaced with “aed\_\*\*\*”. An example is provided below

<pre style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;">&amp;aed2_oxygen ...oxy_initial = 225.0 ...Ksed_oxy = 150.0 ...theta_sed_oxy = 1.08 ...Fsed_oxy_variable = 'SDF_Fsed_oxy' ...oxy_min = 0 ...oxy_max = 500 /</pre>	becomes	<pre style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;">!----- &amp;aed_oxygen ...oxy_initial = 225.0 ...Ksed_oxy = 150.0 ...theta_sed_oxy = 1.08 ...Fsed_oxy_variable = 'SDF_Fsed_oxy' ...oxy_min = 0 ...oxy_max = 500 /</pre>
--	---------	--

3. If phytoplankton is simulated, and the name of the phytoplankton parameters file has been changed to reflect upgrading from AED2 to AED (e.g. aed2\_phyto\_pars.nml to aed\_phyto\_pars.nml), then the dbase reference to this phytoplankton parameters file will need to be updated in aed.nml, within the &aed\_phytoplankton block:

```
dbase = '\aed_phyto_pars_fv.nml'
```

The content and format of the phytoplankton parameters file does not need to be changed when upgrading from AED2 to AED.

4. If phytoplankton is simulated without resuspending to AED sediment (not to be confused with TUFLOW FV sediment), then the following command will need to be included in the &aed\_phytoplankton nml block:

```
resus_link = "
```

5. Depending on the AED configuration, this same command may also be required in the &aed\_organic\_matter block of aed.nml

If aed.nml is not located in the same directory as the TUFLOW FV control file (i.e. if the command `external water quality model dir == ..\<some directory>` is used), then the user should ensure that if a new AED directory is used (to differentiate from AED2) then this fvc command reflects that new folder location. A typical example is where a user has provided this command in the fvc:

```
external water quality model dir == ..\aed2
```

If a new folder called “AED” has been created to house the new aed.nml file, then this command will need to be changed to

```
external water quality model dir == ..\aed
```

Other fvc water quality commands (including the initial condition specification) can remain unchanged.

The user should ensure that the corresponding `tuflofv_external_wq.dll` (compiled with the new API) is located in the same directory as the TUFLOW FV 2020.03 executable.

Note that the AED2 (older) and AED (newer) `tuflofv_external_wq.dll`s have the same name. As such, particular care is needed to ensure that the AED2 dll is used with `aed2.nml`, and the AED dll is used with `aed.nml`. To allow checking, the model being used (i.e. AED2 or AED) in a water quality simulation is reported to screen (and log file) during set up as follows (red underlines, in this example AED is being used):

```
Configuring "AED" external module...
using fv_aed version 1.3.2
initialise aed_core
reading fv_aed config from ..\AED\aed.nml
link options configured between TFW & AED -
link_ext_par      : F
link_water_clarity : T
link_surface_drag  : F (not implemented)
link_bottom_drag   : F
link_wave_stress   : F
link_solar_shade   : F
link_rain_loss     : F
link_particle_bgc  : F (under development)
link_water_density : F (not implemented)
reading aed_models config from ..\AED\aed.nml
aed_sedflux initialization
aed_oxygen initialization
aed_carbon initialization
aed_silica initialization
aed_nitrogen initialization
aed_phosphorus initialization
aed_organic_matter initialization
aed_phytoplankton initialization
Solving temperature functions for phytoplankton
using the form : f(T) = v^(T-20)-v^(k(T-a))+b
configured variables -
S(      1 ) AED pelagic(3D) variable: OXY_oxy
S(      2 ) AED pelagic(3D) variable: CAR_dic
```

If AED2 is being used, then the water quality model will automatically look for a file called `aed2.nml` in either the fvc directory, or the directory specified by the `external water quality model dir` command. Similarly, if AED is being used, then the water quality model will automatically look for a file called `aed.nml` in the same locations. Therefore, if AED and AED2 have inconsistent dll or driver files, then an error message such as the following may appear at model setup time:

```
Configuring "AED" external module...
using fv_aed version 1.3.2
initialise aed_core
reading fv_aed config from ..\AED2\aed.nml
Cannot open file ..\AED2\aed.nml
```

This error tells the user that the AED version of the `tuflofv_external_wq.dll` binary is being used, but that the `external water quality model dir ==` command is pointing to the old AED2 directory (where only `aed2.nml` exists, not `aed.nml`).

### 7.3 Output headers

The 2020.03 build of TUFLOW FV has improved the reporting of AED2 and AED water quality variables so that column headers in FLUX, MASS, STRUCTFLUX and STRUCTCHECK output files now contain the water quality variable and corresponding units. These are reported consistently with TUFLOW FV hydrodynamic and other variables, in the format: `variable_name [units]`. For example, for a mass file output, the dissolved oxygen column header is `OXY_oxy_MASS [10^-3 mol]`.

## 8 General Updates

### 8.1 Enhancement to the QGS file

The .qgs file written by TUFLOW FV (runs\log) now includes projections. When opened in QGIS version 3.0 or later it should now correctly set the project and vector layer projections.

Additionally, qgs files write relative file paths rather than absolute paths. This should improve model review when transferring between computers or users, provided the folder structure is.

### 8.2 Shapefile Projection Changes

#### 8.2.1 Projection Check

Build 2020.03 introduces a more robust processing of the shapefile format projection (.prj) file to ensure that an input GIS file has the same projection as the model projection. Prior to 2020.03 this file was read as a character string and when a projection check was performed a string compare was executed. This could cause issues if moving between GIS platforms, e.g. between ArcMap and QGIS as these could write the .prj files slightly differently, particularly with a different number of decimal places. Build 2020.03 parses the parameter data into number values and compares each of these numbers to check they are consistent. A new .tcf command has been added to allow setting to the previous "Simple" string compare:

The consistency check for GIS projections can be turned from an ERROR (default) to a WARNING with the command: `GIS Projection Check == {ERROR} | WARNING`

## 9 Visualisation and Model Development Tools

### 9.1 MATLAB Toolbox

A toolbox for visualising 2D and 3D TUFLOW FV results in MATLAB is available for download from the TUFLOW website. <http://www.tuflow.com/FV%20Utilities.aspx>. A wiki page with instructions on using the MATLAB toolbox is available at:

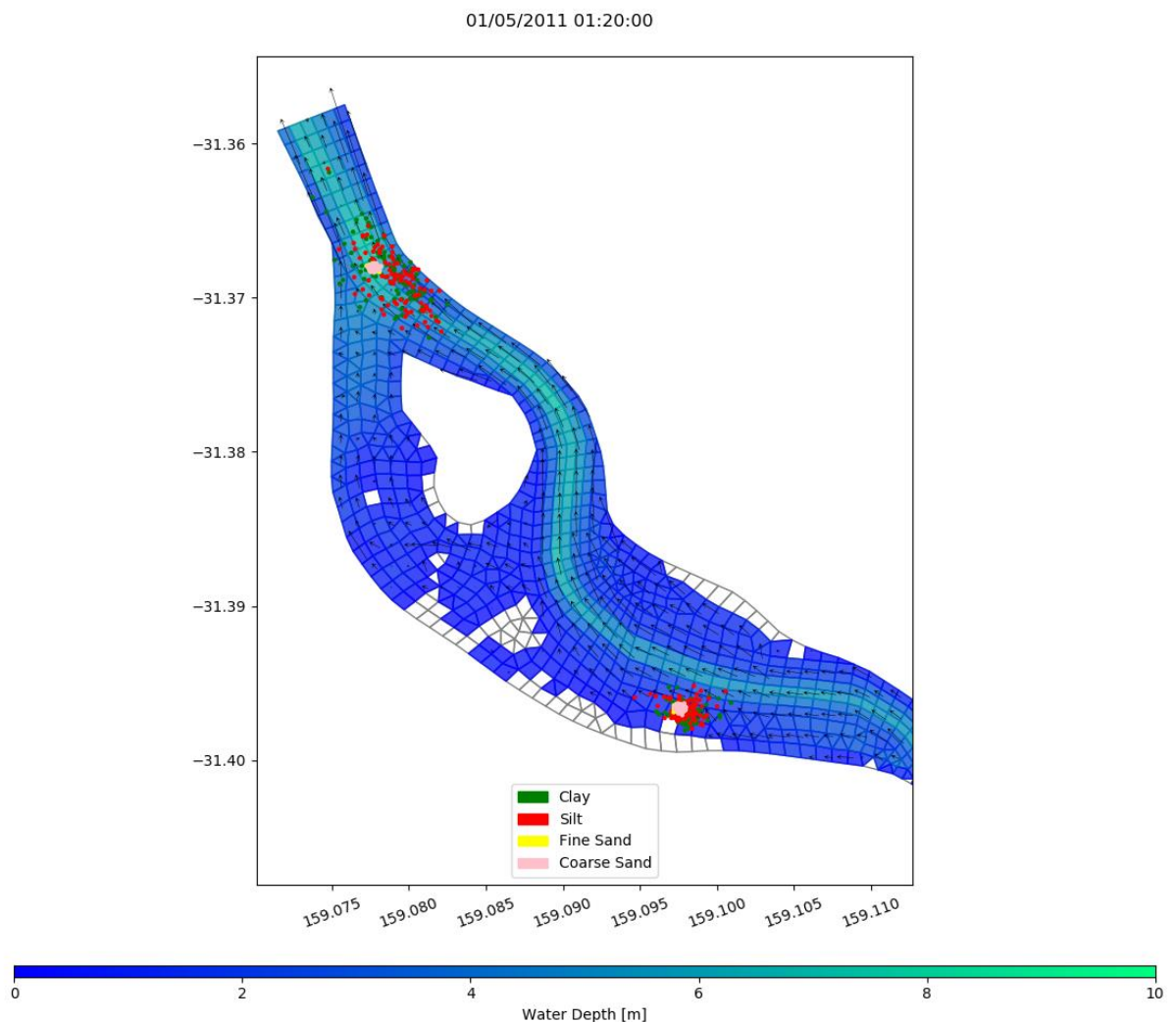
[https://fvwiki.tuflow.com/index.php?title=FV\\_MATLAB\\_Tools#MATLAB](https://fvwiki.tuflow.com/index.php?title=FV_MATLAB_Tools#MATLAB)

### 9.2 Python Toolbox

Tools from the MATLAB Toolbox have been ported over to Python 3 including sheet, curtain, profile and timeseries plots. Particle visualisation tools are also available. Please refer to:

[https://fvwiki.tuflow.com/index.php?title=FV\\_Python\\_Tools](https://fvwiki.tuflow.com/index.php?title=FV_Python_Tools).

The below figure has been produced using the Python Toolbox and overlays the model mesh, modelled water depth, velocity vectors (Eulerian outputs) and the trajectory of clay, silt and sand particles (Lagrangian PTM output).



### 9.3 TUFLOW Viewer QGIS Plugin

The [TUFLOW Viewer QGIS Plugin](#) provides tools to improve the efficiency of setting up, running and viewing the results of TUFLOW FV models. There is no cost associated with using the TUFLOW Viewer. Some of the available functions are described below:


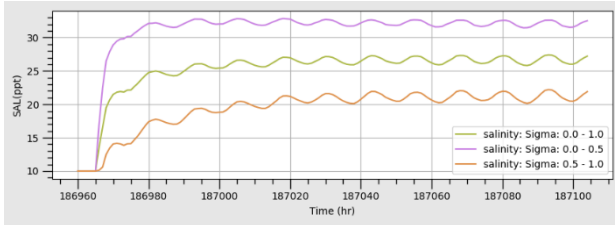

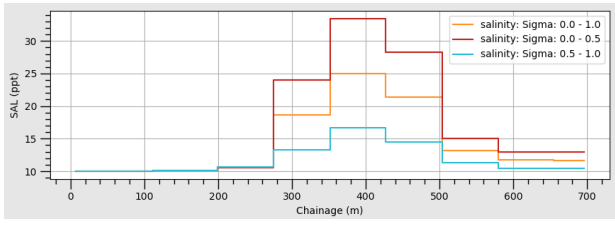
- Automated methods to create the TUFLOW FV folder directory and generate empty files.
- Incrementing the active layer by creating a copy, assigning a new revision number, and closing the original layer.
- Start a TUFLOW FV simulation from within QGIS.
- Tools to assist with model review and check files.

For further details on using the tools please consult the [GIS Integration](#) section of [TUFLOW FV Tutorial Model 1](#).


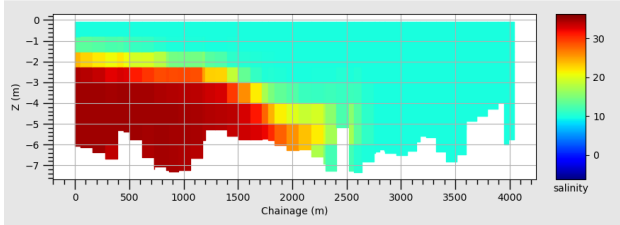

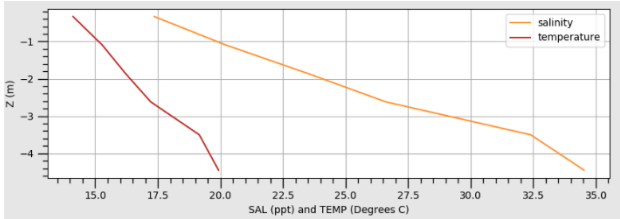
#### 9.3.1 3D NetCDF Mesh and Particle Result Viewing

A development build of the TUFLOW Viewer Plugin for QGIS builds 3.12 and later includes new options to visualise 3D NetCDF mesh and Lagrangian particle outputs in their native format. Please contact [support@tufLOW.com](mailto:support@tufLOW.com) and we can provide you with a copy and installation instructions. Table 5 below indicates the new 3D plotting features available.

**Table 5 TUFLOW Viewer 3D Plotting Options**

Name	Icon	Purpose/Features
3D Depth Averaged Point Time Series		<p>Multiple time series can be plotted using the 3D depth averaging methods: 'Sigma', 'Depth', 'Height', 'Elevation', 'Top Layer' and 'Bottom Layer'. Refer: <a href="https://fwiki.tufLOW.com/index.php?title=Depth_Averaging_Results">https://fwiki.tufLOW.com/index.php?title=Depth_Averaging_Results</a> for more information on available depth averaging methods.</p> 
3D Depth Averaged Cross Section		<p>Multiple cross sections can be plotted using the 3D depth averaging methods: 'Sigma', 'Depth', 'Height', 'Elevation', 'Top Layer' and 'Bottom Layer'. Refer: <a href="https://fwiki.tufLOW.com/index.php?title=Depth_Averaging_Results">https://fwiki.tufLOW.com/index.php?title=Depth_Averaging_Results</a> for more information on available depth averaging methods.</p> 



<p>2D/3D Long Section (Curtain)</p>		<p>2D or 3D long section with colour map of variable. The below example is a salinity long section (colour bar units are ppt).</p> 
<p>3D Point Profile</p>		<p>Vertical profile at a point location. Map output variables ( x axis) are plotted as a function of elevation (y axis).</p> 

## 10 Mesh Generation Tools

Although a separate component to TUFLOW FV, a suitable mesh generator is a fundamental component of flexible mesh modelling.

### 10.1 Aquaveo SMS

Aquaveo have recently released version 13 of SMS. Users can find more information via the links below:

<https://www.aquaveo.com/software/sms-surface-water-modeling-system-introduction>

<https://www.aquaveo.com/software/sms-whatsnew>

For a free trial of SMS, download an evaluation version from:

<https://www.aquaveo.com/downloads?tab=2#TabbedPanels>

There is also now a community edition of SMS that provides a reduced set of functionalities to the full version of SMS, but is still useful for result viewing and mesh generation. It can be downloaded from:

<https://www.aquaveo.com/software/sms-community>

### 10.2 Rising Water Software GIS Mesher

The GIS Mesher from [Rising Water Software](#) allows the user to create a mesh within their preferred GIS software and review it prior to running TUFLOW FV. Numerous GIS input file formats are supported including ESRI Shapefiles and MapInfo GIS files (tab or mif/mid). The shape, alignment and size of mesh elements can all be controlled using GIS layers. A series of mesh check files is also available for review.

The GIS Mesher supports the use of:

- Raster files (several formats are supported) to provide elevation data.
- Material assignment to mesh elements via GIS polygon layers.
- Nodestring assignment to mesh faces via GIS polyline layers.
- Topographic break line assignment using a combination of GIS polyline and point layers.
- Solution guided meshing.

For a demonstration of the GIS Mesher's functionality please refer to this series of short videos:

<https://www.youtube.com/channel/UC9SNGcShk6era1BpjKZdvZQ>

For more information on the GIS Mesher please check out the [GIS Mesher Website](#).

## 11 User Manuals, Tutorials and Supporting Materials

### 11.1 User Manuals

Available TUFLOW FV Manuals are available via the following TUFLOW Webpage:

<https://www.tuflow.com/FV%20Documentation.aspx>

The new features within these Release Notes will be progressively added to the relevant manuals. As documentation updates are being completed we recommend you refer to both the current Release Notes and latest [TUFLOW FV Manuals](#) to ensure you are up-to-date with the latest features. If unsure or if you have any queries, please contact [support@tuflow.com](mailto:support@tuflow.com).

#### 11.1.1 TUFLOW FV 2020 User Manual

Supporting the TUFLOW FV 2020.01 Release is an updated TUFLOW FV User Manual 2020. Specifically, the manual includes the following:

- **Advection Dispersion Chapter:** Overview of AD Module's role in transporting scalar variables such as salinity, temperature, suspended sediments, tracers and water quality constituents. Improved documentation on the atmospheric heat module is provided along with further guidance on the modelling on passive tracers.
- **Sediment Transport and Particle Tracking Overview:** Describes the required TUFLOW FV control file (.fvc) commands and links to the Sediment Transport and Particle Tracking User Manual.
- **Water Quality Overview:** Describes the required TUFLOW FV control file (.fvc) commands and links to the University of Western Australia's [Aquatic Ecodynamics Modelling](#) Library.
- **Waves Chapter:** How to integrate spectral wave modelling results in either one-way or two-way coupled mode.

#### 11.1.2 TUFLOW FV Science Manual

There is no addition material added to the Science Manual at this time. Heat module documentation will be the next inclusion. Scientific documentation concerning sediment transport and particle tracking is included in the Sediment Transport and Particle Tracking User Manual.

#### 11.1.3 TUFLOW FV Sediment Transport and Particle Tracking User Manual

A comprehensive Sediment Transport and Particle Tracking User Manual is now available for use with the Sediment Transport and Particle Tracking Modules of TUFLOW FV. This manual contains Scientific Documentation on the various sediment and particle model options available. The format of the Sediment Control File (.fvscd) and Particle Tracking Control File (.fvptm) is detailed with example code snippets and command appendices also provided. Visit the link below to download:

<https://www.tuflow.com/FV%20Documentation.aspx>

### 11.2 Tutorial Models

TUFLOW FV offers a range of tutorial models via our [TUFLOW FV Wiki](#). These models can be run license free by adding the following command to the top of the .fvc file:

```
Tutorial Model == ON
```

### 11.2.1 Module 1: Simple Trapezoidal Channel

The first tutorial module introduces the user to the TUFLOW FV software. The tutorial works through the steps required to develop a simple mesh model and run a simulation. The tutorial also offers an optional workflow to setup and use TUFLOW FV's GIS integration features. To access this module of the tutorial please see this page: [Tutorial Module 1](#).

### 11.2.2 Module 2: Simple River Bend

The second module builds a simple river bend model using the TUFLOW FV SMS interface. To access this module of the tutorial please see this page: [Tutorial Module 2](#).

### 11.2.3 Module 3: Floodplain Application

In the third module a real-world floodplain model example is used demonstrate the following:

- Mesh optimisation
- Geometry commands (e.g. break line commands, region commands)
- Optional workflow to setup and uses TUFLOW FV's GIS integration features
- Hydraulic structure commands (weirs, bridges, culverts)
- Advection dispersion modelling options (salinity, heat, temperature, tracer)
- Various output processing option

To access this module of the tutorial please see this page: [Tutorial Module 3](#).

### 11.2.4 Module 4: Coastal Application

The fourth module is a real-world coastal model example used to demonstrate the following:

- Application of a sloping water level boundary
- Cyclone/hurricane modelling using an internal Holland wind/pressure model
- Cyclone/hurricane modelling using an external wind/pressure and wave model
- Various output options

To access this module of the tutorial please see this page: [Tutorial Module 4](#).

### 11.2.5 Module 5: 3D Estuary Application

In the fifth module a small coastal estuary is examined that demonstrates:

- Assignment of 3D layering
- Running in 3D barotropic and baroclinic mode
- Setup of the atmospheric heat module
- Coupling with the External Turbulence Model GOTM
- Introduction in the use of TUFLOW FV's [MATLAB Toolbox](#) and [Python Toolbox](#) for 3D result visualisation
- Basic particle tracking example files (provided in the tutorial download package)

To access this module of the tutorial please see this page: [Tutorial Module 5](#).

### 11.2.6 NEW Module 6: 3D Particle Tracking Tutorial

The TUFLOW Particle Tracking Module (PTM) allows the 2D or 3D simulation of discrete Lagrangian particles as they are transported by the flow field (or other drivers such as wind or waves). Particle behaviour such as settling, buoyancy, decay, sedimentation and re-suspension can all be simulated. The tracking of discrete particles can be used to output particle fate and age, which is useful for purposes of animal migration, search and rescue, environmental contaminants and oil spill modelling.

The PTM is invoked through the TUFLOW FV hydrodynamic engine which controls the overall simulation and supplies the hydrodynamic forcing to the particle transport module. Tutorial Module 06 covers a number of aspects of particle tracking modelling, including:

- Add a single set of particles via a point source and simulate their fate
- Review the outputs using the QGIS TUFLOW Viewer plugin
- Add additional groups of particles via a polygon source
- Add deposition and erosion to the particles to allow them to interact with the bed

To access this tutorial please refer to: [https://fvwiki.tuflow.com/index.php?title=Tutorial\\_Module06](https://fvwiki.tuflow.com/index.php?title=Tutorial_Module06)

### 11.2.7 NEW Module 7: Riverine Sediment Transport Tutorial

The TUFLOW Sediment Transport Module is a flexible and powerful bed load and suspended load sediment transport model that enables the 2D/3D simulation of sediment transport in rivers, estuaries and coastal environments. One or more sediment fractions can be simulated as they are distributed within the bed and transported as bed or suspended load. The tutorial investigates a riverine system and floodplain including the setup and review of:

- Modelling suspended sediment
- Erosion and deposition processes
- Bed load and bed armouring
- Coupled bed morphology
- Sediment transport visualisation tools

To access this tutorial please refer to: [https://fvwiki.tuflow.com/index.php?title=Tutorial\\_Module07](https://fvwiki.tuflow.com/index.php?title=Tutorial_Module07)

### 11.2.8 NEW Module 8: SWAN GIS Tools

The SWAN GIS Tools provide a simple graphical user interface for the development of SWAN model control and input files and are designed to streamline the development of SWAN models via easy visualisation and automated procedures in a familiar GIS setting. The tools are provided for free as a sub-menu of the TUFLOW Viewer Plugin for QGIS.

Core functionality includes the ability to:

- Generate and visualise SWAN computational grids
- Inspect many Digital Terrain Models (DTM) onto computational grids
- Pre-process wind input grids and spectral wave boundary conditions

- Detect and connect nested simulations
- Generate SWAN model control files
- Post-process SWAN results for input to TUFLOW FV

To access this tutorial and for instructions on how to setup and install the SWAN GIS Tools please refer to: [https://fvwiki.tuflow.com/index.php?title=SWAN\\_GIS\\_Tools](https://fvwiki.tuflow.com/index.php?title=SWAN_GIS_Tools)

In addition to the SWAN GIS Tools Tutorial we have also updated the TUFLOW FV User Manual ([https://www.tuflow.com/downloads/#tuflow\\_fv](https://www.tuflow.com/downloads/#tuflow_fv)) to include wave commands (refer Chapter 14: 'Waves').

### 11.3 TUFLOW FV Wiki

We will be progressively adding new tutorial models (most likely sediment transport and particle tracking examples), tips and tricks and toolboxes. You can access the TUFLOW FV wiki via this link: <https://fvwiki.tuflow.com/>.

As the Wiki is updated, we will keep you informed via our next release notes via the TUFLOW FV Forum and the TUFLOW User Group on LinkedIn – see sections below for info/links.

### 11.4 TUFLOW FV Forum

The TUFLOW FV Forum provides a Question & Answer environment and has plenty of good information to help you with you TUFLOW FV modelling. You can register here: <https://www.tuflow.com/fvforum/index.php>.

### 11.5 TUFLOW LinkedIn User Group

If you're interested the TUFLOW Users Group on LinkedIn is a great place to keep up to date with all things TUFLOW: <https://www.linkedin.com/groups/1908583>.

## 12 Licensing and Installing

### 12.1 Linux Installer Packages

The RPM installer package has been updated to make it easier to install multiple versions of TUFLOW FV on your system. This package has been tested on the CentOS distribution.

There is also now a DEB installer that can be used for Debian Linux derivatives such as Ubuntu.

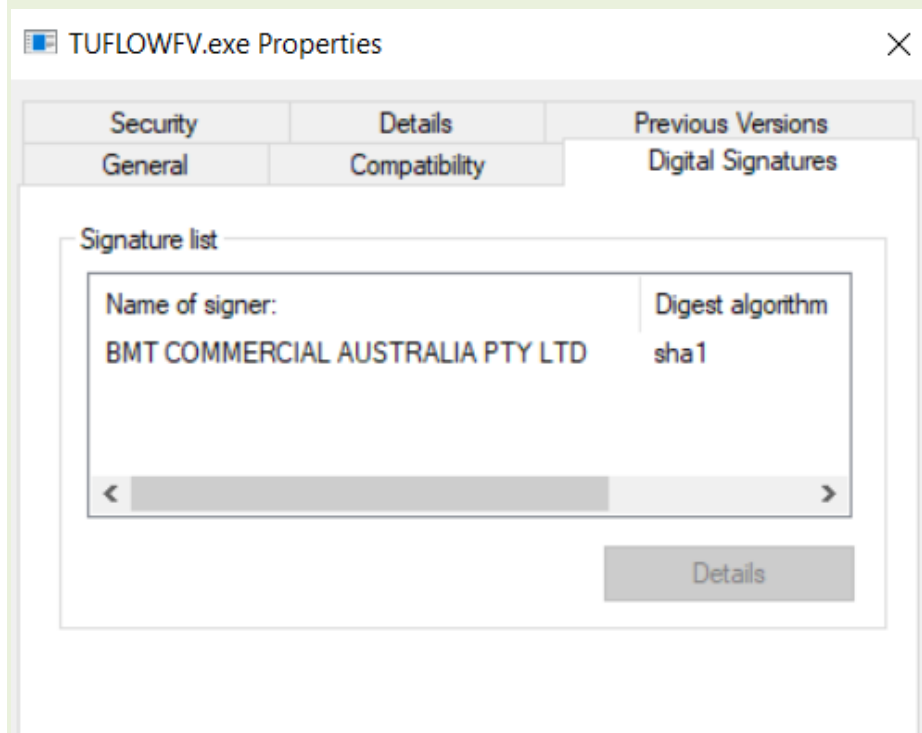
The installation process for both packages is provided on the [Linux Installation](#) page of the TUFLOW FV Wiki.

#### 12.1.1 Linux CPU/GPU

For Build 2020.02 it was required to have a separate RPM/DEB installer dependant on whether CUDA GPU libraries were installed or if the system was CPU only. Build 2020.03 now only requires a single installer for Linux that can be used for both CPU only and CPU/GPU platforms.

### 12.2 Security Certificate

Windows builds of the TUFLOW 2020.02 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.



### 12.3 Extra Licence Control File location

Build 2020.02 allows the user to specify the licence control file in “C:\ProgramData\TUFLOW\TUFLOW\_Licence\_Settings.lcf”. The search locations in decreasing order of priority are:



1. TUFLOW\_Licence\_Settings.lcf in the same location as the TUFLOW executable
2. C:\ProgramData\TUFLOW\ TUFLOW\_Licence\_Settings.lcf
3. C:\BMT\_WBM\TUFLOW\_Licence\_Settings.lcf
4. C:\BMT\_WBM\TUFLOW\_Dongle\_Settings.dcf